College of
ENGINEERING
VILLANOVA
UNIVERSITY

Center for Advanced Communications

FINAL TECHNICAL REPORT

Digital Watermarking of Autonomous Vehicles Imagery and Video Communications

May 2004-Septmeber 2005

Submitted to
Office of Naval Research
Grant number: N00014-04-1-0630

Principal Investigator
Bijan G. Mobasseri

Contributors
Dr. Yimin Zhang
Dr. Moeness Amin
Behzad M. Dogahe
Christopher Fleming

October 2005

ACKNOWLEDGMENTS

The PI gratefully acknowledges the support of the US Navy and Office of Naval Research under grant N00014-04-1-0630. Key contributors to this effort have been Dr. Yimin Zhang, Dr. Moeness Amin, Mr. Behzad M. Dogahe and Mr. Christopher Fleming. Generous support of Villanova Center for Advanced Communications has been essential to the success of this effort.

Digital Watermarking of Autonomous Vehicles Imagery and Video Communications

Executive Summary

We have developed, implemented and tested a known-host-state methodology for designing image watermarks that are particularly robust to compression. The proposed approach outperforms traditional spread spectrum watermarking across all JPEG quality factors. The fundamental approach is based on using 2D chirps as spreading functions, followed by chirp transform to recover the watermark. The reason for enhanced performance is the ability to spectrally shape the chirp to match image content and JPEG quantization. The energy localization of chirp is exploited to embed low power watermark per image blocks while maintaining reliable detection performance. In the course of this study we discovered that a chirp defined over a square grid is very susceptible to rotations of even small amounts. To address this difficulty we defined a *ring chirp* defined over a polar coordinate system. Embedding capacity is reduced but substantial robustness to image rotation is achieved.

Table of Contents

Publications

Reports

Presentations

ROTATIONALLY ROBUST DATA HIDING IN JPEG IMAGES USING A TUNABLE SPREADING FUNCTION

Christopher E. Fleming and Bijan G. Mobasseri
Electrical and Computer Engineering Department and
Center for Advanced Communications
Villanova University
Villanova, PA

ABSTRACT

Digital media is incredibly easy to create, store, copy and manipulate. For these reasons it is desirable to authenticate, trace and otherwise fingerprint digital media at the point of origination. These objectives can be accomplished by digital watermarking whereby a digital signature is embedded in imperceptible portions of cover media. Since the marked media may undergo friendly operations such as filtering, compression and scaling the watermark must be robust. One of the more difficult operations that the digital watermark must survive is geometric manipulation. In this paper we present a digital watermarking algorithm that can survive the combined compression/rotation attack and still remain detectable.

INTRODUCTION

In light of the world's recent events, the increasing need in homeland security and defense is a critical topic within the government. Because of this, the necessity for information security and analysis is also vital, especially when using digital media. With this in mind, digital watermarking applications for military use are becoming increasingly prevalent both on and off the battlefield.

Digital watermarking, by definition, is the process of embedding invisible signatures within a cover media with little or no perceptual impact. The cover media may take on a variety of forms including digital audio, video and imagery. The data or signatures which are embedded change with each application. Whether it is secure covert communications, image authentication or metadata embedding, the performance metrics for any watermarking process include perceptual transparency, security and robustness.

The most demanding constraint for any watermarking algorithm is perceptual transparency. Any information embedding into the host signal must remain undetectable to the Human Visual System. However, the information must also be detectable and extractable given the correct key by a trusted recipient. Security becomes an issue if and when the watermarked media is transferred via insecure channels. Equally important to these two requisites is robustness. Robustness not only to malicious attacks intended to destroy or remove the watermark, but also robustness to common, everyday operations. This paper explores the latter form, specifically the ability to survive both compression and rotational attacks.

Reviewed herein are standard block-based watermarking schemes, both spatial and spectral, and the shortcomings that come with rotational attacks. In addition, a solution is proposed involving a circular chip watermarking algorithm enabling it to survive both compression and rotation. These results and comparisons are further discussed within.

PREVIOUS WORK

Watermarking algorithms can be implemented in three different domains, those being: spatial, spectral and compressed. There are pros and cons associated with each. Compressed domain watermarking schemes benefit in that there is no need for full or partial decompression, which makes it ideal for real-time applications. However, such algorithms are not robust to any form of recompression. Because of this, only the former two domain watermarking will be further discussed.

Two significant algorithms involve the embedding of a pseudo-random (PN) sequence into the spatial and spectral domains respectively.

Arguably, the most widely used technique is spread spectrum watermarking (SS), an idea first introduced by Cox et al [2] whereby a DCT transform was performed across an entire image and the coefficients were modified by embedding a Gaussian sequence. These coefficients were perceptually significant when recreating the image; therefore in order to maintain transparency the change was fractional. Another SS watermarking technique was by Hernandez et al [5]. Here, Cox's idea was applied to 8x8 blocks of DCT transform coefficients, thus modeling itself after the JPEG standard itself. The idea utilized knowledge about the DCT coefficients and JPEG's quantization tables in order to determine an optimal watermarking location within the transform coefficients. The high frequency DCT coefficients barely distort the image but are removed too easily as compression increases. On the other hand, the low frequency DCT coefficients would survive compression very well but at the cost of greatly distorting the image. Ideally, it was found that the lower-middle frequency coefficients have the best range of acceptable compression resistance and minimal image distortion.

However, concept of SS watermarking is not limited to just the spectral domain. Spatial domain PN sequence watermarking involves the same design, however, the alterations are of the actual pixel values rather than DCT coefficients.

These two approaches utilize PN sequences to embed the data. Though both provide decent robustness, an improved approach to both spatial and spectral domain watermarking was suggested by Mobasseri et al [1]. The idea used a spatially embedded polynomial chirp signal [4].

BLOCK-BASED CHIRP WATERMARKING

The premise for using the block-based chirp is to produce a watermark structure, which spectrally, is unrelated with the cover media and capable of avoiding frequency-selective JPEG compression, all while maintaining imperceptibility.

The chirp is a block-based tunable spreading function for watermark embedding. The tunable values are the chirp rate, β, and the natural frequency, f. These values are optimized for each image resulting in the *key* used for the detection process. The chirp equation is defined as follows:

$$W(x,y) = e^{j\pi\beta\left(x^2+y^2\right)+j2\pi f(x+y)} \qquad (1)$$

The overall efficiency of the chirp algorithm is best proven in a performance comparison of the previously mentioned PN sequence spatial and spectral embedding methods. Below shows the results for each algorithms' bit error rate (BER) performance versus JPEG compression.

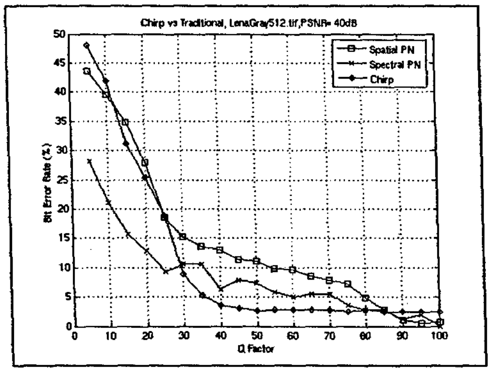


Figure 1: Chirp vs. Traditional Methods

As illustrated, the chirp noticeably outperforms both of the other embedding methods over the most common quality factors, between 40 and 90.

In addition to robustness to JPEG compression, the other attack that will be explored is rotation. Early results, shown in Figure 2, indicate that block-based embedding schemes are extremely fragile to rotational attacks, thus producing poor BER plots.

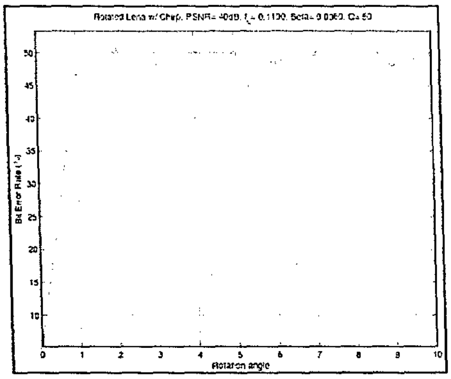


Figure 2: Rotation BER Plots

The lack of performance involving the chrip's block-based design is attributed to rotational displacement, or angle. Rotational displacement generally involves separate points at set distances that move circularly around a central point. When comparing the two points set at different distances rotation around the origin by the same angle, the point farthest from the center will move a greater distance. The same is said for blocks of points, or pixels. The farther blocks are moved or displaced (up to 45 degrees), the more data is lost or removed from the detection region. This can be seen in Figure 3.

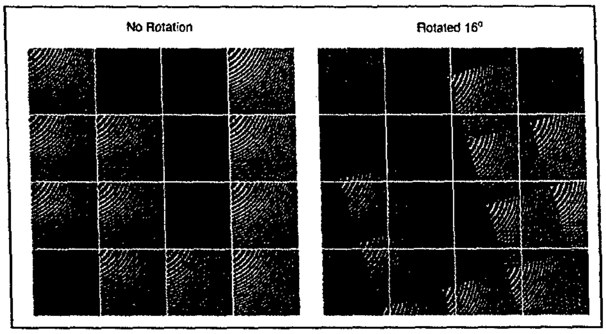


Figure 3: Block Rotational Displacement

Due to this loss and smearing of data across blocks, it is impossible to accurately detect the embedded watermark. This is even true when using a rotationally invariant algorithm such at the Fourier-Mellin Transform (FMT) [6,7,8] for detection. Though the results have improved, they are still unreliable and non-robust to all rotations.

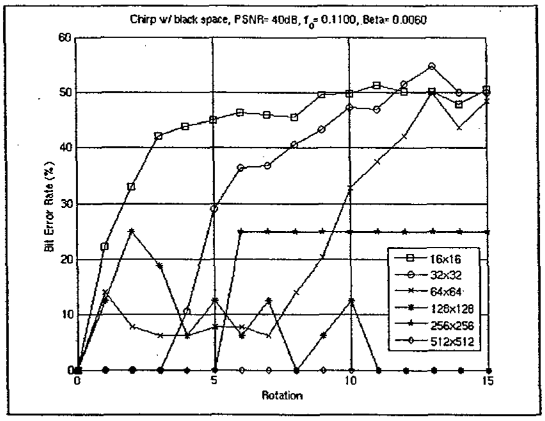


Figure 4: Block-Based Watermark with FMT Detection

Because of this a new approach is needed to ensure robustness to both JPEG compression and rotation attacks.

CIRCULAR CHIRP WATERMARKING

The previously examined watermark's vulnerability is due to its geometric shape. A circular watermark would correct this effect, since its geometric shape is conducive to rotation. Unlike the block-based watermark that loses its information after an amount of rotation, a circular watermark's data never leaves the image after rotation. This property alone gives a circular watermark a sizeable advantage over a block-based watermark.

The JPEG standard uses blocks in its quantization process, which was the basis of using a block-based watermark. This fact alone will give a block-based watermark an advantage when comparing it to a circular watermark. Converting the chirp watermark's equation to a polar mapping instead of a Cartesian mapping overcome this shortcoming. This can be seen in the equation below.

$$x = \rho\cos(\theta), \quad y = \rho\sin(\theta)$$
$$W(\rho,\theta) = e^{j\pi\beta\left(2\rho^2\right) + j2\pi f\rho\left(\cos(\theta) + \sin(\theta)\right)} \tag{2}$$

Circular watermarks are also at a disadvantage with its shape in some respect since a majority of images are rectangular. This would prevent the watermark from using the entire image's pixel values, because the circular watermark's content only goes to the image's sides. This is illustrated in the figure below were the unused outer regions are removed.

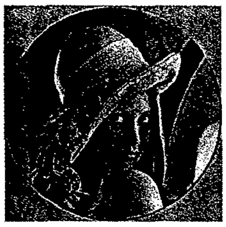


Figure 5: Detectable Content

If the image's corner regions were apart of the watermarkable content, then the overall performance would decrease, and therefore giving it an overall smaller possible capacity.

- **Embedding**

The circular chirp similar to its block-based counterpart is also embedded in the spatial domain, but interpolation is necessary due to its circular design. This is because circular based objects that use polar coordinates do not map directly to Cartesian coordinates. To perform the conversion seamlessly, the polar chirp spreading function begins in the polar domain. After interpolation, the circular watermark that is equivalent in size to the image is added directly to each corresponding pixel. An example of the resulting embedding operation is in the figure below.

Figure 6: Lena Original & Watermarked

- **Detection**

Similar to the block-based chirp, [ß,*f*], are necessary to detect the circular chirp watermark. These tunable values are used to recreate the embedded watermark, which is then used to correlate against the watermarked image. Unlike the block-based approach, the circular chirp performs its detection in the polar domain. This is because the polar domain converts rotation to translation, and it aligns the circular data into vectors necessary for correlation. In this domain cross correlation utilizes these properties to detect the signal after rotation. This can also be accomplished using the Discrete-Fourier Transform, but cross correlation allows the detector to possibly determine the angle of rotation. The detector then determines the polarity of the greatest correlation value and the information bit since they are antipodal binary values.

When data is interpolated from Cartesian to polar and vice versa visual distortion occurs. To thwart these effects, the recreated circular chirp endures the same polar to Cartesian and Cartesian to polar mappings as the watermarked image. An example of the interpolation process is displayed in the following figure.

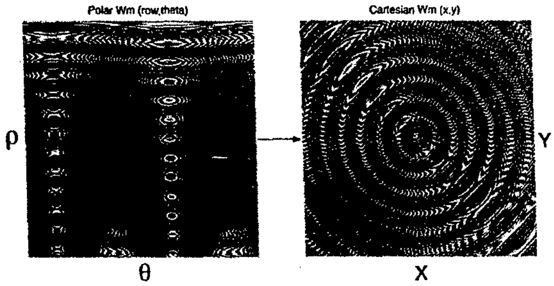


Figure 7: Polar to Cartesian Watermark Mapping

- **Security**

In the event an unintended user would obtain the image through an unsecured channel, ß and *f* are also useful as a security measure within this algorithm. Without the proper values for ß and *f* it is very unlikely that the message will be decoded. The complexity of an exhaustive search makes determining ß and *f* undesirable.

EXPERIMENTAL RESULTS

The effects of the circular chirp are better understood by comparing it with the block-based results. The figure below displays the output of a 255 ring circular chirp, which is a smaller overall capacity then the block-based algorithm. The capacity can be increased using sectoring, which is discussed in the future works section. The circular chirp on the other hand can survive the specified range of rotation.

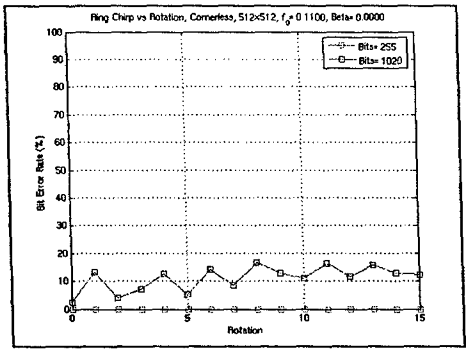


Figure 8:Circular Chip vs. Rotation

While this is an imageless case, it shows a significantly improved detection benchmark in comparison to the block-based rotation invariant scheme in figure 4. As the image is added to this scheme the performance should decrease as a result of a decrease in signal to noise.

After adding the circular chirp to the cover image, it is apparent that the previous hypothesis was correct. One example of a 16 ring circular chirp added to the cover image is shown in the figure below.

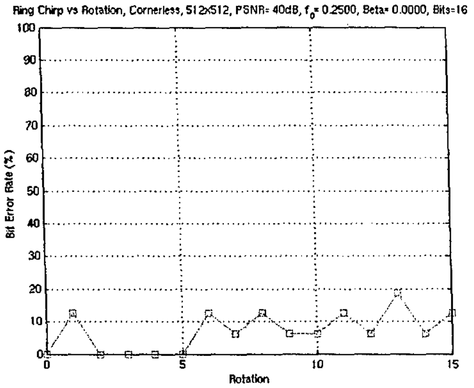


Figure 9: Watermarked Image vs Rotation

The noisy case has an average BER of 7%, which far surpasses the performance of the block-based approach. Other variables also affect the performance of the algorithm. As ring size increases the total amount of information available to the detector increases and therefore performance increases.

To obtain an optimum rotational performance for a specific ring size, a series of detection trials for a range of tunable variables are necessary. Using the detection outputs form the combinations of tunable variables, a rotational performance table can be created, and an optimal ß and *f* are established.

This also holds true for determining the optimal ß and *f* values for fighting compression. This technique was used in Mobasseri et al [1]. To help eliminate some unnecessary trials compression and rotation are joined to yield tables more conducive to this algorithm.

CONCLUSION AND FUTURE WORK

In this paper, the problem of rotation/compression resistant watermarking algorithms was described. While block-based watermarking has many beneficial traits when combating the effects of JPEG compression, but these same traits also hinder the watermark's ability to avoid the effects of rotation. This paper's research has explained that the geometric design of the block-based watermark is naturally flawed with respect to rotation, and a rotational design is necessary.

The circular chirp watermark combines the chirp's ability to avoid JPEG compression and a circular design to combat rotation. The noiseless results of the circular chirp alone show that it out performs the block-based chirp, and when the common attacks of JPEG compression and rotation are applied. Even though the chirp gives up some capacity to the block-based design, a circular chirp watermark is a solution to the block-based watermark's inability to avoid both JPEG compression and rotation.

Future work for this project will include a series of studies with respect to increasing capacity and the analysis of the tunable variable's significant digits.

The information structure of the circular chirp watermark is initially based on rings. Capacity within these rings can be increased if divisions are applied over a set quantity of angles. These divisions are also called sectors. By limiting the total range of possible rotations within the system, it is possible to increase the capacity greatly.

After running the several trials with rotation, it has become apparent that the values for ß and *f* are more unpredictable then original anticipated. By altering the current tunable value's range by another significant digit it should be more evident how conventional these values are.

ACKNOWLEDGMENTS

This research was supported by the United States Office of Naval Research, grant #N00014-04-1-06-30.

REFERENCES

[1] Bijan G. Mobasseri, Yimin Zhang, Moeness G. Amin, Behzad M. Dogahe, "Designing Robust Watermarks Using Polynomial Phase Exponentials", *ICASSP*, vol. II, Philadelphia, PA, March 18-23, 2005, pp. 833-836.

[2] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking of Multimedia," *IEEE Transactions on Image Processing*, 6, 12, 1673-1687, 1997.

[3] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," Proc. IEEE, vol. 87, no. 7, pp. 1127-1141, July 1999.

[4] Stankovic', I. Djurovic, and I. Pitas, "Watermarking in the space/spatial-frequency domain using two-dimensional radon-wigner distribution," *IEE Trans. Image Processing*, vol. 10, no. 4, pp. 650-658, Apr. 2001

[5] J. R. Hernadez, M. Amando and F. Perez-Gonzalez, "DCT-domain watermarking techniques for still images: detector performance analysis and a new structure," *IEEE Trans. Image Processing*, vol. 9, pp. 55-68, Jan. 2000.

[6] J. J. K. O'Ruanaidh, and T. Pun, "Rotation, Translation and Scale Invariant Spread Spectrum Digital Image Watermarking", Signal Processing, 66, pp.303-318, 1998.

[7] Kelly, Roberto M., H. S. Salguero, and E. S. Salguero, "Image Recognition Using the Fourier-Mellin Transform," *Instituto Politeco Naciopnal Mexico,* Mexico.

[8] Zheng, Dong, Jiying Zhao, and Abdulmotaleb El Saddik, "RST Invariant Digital Image Watermarking Based Log-Polar Mapping and Phase Correlation," IEE Trans. Circuits and Sys., vol. 20, no. 9, September 2003.

DESIGNING ROBUST WATERMARKS USING POLYNOMIAL PHASE EXPONENTIALS

Bijan G. Mobasseri, Yimin Zhang, Behzad M. Dogahe, Moeness G. Amin

Center for Advanced Communications
Villanova University
Villanova, PA 19085

ABSTRACT

In this paper, we propose a known-host-state methodology for designing image watermarks that are particularly robust to compression. The proposed approach outperforms traditional spread spectrum watermarking across all JPEG quality factors. The fundamental approach is based on using 2D chirps as spreading functions, followed by chirp transform to recover the watermark. The reason for enhanced performance is the ability to spectrally shape the chirp to match image content and JPEG quantization. The energy localization of chirp is exploited to embed low power watermark per image blocks while maintaining reliable detection performance.

1. INTRODUCTION

Digital watermarking is the process of securely embedding invisible signatures within a cover media with no perceptual impact. Developing a new watermarking algorithm requires the definition of five components, 1): cover media, 2): watermark, 3): embedding and extraction, 4): perceptual metric and 5): resilience and security criteria. Watermarking has been implemented in spectral as well as spatial domain. Arguably, the best known watermarking technique is spread spectrum (SS). One of the earlier references to SS watermarking is due to Cox[1]. Hernandez et al[2] have applied the same idea to 8x8 block DCT transform of images, closely following the JPEG standard. Their approach is similar to spatial domain SS watermarking proposed earlier by Hartung and Girod[3] which did not resort to masking models.

The concept of spread spectrum can be applied equally in spectral as well as spatial domains. In [4], each watermark bit is spread by a 2D modulation function and added to nonoveralpping sets of image pixels driven by a density metric. This is similar to the phase dispersion method proposed in [5]. The two approaches in [4,5] use the same model to spread watermark bits. While one uses a PN sequence, the other designs a carrier with flat spectrum but pseudorandom phase. SS watermarking is then an attempt to find the proper spreading function.

In this paper, we propose the use of polynomial phase exponentials, specifically a chirp function, as the spreading function. Chirps bring three properties to the table. First, chirp signals allow for tuning and spectral shaping of the watermark in a way that traditional spread spectrum watermarking using PN sequences are incapable of. Second, as a highly localized signal, chirp/watermark energy can be spread out in the image and then integrated at detection. This allows for low power watermark embedding on a local basis. Third, there has been considerable work in time-frequency processing techniques in the areas of speech, communications, fault structures, automation, biomedicine, radar, and sonar. These techniques provide easily accessible information about the signal spectral localization over short time periods and spatial segments[6].We apply the chirp transform and matched filter processing to both design and detect the proper chirp characteristics suitable for watermarking. The chirp transform applied in this paper does not account for the fast computations offered by the discrete chirp-Fourier transform proposed[7]. In a prior work, Stankovic et al[8] used chirps as digital watermarks by adding a single chirp to the entire image. This algorithm is best suited to copyright and ownership verification applications. The ability to embed and detect different chirps per image block, however, allows for data hiding applications where the extracted watermark may be an information-bearing bitstream. Another point of departure from[8] is exploitation of *known-host-state-methods*[9]. This approach was first suggested by Cox as communication problem with side-information[10], which was in turn based on Costa's dirty paper writing[11]. We have incorporated this idea into our work through tuning of the chirp. This observation is in contrast to spread spectrum watermarking, where spreading function, in the form of a PN sequence, is unrelated to host signal statistics.

2. RATIOANL FOR A NEW WATERMARK

We follow the watermarking model in [4]. It is desired to embed p bits $B=\{b_o,b_1,...,b_{p-1}\}$ in image $I(x,y)$. For

This research was supported in part by ONR grant N00014-04-1-06-30.

each bit b_i we define a 2D modulation function defined over set of pixels S_i .

$$s_i(x,y) = \begin{cases} p_i(x,y) \; if(x,y) \in S_i \\ 0 \end{cases} \quad (1)$$

$p_i(x,y): PN$ sequence

where $S_i \cap S_j = \varnothing, i \neq j$. The watermarked image is now defined by

$$I_w(x,y) = I(x,y) + w(x,y) \quad (2)$$

where

$$w(x,y) = \sum_{i=0}^{p-1} (-1)^{b_i} s_i(x,y)\alpha(x,y) \quad (3)$$

$\alpha(x,y)$ controls watermark strength. The above model spreads each watermark bit by a PN sequence and then additively modifies image pixels over the defined region of the image. Watermark detection is done through classical correlation detector, provided the decoder has access to the seed of the PN sequence, presumably communicated via a secret key exchange protocol. The proposed model, although effective, is suboptimal in the sense that no host-state-statistic is taken into account. In this work, we propose a class of watermarks that are *host-aware*. Robustness to compression is frequently a basic requirement in watermarking. In order to achieve such robustness, we suggest that watermark be spectrally shaped to escape JPEG. Since the high frequency suppression characteristics of JPEG standard is well known, it should be possible to design the watermark to remain relatively unaffected by compression.

3. WATERMARK EMBEDDING AND DETECTION

Partition an *NxN* image into *M* square blocks. A complex 2D chirp is defined as follows,

$$W(x,y) = e^{j\pi\left(\beta_x x^2 + \beta_y y^2\right) + j2\pi\left(f_x x + f_y y\right)} \{(x,y) \in 0, M-1\} \quad (4)$$

where β_x and β_y are chirp rates, f_x and f_y are initial spatial frequencies. For the rest of this paper, we use a single pair $\{\beta, f\}$. Spectral shaping can now take place by adjusting the pair $\{\beta, f\}$ as shown in Figure 1. Following watermarking model in (3), the image block located at pixels (m,n) is watermarked as follows,

$$I_w(m,n,x,y) = I(m,n,x,y) + k\,\mathrm{Re}[d(m,n)W(x,y)] \quad (5)$$

where [] stands for the integer part, $d(m,n)$ is the watermark bit drawn from *B* and *k* controls the PSNR in watermarked image. Watermark detection is based on 2D chirp transform defined in (6). To recover *B*, the decoder requires knowledge of the specific pair$\{\beta_o, f_o\}$ of the embedded chirp. This pair can be obtained in the chirp transform domain by seeking the peak of $C(m,n,\beta,f)$. This peak can be enhanced by integrating $C(m,n,\beta,f)$ over all image blocks followed by peak searching in (7).

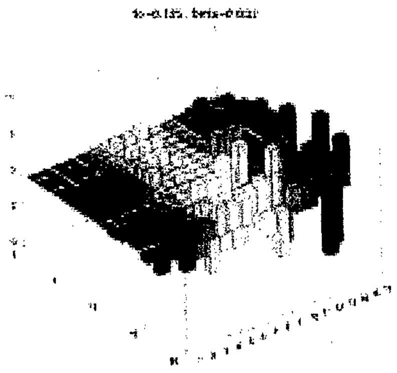


Figure 1- DCT of a 16x16 chirp.

$$\begin{aligned} C(m,n,\beta,f) &= \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} I_w(m,n,x,y) U^*(x,y,\beta,f) \\ &= \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} I(m,n,x,y) U^*(x,y,\beta,f) \qquad (6) \\ &+ \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} k\,\mathrm{Re}[d(m,n)W(x,y)] U^*(x,y,\beta,f) \end{aligned}$$

with

$$U(x,y,\beta,f) = e^{j\pi\beta\left(x^2+y^2\right) + j2\pi f(x+y)}$$

$$C(\beta,f) = \sum_m \sum_n |C(m,n,\beta,f)| \quad (7)$$

Eq(7) provides the ability to distribute watermark power over the entire image then integrate. This power distribution makes watermark detection more difficult by unauthorized users since each block alone will not carry sufficient power for reliable detection. Note that $C(\beta,f)$ for an unmarked image will peak at $\{\beta,f\} \cong 0$. To prevent others from performing identical peak detection and recover the same information, the following procedure is implemented. Instead of embedding the same chirp in every block, we draw from a family of $\{\beta,f\}$ and embed different pairs in different blocks. This association is then communicated to the decoder via secure key exchange. Unless this key is known and image blocks are de-chirped with correct$\{\beta,f\}$, the attacker will not observe a peak in (7). The embedded bit can now be recovered from the sign of $C(m,n,\beta_o,f_o)$ as follows,

$$\hat{d}(m,n)=\begin{cases}1 & \mathrm{Re}\{C(m,n,\beta_o,f_o)\}\geq 0\\ 0 & \mathrm{Re}\{C(m,n,\beta_o,f_o)\}<0\end{cases} \tag{8}$$

The bit error rate (BER) is controlled by the relative strength of projection of the chirp on the image vs. the chirp itself. Let

$$I_w = I + kdW_{\beta f}, d \in \{+1,-1\} \tag{9}$$

Decoder output is given by,

$$\begin{aligned}&< I_w, W_{\beta f} > = < I, U_{\beta f} > + kdE_{\beta f}\\ &where\\ &E_{\beta f} = < W_{\beta f}, W_{\beta f} >\end{aligned} \tag{10}$$

Correct detection is guaranteed if $|< I,U_{\beta f} >| < E_{\beta f}$. Probabilities of error are given by (11). It is noteworthy to point out that not every block needs to be watermarked. For example, a block that is slated to carry +1 and satisfies $< I,U_{\beta f} >> kE_{\beta f}$ is left alone since the decoder will decide in favor of +1 anyway.

$$\begin{aligned}P(e|s=-1) &= P(< I,U_{\beta f} > \ > kE_{\beta f})\\ P(e|s=+1) &= P(< I,U_{\beta f} > \ < -kE_{\beta f})\end{aligned} \tag{11}$$

4. JPEG COMPRESSION AND SPECTRAL SHAPING

Below, we show the flexibility of using chirp over traditional SS watermarking in compressive environments. SS watermarking offers substantial robustness to compression [12]. This robustness is achieved through the available processing gain. Increasing processing gain in spread spectrum watermarking comes at the cost of reduced embedding rate. The reduction in embedding rate is due to the fact that higher processing gains can only be achieved through using larger image blocks. In chirp-based watermarking, on the other hand, robustness to compression is achieved in an entirely different manner. To prevent JPEG from removing the watermark, it is possible to spectrally shape the chirp to make it survive compression. This shaping can be achieved by varying $\{\beta, f\}$ and monitoring BER. In contrast, the PN sequence in SS watermarking has no such tuning capability and will retain a white spectrum regardless.

The key issue is the chirp selection which survives specific compression factor. Rewrite (5) as follows

$$I_w = I + kdW_{\beta f}, d \in \{+1,-1\} \tag{12}$$

Define JPEG quantization matrix by $Q = [q_{ij}], \{i,j = 1,...,8\}$. Quantized DCT coefficients of watermarked image block is given by,

$$\left[\frac{dct(I_w)}{Q}\right] = \left[\frac{dct(I)}{Q} + \frac{dct(ksW_{\beta f})}{Q}\right] \tag{13}$$

where [ ] designates rounding to the nearest integer. Division in (13) is a term-by-term division of two 8x8 matrices. Dequantized image block is given by,

$$I_w^* = dct^{-1}\left(Q \times \left[\frac{dct(I)}{Q} + \frac{dct(ksW_{\beta f})}{Q}\right]\right) \tag{14}$$

I_w^* is then used in (6). The way the watermark is eliminated by compression is through quantization. Since quantization is a nonlinear operation and it is the sum of image and watermark components that is quantized, DCT values of the chirp alone, without considering the content of the image, does not determine watermark survival. Watermark can be considered removed if DCT coefficients are quantized to the same value, with or without the watermark. Since both image and watermark are available to the encoder, it is possible to ensure watermark survival by choosing appropriate $\{\beta, f\}$ pairs. A finer point here is that watermark survival is not absolute; there are different degrees of watermark content in I_w^*. The reason is that there are actually 64 terms in (13). Theoretically, even if one frequency out of 64 retains the watermark, watermark has survived but may not be reliably detected, resulting in large BER.

We can quantify degree of watermark survival by the following measure,

$$e = \frac{1}{M}\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\left|I_w^*(i,j) - I^*(i,j)\right| \tag{15}$$

$I^*(i,j)$ is unmarked compressed image block. If this difference is zero, then it is concluded that the watermark is entirely removed by compression. For a fixed PSNR and compression ratio, e is a function of $\{\beta, f\}$. Chirp design amounts to selecting the pair that results in large e.

5. EXPERIMENTS

Our test image is *lena* in grayscale. The 512x512 image is divided into 16x16 blocks for a total 1024 blocks. The embedding capacity for this image is then 1024 bits. In order to tune the chirp to the image and a range of JPEG quality factors, we compute BER contours at the encoder. Figure 2 shows BER contours for various $\{\beta, f\}$. The horizontal and vertical axis are $\{\beta, f\}$ respectively.

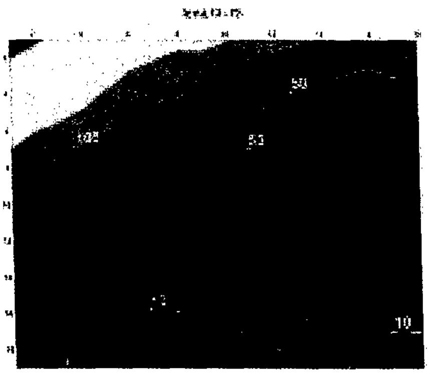


Figure 2- *BER contours for JPEG quality factors 75. The numbers indicate bit errors out of 1024 embedded bits.*

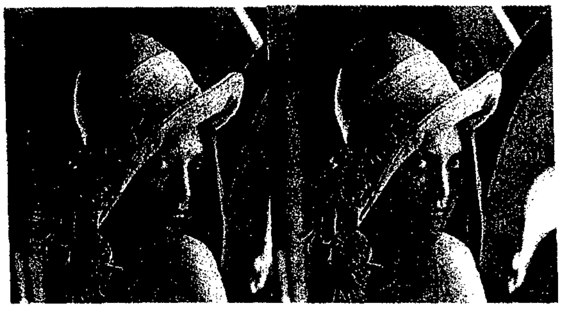


Figure 3- *Two watermarked images using chirp(left) and spread spectrum(right). Both carry 1024 bits at PSNR=40dB.*

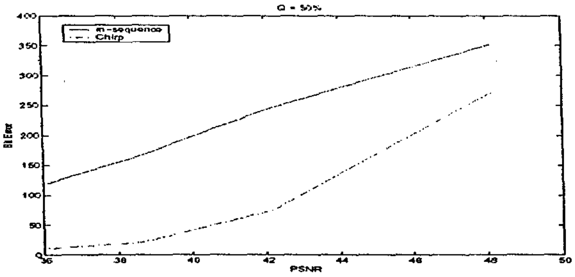


Figure 4-*BER vs PSNR for JPEG quality factor=50. Chirp outperforms m-sequences.*

BER contours can be used to pick $\{\beta, f\}$ pairs that meet specific BER requirements. Note that many different chirps can be used to achieve the same BER requirement. This freedom allows for using different chirps in the image for security purposes and that nevertheless provide the same BER. Figure 3 shows two watermarked images, both with acceptable quality. Figure 4 shows BER performance for chirp and spread spectrum watermarking respectively. Table 1 shows specific numbers drawn from another plot not shown here.

Table 1- Chirp spreading function outperforms spread spectrum across JPEG quality factors.

Bit Error	no compression	$Q = 75\%$	$Q = 50\%$
chirp	14	14	21
m-sequence	16	112	166

6. CONCLUSIONS

We have proposed using 2D chirp functions for image watermarking. The flexibility provided by the chirp allows for tuning of the watermark in ways that has not been previously available. Performance advantages over spread spectrum technique has been demonstrated.

7. REFERENCES

[1] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673–1687, Dec. 1997.

[2] J.R. Hernandez, M. Amando and F. Perez-Gonzalez, "DCT-domain watermarking techniques for still images: detector performance analysis and a new structure," *IEEE Trans. Image Processing*, vol. 9, pp. 55-68, Jan. 2000.

[3] F. Hartung and B. Girod, "Digital watermarking of uncompressed and compressed video," *Signal Process.*, vol. 66, pp. 283–301, May 1998.

[4] M. Kutter and S. Winkler, "A vision-based masking model for spread-spectrum image watermarking, "*IEEE Trans. Image Processing,* vol. 11, pp. 16-25, Jan. 2002.

[5] C. Honsinger and M. Rabbani, " Data embedding using phase dispersion," Eastman Kodak, 2000.

[6] A. Papandreou-Suppappola, *Applicartons in Time-Frequency Signal Processing*, CRC Press, 2003.

[7] X. Xia, "Discrete chirp-Fourier transform and its application to chirp rate estimation," *IEEE Transactions on Signal processing,* pp. 3122-3133, November 2000.

[8] S. Stankovic, I. Djurovic and I. Pitas, "Watermarking in the space/spatial-frequency domain using two-dimensional Radon–Wigner distribution," *IEEE Trans. Image Processing*, vol. 10, pp. 650-658, April 2001.

[9] F. Pérez-González, F. Balado, and Juan R. Hernández Martin, "Performance analysis of existing and new methods for data hiding with known-host information in additive channels," *IEEE Trans. Image Processing*, vol. 51, pp. 960–980, April 2003.

[10] I. J. Cox, Matthew L. Miller, and Andrew L. McKellips, "Watermarking as communications with side information," *Proc. IEEE*, vol. 87, no. 7, pp. 1127–1141, Jul. 1999.

[11] Max H. M. Costa, "Writing on dirty paper," *IEEE Trans. Information Theory*, vol. IT-29, no. 3, pp. 439–441, May 1983.

[12] B. G. Mobasseri, "A spatial video watermark that survives MPEG," *IEEE International Conference on Information Technology: Coding and Computing, Las Vegas, March 27-29, 2000.*

Digital Watermarking Using Two-Dimensional FM Waveforms

Yimin Zhang, Behzad Mohammadi Dogahe, Bijan Mobasseri, and Moeness G. Amin
Center for Advanced Communications
Villanova University, Villanova, PA 19085, USA

ABSTRACT

In this paper, we propose two-dimensional (2-D) frequency modulated (FM) signals for digital watermarking. The hidden information is embedded into an image using the binary phase information of a 2-D FM prototype. The original image is properly partitioned into several blocks. In each block, a 2-D FM watermark waveform is used and the watermark information is embedded using the binary phase. The parameters of the FM watermark are selected in order to achieve low bit error rate (BER) detection of the watermark. Detailed study of performance analysis and parameter optimizations is performed for 2-D chirp signals as an example of 2-D FM waveforms. Experimental results compare the proposed methods and support their effectiveness.

1. INTRODUCTION

Digital watermarking is the process of securely embedding invisible signatures within a cover media with no perceptual impact. Depending on the application, this process is also referred to as data hiding or steganography. Data hiding, if used as a means for covert communications, may require a heavier embedding capacity than digital watermarking. The cover media is of primary interest in watermarking whereas in data hiding the cover media is only useful to the extent that it provides a container for hidden communications. Developing any new watermarking algorithm requires the definition of five components, i.e., 1) cover media, 2) watermark, 3) embedding and extraction, 4) perceptual metric, and 5) resilience and security criteria.

Watermarking has been performed in spectral as well as spatial domain. Arguably, the best known watermarking technique is spread spectrum (SS). SS watermarking has been used in several different contexts. One of the earliest references to SS watermarking is due to Cox.[1] The watermark is drawn from a Gaussian source and additively modifies the full frame discrete Fourier transform (DFT) transform coefficients of the image. Watermarked portions of the DFT consist of the perceptually significant transform coefficients. Hernandez et. al[2] have applied the same idea to 8 × 8 block discrete cosine transform (DCT) transform of images, closely following the JPEG standard. First, the watermark is mapped to a one-dimensional (1-D) binary vector. Then, a 2-D binary mask is generated by an expansion process by repeating each bit of the 1-D vector in different subsets of DCT coefficients. The strength of watermark is driven by a perceptual mask. This approach is similar to spatial domain SS watermarking proposed earlier by Hartung and Girod[3] which did not resort to masking models.

The concept of SS can be applied equally in spectral as well as spatial domains. Case in point is the watermarking model proposed by Kutter and Winkler.[4] The watermark consists of a binary array. Each bit of the array is spread by a 2-D modulation function and added to nonoverlapping sets of image pixels driven by a density metric. This is similar to Honsinger and Rabbani's phase dispersion method.[5] Both Kutter and Honsinger use the same model to spread watermark bits. The former uses a pseudo-random (PN) sequence for the job whereas the latter designs a carrier with flat spectrum but PN phase. The amplitude of the carrier is driven by a perceptual masking profile. SS watermarking model, therefore, is an exercise in selecting the optimum spreading function for a given task.

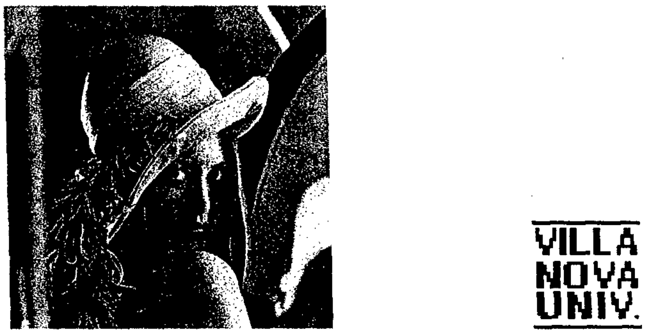
PN spreading sequences are among the earliest spreading functions used in digital watermarking. Although PN sequences provide respectable robustness against malicious attacks through the processing gain, they provide little in terms of spectral shaping. The ability to spectrally shape the watermark allows us to design the spreading function with as little overlap as possible with image data. As importantly, spectral shaping allows for circumventing compression in general and JPEG in particular. Since JPEG compression profile is already known, it is possible to shape the watermark in order to avoid frequency-selective JPEG compression.

In this work we propose a 2-D wideband signal as our choice for spreading function and implement a block-based digital watermarking algorithm. We then evaluate the performance of this function for a special case of 2-D chirp signals. In a prior work, Stankovic et al[6] used chirps as digital watermarks too. They added a chirp to the entire frame then used energy-concentrating property of Radon-Wigner transform to establish the presence of the watermark by peak-searching. This algorithm is best suited to copyright and ownership verification applications where a binary decision is sufficient to establish the presence or absence of the watermark. The ability to embed and detect different chirps per image block, however, allows for data hiding applications where the extracted watermark may be an information-bearing bitstream. Another point of departure from[6] is exploitation of known-host-state-methods.[7] This approach was first suggested by Cox as communication problem with side-information[8] which was in turn based on Costa's dirty paper writing.[9] We have incorporated this idea into our work and show that it is possible to achieve zero BER by exploiting knowledge of host signal at the encoder. This observation is in contrast to SS watermarking and others where watermark structure is unrelated to host signal statistics.

2. WATERMARK EMBEDDING

Consider a problem that a digital watermark containing N-bit information is to be embedded in a gray-scale image. The image is partitioned into non-overlapping blocks whose sizes depend on the picture size and the amount information to hide. If N_p is the number of bits that a block can host, then $\lceil N/N_p \rceil$ blocks are needed, where $\lceil x \rceil$ denotes the minimum integer equal to or larger than x. In addition, when JPEG image compression is considered, it is preferred that the size of each block be 8 × 8 or its multiples.

As an example, we consider in this paper a 32 × 32 binary seal, which is the name of the authors' affiliation, to be embedded into the 512 × 512 gray-scale Lena picture (Fig. 1). If the watermark is embedded through binary phase modulations, each block hosts one bit of information. Therefore, we can partition the image into 32 × 32 = 1024 blocks, with each block consisting of 16 × 16 = 256 pixels.



(a) Gray-scale Lena image (512 × 512). (b) Binary seal (32 × 32).

Figure 1. Original image and binary seal.

In each block (m, n), $m, n = 0, ..., 31$, the watermarked image $G(x, y)$ is expressed as

$$G(m,n,x,y) = I(m,n,x,y) + Q\{k\text{Re}[s(m,n)W(x,y,\Theta_0)]\} \tag{1}$$

where $I(m,n,x,y)$ is the original image at block (m,n), extending over the spatial axes, x and y, where $x,\ y = 0, \cdots, 15$. In (1), $W(x,y,\Theta_0)$ represents the employed complex 2-D FM waveform basis with Θ_0 representing a set of parameters that defines the 2-D FM waveform basis, and $s(m,n)$ is the information to be mapped into the 2-D waveform basis in block (m, n). When binary phase data modulation is used, $s(m, n)$ takes value of either $+1$ or -1, corresponding to either 0 (black) or 1 (white) of the seal pixels. The parameter k is introduced to control the image-to-watermark ratio, which is usually referred to as the peak signal-to-noise ratio (PSNR). Moreover, $\mathrm{Re}[\cdot]$ denotes the real-part operator, emphasizing the fact that while the original 2-D FM basis waveform is complex, the hidden information in the image is real. $Q[x] = \lfloor x + 0.5 \rfloor$ denotes quantization operation, where $\lfloor \cdot \rfloor$ stands for rounding down to the nearest integer.

3. WATERMARK DETECTION AND RECOVERY

3.1. Watermark Detection and Parameter Estimation

We consider blind decoding of the watermarked image, that is, the unmarked image is not used in the detection. When the parameters Θ_0 that define the 2-D FM basis waveform are not available at the detector, they must be estimated before detection can be made. On the other hand, when the waveform parameters are known at the detector, this step can be skipped.

The detector first estimates $\hat{\Theta}_0$ by maximizing the following criterion,

$$\hat{\Theta}_0 = \arg\max_{\Theta} |C(m, n, \Theta)|, \tag{2}$$

where

$$\begin{aligned} C(m,n,\Theta) &= \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} G(m,n,x,y)W^*(x,y,\Theta) \\ &= \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} I(m,n,x,y)W^*(x,y,\Theta) + \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} Q\{k\mathrm{Re}[s(m,n)W(x,y,\Theta_0)]\}W^*(x,y,\Theta) \\ &= C_I(m,n,\Theta) + C_W(m,n,\Theta) \end{aligned} \tag{3}$$

In (3),

$$C_I(m,n,\Theta) = \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} I(m,n,x,y)W^*(x,y,\Theta)$$

is the output of the matched filter corresponding to the original image, whereas

$$C_W(m,n,\Theta) = \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} Q\{k\mathrm{Re}[s(m,n)W(x,y,\Theta_0)]\}W^*(x,y,\Theta)$$

is the output corresponding to the watermark.

It is noted that, unlike the conventional communications where the data is often zero-mean, the image in its original format is all non-negative. To avoid any potential bias in the detection, therefore, it is important to remove the DC component from the image before he watermark detection, and it is desirable that the waveform basis is designed to be zero-mean.

Because of the different signatures between the image and the watermark waveform basis when they are projected into the Θ domain, the waveform basis achieves much higher gain through the matched filtering at the detection. When the watermark has enough energy such that $|C_W(m,n,\Theta_0)| > |C_I(m,n,\Theta_0)|$, the waveform parameters Θ_0 may be detected by locating the peak of $C(m,n,\Theta)$.

In practical watermarking applications, however, the low probability of detection is important. For this purpose, the embedded information usually does not have enough strength such that the waveform parameters can be estimated in each partitioned block. When the hidden information is embedded such that

$$|\tilde{C}_W(\Theta_0)| \gg |\tilde{C}_I(\Theta_0)|, \tag{4}$$

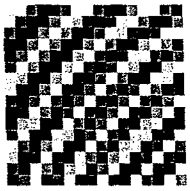


Figure 2. Quantized 2-D chirp waveform.

in the vicinity of the watermark, away from the region where the image energy is concentrated, then the existence of watermark can be detected and reliable estimation of $\hat{\Theta}_0$ can be made by maximizing the following criterion,

$$\hat{\Theta}_0 = \arg\max_{\Theta} \tilde{C}(\Theta), \tag{5}$$

where

$$\tilde{C}(\Theta) = \sum_m \sum_n |C(m,n,\Theta)|, \tag{6}$$

and $\tilde{C}_I(\Theta)$ and $\tilde{C}_W(\Theta)$ are defined similarly.

3.2. Watermark Recovery

When the watermark is detected and waveform parameters Θ_0 are known or reliably estimated, the watermark information at block (m,n) can be recovered from the phase information of the matched filter output, i.e., $C(m,n,\Theta_0)$. In particular, when the binary phase modulation is used, the embedded information is estimated as

$$\hat{s}(m,n) = \begin{cases} 1, & \text{if } \mathrm{Re}[C(m,n,\Theta_0)] \geq 0 \\ 0, & \text{if } \mathrm{Re}[C(m,n,\Theta_0)] < 0. \end{cases} \tag{7}$$

4. WATERMARK WAVEFORM DESIGN

4.1. 2-D Chirp Waveform

In this section, we consider the design of watermark waveforms. A 2-D chirp signal is used as a simple example of 2-D FM basis waveform. The 2-D chirp waveform basis, $W(x,y,\beta_x,\beta_y,f_x,f_y)$, in the complex format, is expressed as[6,10]

$$W(x,y,\beta_y,f_x,f_y) = e^{j\pi(\beta_x x^2+\beta_y y^2)+j2\pi(f_x x+f_y y)}, \tag{8}$$

where β_x and β_y are the chirp rates at the x and y axes, and f_x and f_y are the respective initial frequencies of the chirp signal. These four variables form the waveform parameters, i.e., $\Theta_0 = (\beta_x,\beta_y,f_x,f_y)$. For notational simplicity and without loss of generality, we consider symmetric cases and denote $\beta_x = \beta_y = \beta_0$, and $f_x = f_y = f_0$, and Θ_0 simplifies to $\Theta_0 = (\beta_0, f_0)$. Accordingly, the 2-D chirp basis function becomes

$$W(x,y) = e^{j\pi\beta_0(x^2+y^2)+j2\pi f_0(x+y)}. \tag{9}$$

The variables x and y take values from $[0,...,T-1]$, for $T \times T$ blocks, and in this specific example, $T = 15$. Therefore, the instantaneous frequency in (9) ranges from f_0 to $\beta_0(T-1)+f_0$.

The spectrum of the 2-D chirp waveform is important in the performance of the detection and robustness. In designing the FM waveform, the initial frequency and chirp rate are selected such that, at the specific (β_0, f_0), the projection of the image spectrum is relatively low and the chirp is robust against image compression. For this sake, the high-frequency band is first excluded from consideration and then the parameters are optimized by choosing those where the image spectrum is low.

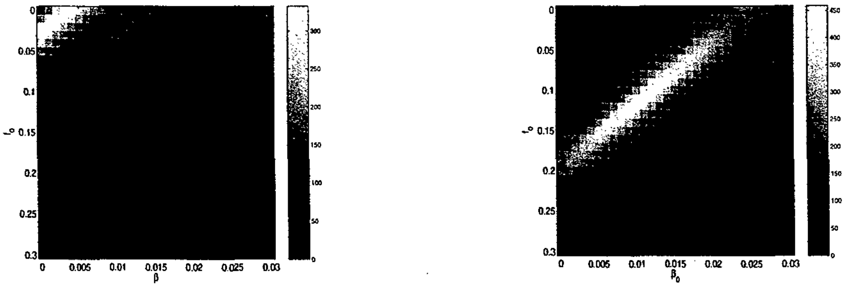


Figure 3. Chirp transform spectra of a picture block $C_I(\beta, f)$ (left) and the 2-D chirp FM waveform $C_W(\beta, f)$ (right).

Figure 2 illustrates the quantized waveform of a 2-D chirp. The chirp transform spectrum of the first (upper-left corner) block of the original Lena image as well as that of the 2-D chirp waveform are shown in Fig. 3, where the PSNR is 40 dB. It is clear that the image has a wide spectrum with its peak power located at low frequencies and low chirp rates. On the other hand, the chirp spectrum is usually designed to be away from the region where the image spectrum is concentrated.

4.2. Chirp Parameter Selection

For the binary phase modulation, the probability of erroneous detection, i.e., embedding information s and deciding in favor of $r \neq s$, is given be

$$P_e = P(r \neq s) = P(s = -1)P(r = +1|s = -1) + \; P(s = +1)P(r = -1|s = +1). \qquad (10)$$

The probability is evaluated for all blocks. The chirp parameter selection, in essence, is to find (β_0, f_0) such that, given an embedding power, the above error probability, i.e., the total error bits divided by the total information bits, can be minimized. It is noted that, although we used the term probability here for convenience, the image information over different blocks is determinant and is known at the embedder. This is in fact the known-host state method that we exploit below.

Therefore, the optimum values of (β_0, f_0) can be selected by searching (β, f) such that the above error probability is minimized. However, an insight look of the decision process can deepen our understanding as well as help us in determining the waveform parameters. In the next, we consider the adaptive chirp power allocation.

4.3. Adaptive Chirp Power Allocation

The known-host state method allows us to embed the watermark in such a way as to push the decision metric into correct decision region. To ensure correct detection at all blocks, the following condition should be satisfied,

$$C_W(m, n, \beta_0, f_0) \overset{>}{\underset{<}{}} - C_I(m, n, \beta_0, f_0), \text{ if } s(m, n) \overset{>}{\underset{<}{}} 0. \qquad (11)$$

The matched filter output of the embedded waveform at block (m, n), $C_W(m, n, \beta_0, f_0)$, takes the form of

$$C_W(m, n, \beta_0, f_0) = s(m, n)H(\beta_0, f_0, k), \qquad (12)$$

where $H(\beta_0, f_0, k) = |C_W(m, n, \beta_0, f_0)|$ is the magnitude. We emphasize that H is a function of k as k constitutes an important part of the chirp waveform design.

Substituting (12) to (11) yields,

$$s(m, n)H(\beta_0, f_0, k) \overset{>}{\underset{<}{}} - C_I(m, n, \beta_0, f_0), \text{if } s(m, n) \overset{>}{\underset{<}{}} 0. \qquad (13)$$

or, equivalently,

$$H(\beta_0, f_0, k) > -s(m,n)C_I(m,n,\beta_0,f_0). \tag{14}$$

Because both $s(m,n)$ and $C_I(m,n,\beta_0,f_0)$ are known to the embedder, we can choose different values of k at different blocks. That is, at each block (m,n), $k(m,n)$ is chosen to be minimal to maintain error free detection. In particular, when $s(m,n)$ and $C_I(m,n,\beta_0,f_0)$ have the same sign, requirement (14) is always satisfied irrespective of the value of k. Therefore, $k = 0$ can be chosen. For this value of k no chirp is actually added to the image block.

It is noted that, however, if the watermark has to be detected blindly, the total watermark energy has to be maintained such that the detection and parameter estimation can be carried out successfully.

4.4. Watermark Encryption Using Random Parameters

Encryption of watermark information is essential to protect it from possible interception, alteration, or removal. We use a set of random, rather than constant, (β_0, f_0) parameters, over the different blocks. The random (β_0, f_0) parameters are uniformly chosen among a predesigned region which provides low BER. Therefore, the generation formulation of the random numbers as well as the initial state act as the key. For a given generation formulation, the initial values can be optimized for BER reduction.

The use of random (β_0, f_0) parameters has two advantages. First, it reduces the energy at any (β_0, f_0) and, therefore, reduces the detectability of the watermark by unauthorized users. Second, even when the existence of watermark is detected, the watermark information can not be detected and is difficult to remove and alter.

5. CHIRP PARAMETERS OPTIMIZATION FOR JPEG COMPRESSED IMAGES

A common signal processing operation on most images is compression based on JPEG. The question to be answered here is the extent to which BER is affected by varying levels of compression, and more importantly, the choice of $\{\beta_0, f_0\}$ to make the watermark robust to compression. Rewrite (1) for the underlying chirp signal case as

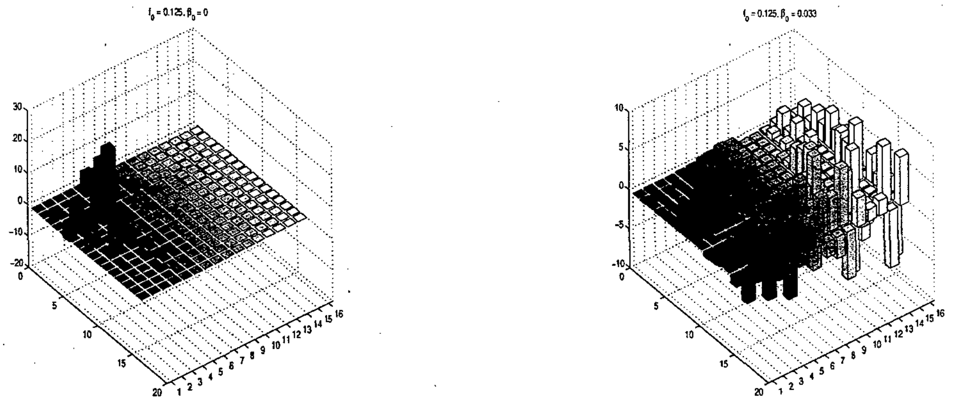
$$G(m,n,x,y) \quad = I(m,n,x,y) + \mathcal{Q}\{k\mathrm{Re}[s(m,n)W(x,y,\beta_0,f_0)]\}. \tag{15}$$

Baseline JPEG consists of 4 sequential steps: (a) block DCT; (b) quantization; (c) zigzag scan; and (d) entropy coding. The goal here is to spectrally shape the chirp to make it most robust to JPEG for a given quality factor Q (Note that, while the same notation is used, the quality factor Q differs to the JPEG quantization matrix $\mathcal{Q}$ defined earlier).

Fig. 4 shows how chirp energy distribution can be changed to counter JPEG quantization matrix. When $\beta_0 = 0$, the watermark is sinusoidal, and the energy is localized in the DCT domain. On the other hand, when $\beta = 0.033$ which is relatively large, the energy is distributed in the DCT domain, particularly in the high frequency regions.

These figures illustrate how chirp's spectrum is modified by different choices of $\{\beta_0, f_0\}$. The distribution of DCT coefficients should be closely matched to the quantization matrix to produce the lowest BER. Since quantization matrix tends to compress higher frequency bands more aggressively, compression affects chirps with higher frequency contents more. The advantage of using a chirp versus sinusoid is clearly demonstrated here. Sinusoid's energy is concentrated at specific frequency bands and can be easily removed by selective filtering or compression. In addition, there are virtually no degrees of freedom to spread the spectrum and optimize detection for varying JPEG Q factors.

The question of chirp survival after JPEG cannot be discerned solely by observing chirp DCT since the quantizer operates on the DCT of the image plus chirp and not the DCT coefficients individually. Denote $G(m',n')$ as the (m',n')th block of the watermark where each block is of size 8 × 8 to match the JPEG compression standard, and let $\mathcal{G}(m',n') = \mathrm{dct}[G(m',n')]$. We also define $\mathcal{I}(m',n') = \mathrm{dct}[I(m',n')]$ and $\mathcal{W}(m',n',\beta_0,f_0) = \mathrm{dct}[\tilde{W}(m',n',\beta_0,f_0)]$ in a similar way, where $\tilde{W}(m',n',\beta_0,f_0)$ is the watermark defined



(a) 16 × 16 DCT transform (β_0=0, f_0=0.125)

(b) 16 × 16 DCT transform (β_0=0.033, f_0=0.125)

Figure 4. Block DCT of a chirp for different $\{\beta_0, f_0\}$.

at the (m', n')th block with chirp parameters (β_0, f_0). Then, at the (m', n')th block, the quantized DCT coefficients are given by

$$\mathcal{Q}\left(\frac{\mathcal{G}(m', n')}{Q}\right) = \mathcal{Q}\left(\frac{\mathcal{I}(m', n')}{Q} + \frac{\mathcal{W}(m', n', \beta_0, f_0)}{Q}\right) \tag{16}$$

where $Q = [q_{i,j}]$ is JPEG quantization matrix, $i, j = 0, ..., 7$. Note that division in (16) is an element-by-element matrix division. The decoder then performs an inverse quantization on (16) followed by inverse DCT to obtain $\hat{G}(m', n')$,

$$\hat{G}(m, n) = \mathrm{dct}^{-1}\left[Q \cdot \mathcal{Q}\left(\frac{\mathcal{I}(m', n')}{Q} + \frac{\mathcal{W}(m', n', \beta_0, f_0))}{Q}\right)\right]. \tag{17}$$

It is clear that the watermark will be eliminated unless the chirp has enough strength to survive compression and decompression cycle.

6. SIMULATION RESULTS

6.1. Chirp Parameter Selection

We first consider the application of a single-component 2-D chirp watermark. To understand the effect of different chirp parameters to the watermark detection performance, the BER performance is shown in Fig. 5 versus the chirp rate β_0, where the initial frequency is fixed to $f_0 = 0.2333$. It is clear that, when JPEG compression is not applied, a high value of β_0 tends to provide low BER performance, because the picture does not have much power at the high-frequency band. With moderate JPEG compression, however, there exists an optimum range of β_0, because the high-frequency band will be suppressed in the process of image compression.

The next example shows the sensitivity of $|C_I(m, n, \beta_0, f_0)|$, i.e., the magnitude of the matched filter output of the original image, to the values of (β_0, f_0). Two different sets of chirp parameters, that is, $(\beta_0, f_0) = (0.008, 0.08)$ and $(\beta_0, f_0) = (0.011, 0.11)$, are considered and compared. Fig. 6 shows their histogram over the 1024 blocks, whereas the corresponding cumulative distribution plots are shown in Fig. 7. The energy of the matched filter output of the watermark, corresponding to PSNR = 35, 40, and 45dB, are depicted in the figures. For the watermark chirp (0.008, 0.08), there are 25, 81, and 175 image blocks with a projection magnitude value more than the energy of the watermark at PSNR = 35, 40, and 45dB, respectively, whereas using the other watermark chirp, there are 8, 30, and 76 blocks having projection magnitude value more than the watermark's energy at the corresponding PSNR. Clearly, at the specific value of PSNR, $(\beta_0, f_0) = (0.011, 0.11)$ is preferable in this case, since the chance is lower for the image to influence the detection of the sign of s.

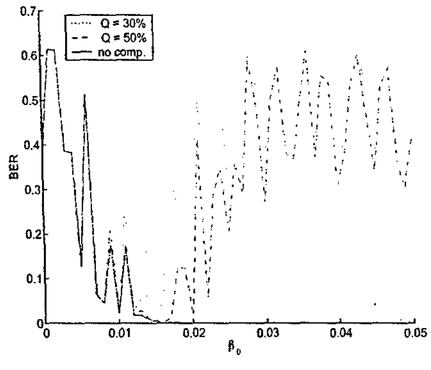
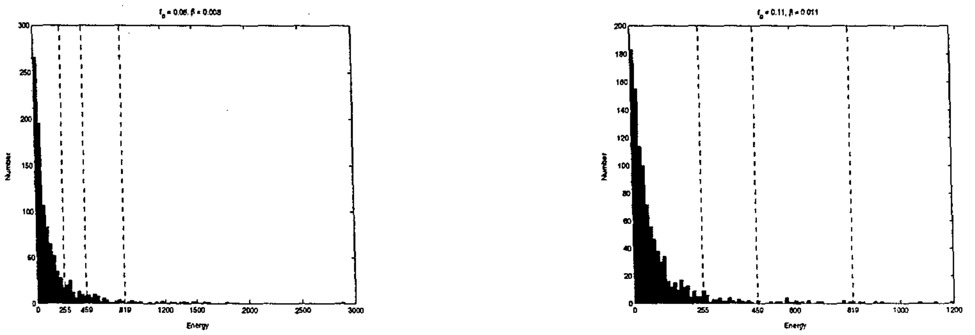
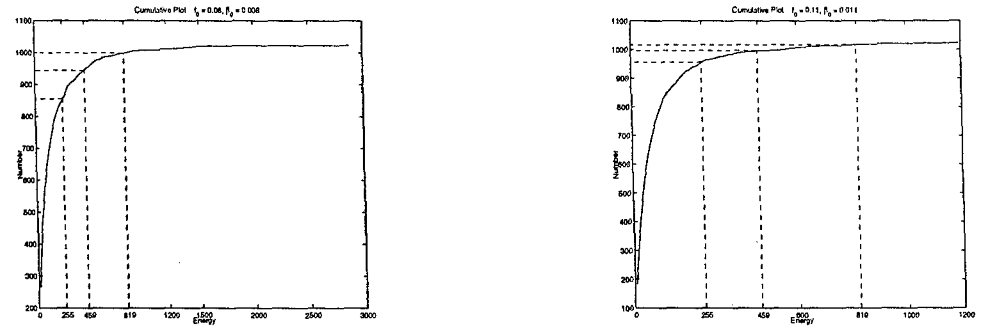


Figure 5. BER versus chirp rate (PSNR = 35dB).



(a) $\beta_0 = 0.008, f_0 = 0.08.$ (b) $\beta_0 = 0.011, f_0 = 0.11.$

Figure 6. Histogram of the matched filter output of the original image



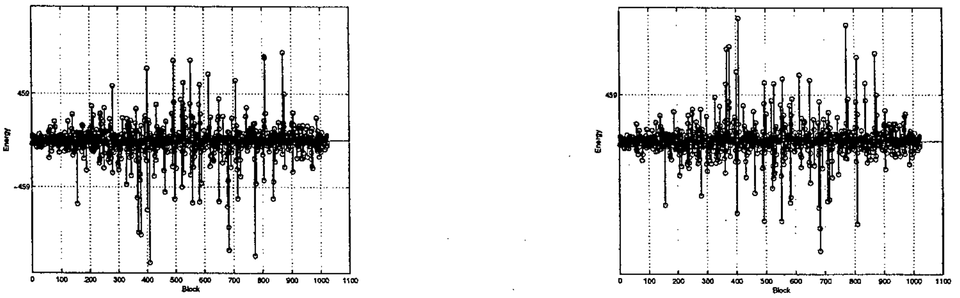
(a) $\beta_0 = 0.008, f_0 = 0.08.$ (b) $\beta_0 = 0.011, f_0 = 0.11.$

Figure 7. Cumulative histogram of the matched filter output of the original image

6.2. Adaptive Watermark Power Allocation

It is evident from the previous example that, the required watermark level to assure correction detection is different for different blocks. To find the minimum watermark energy for each block, we plot in Fig. 8(a) the matched filter output of the original image in a sequential order. The dashed lines show the levels corresponding to the watermark at PSNR=40dB. At those blocks where the magnitude of the image contribution exceeds the watermark output, there is a possibility that the watermark information bit is wrongly decided. However, whether it occurs or not depends on the sign of the embedded information.

To incorporate the watermark information, therefore, we plot in Fig. 8(b) the result of $-s(m,n)C_I(m,n,\beta_0,f_0)$. A watermark decision error will occur in each of the blocks where this value is higher than the watermark output magnitude. In other words, we can choose the watermark energy at each block such that it merely exceeds $-s(m,n)C_I(m,n,\beta_0,f_0)$. As such, the watermark energy is minimized whereas low error-rate watermark embedding is assured. The PSNR required to achieve BER $= 0$ is 51dB.



(a) $C_I(m,n,\beta_0,f_0)$

(b) $-s(m,n)C_I(m,n,\beta_0,f_0)$

Figure 8. Matched filter output of the original image. Dashed line corresponds to watermark output at (PSNR $=$ 40 dB)

6.3. Watermark Detection Capability

Detection capability is another important issue which should be considered in the watermarking. For blind watermark detection, the watermark should have enough strength so that its existence and chirp parameters can be estimated. Fig. 9 shows the sum of chirp transform spectra of all blocks, $|\tilde{C}(\Theta)|$, for both cases of with and without watermark, where PSNR $=$ 40 dB, and the watermark chirp parameters are $\beta_0 = 0.011$ and $f_0 = 0.11$. It is clear from this figure that the watermark can be detected and its parameters can be estimated.

It is mentioned that, when adaptive watermark power allocation technique, implemented in the previous subsection, is applied, the watermark cannot be detected. Therefore, adaptive watermark power allocation should be used for non-blind watermark processing or increased watermark energy should be used at those blocks of low watermark energy.

6.4. Effect of DC Components

The previous simulation results have assumed that the DC components of the watermarked image and the watermark waveform are removed, as we discussed in Section 3. The next example shows that, if such DC component removal is not properly performed, there may be a significant bias which, in turn, will affect the watermark detection.

Figure 10(a) shows the histograms of $C(m,n,\beta_0,f_0)$, the matched filter output of the watermarked image. The chirp parameters are $(\beta_0,f_0) = (0.009, 0.09)$ and the PSNR is 40dB. The histograms corresponding to watermark information $+1$ and -1 are overlapped in this plot. It is obvious that the high bias observed in this figure will make the watermark decision very difficult. When the DC components are removed, however, as shown in Fig. 10(b), no bias is observed.

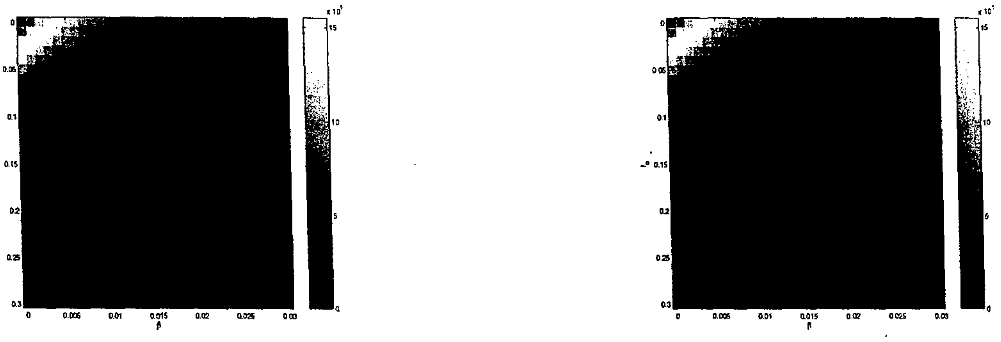
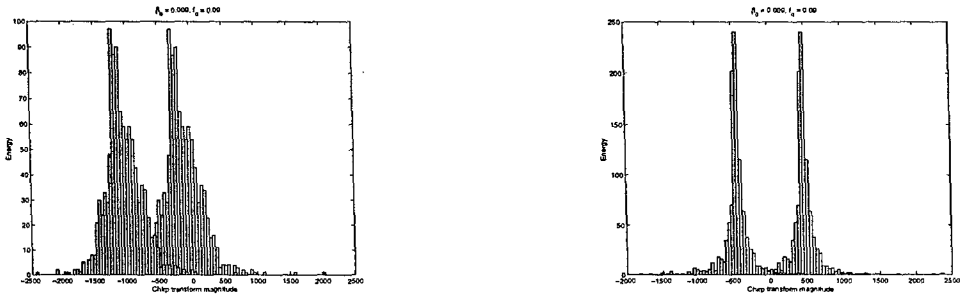


Figure 9. (a) Sum of chirp transform spectra of all blocks of Lena picture and (b) watermarked Lena picture.



(a) DC components not removed.

(b) DC components removed.

Figure 10. Histograms of $C(m, n, \beta_0, f_0)$ with $s(m, n) = +1$ and -1 embedded ($\beta_0 = 0.009, f_0 = 0.09$)

6.5. Watermark Encryption Using Random Parameters

Fig. 11(a) shows the bit error plot for detecting an embedded watermark in Lena image, where PSNR = 40dB and the JPEG quality factor is $Q = 50\%$. The $\beta_0, f_0)$ parameters are uniformly distributed over the region between the two zigzag patterns. Fig. 11(b) shows the sum of chirp transform spectra of all blocks of the watermarked Lena image. Comparing this figure with Fig. 9(b), the advantage of using the encryption is clear. There is no detectable trace of watermark in this figure because the watermark energy is distributed over a number of different (β_0, f_0) combinations. Table 1 compares the numbers of bit errors for chirps with fixed ($\beta_0 = 0.011$, $f_0 = 0.11$) and encrypted parameters. The BER for the encrypted watermark is slightly worse than that of the fixed watermark parameter case because of the use of diversified (β_0, f_0) parameters.

Table 1. Bit error comparison

Bit Error	no compression	$Q = 75\%$	$Q = 50\%$
chirp with fixed parameters	14	19	39
chirp with encrypted parameters	20	25	49

6.6. JPEG Compressed Imagery

Fig. 12 shows contours of the BERs for the Lena image, where the numbers denote the number of erroneously decided bits out of the 1024 total watermarked bits. For comparison, we also show the same results for the Elaine picture in Fig. 13.

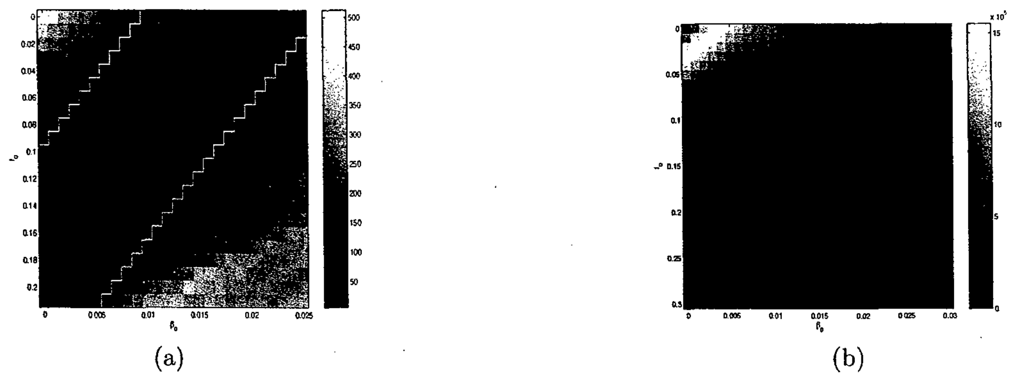


Figure 11. (a) (β_0, f_0) pairs that have been used in embedding (b) Sum of chirp transform spectra of all blocks of watermarked Lena picture using encryption.

It is observed from these figures that, for higher compression factors, the low BERs appear in lower (β_0, f_0) regions. This behavior is consistent with JPEG compression as more high frequency components are suppressed, along with the watermark. Similar simulations have been carried out for Elaine. Trends are similar but variations of BER vs. (β_0, f_0) are clearly image-dependent. It is often desired to select (β_0, f_0) pairs that survive compression across a range of Q factors. Fig. 12 can be used to identify overlapping portions of (β, f) to achieve certain BERs. It is interesting to note that for Elaine, it is possible to select chirps that meet BER < 0.01 across all $Q \geq 50\%$. The same cannot be said for Lena.

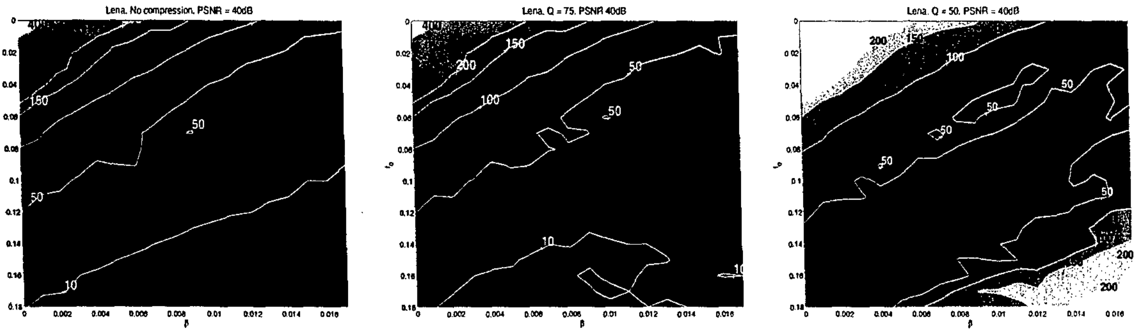
An inspection of BER contours reveals another important property of the chirp. As shown in Fig. 13(a), to achieve BERs below 0.01 using sinusoids allows the use of only a limited number of frequencies below $f_0 = 0.143$. Using a chirp instead greatly expands the possible (β_0, f_0) pairs that achieve the required BER. The expanded choice is important in tuning the watermark in order to counter compression effects. For example, in Lena image there are no sinusoids that achieve BER < 0.01 for $Q = 75\%$, whereas there are plenty of chirps that would achieve this specified BER.

7. CONCLUSIONS

In this paper, a novel method has been proposed for digital watermark embedding and detection. The watermarking is based on 2-D FM waveforms which can be designed for flexible spectrum allocation, so that the parameters can be optimized to distinguish itself from the original image for improved watermark detectability, and to avoid high frequencies so that the robustness against compression attacks can be maintained. In particular, we used 2-D chirp signals as examples for extensive investigation. The adaptive chirp power allocation technique improves the performance as well as the imperceptibility of the watermarked image. The advantage of watermark encryption using random chirp parameters have also demonstrated.

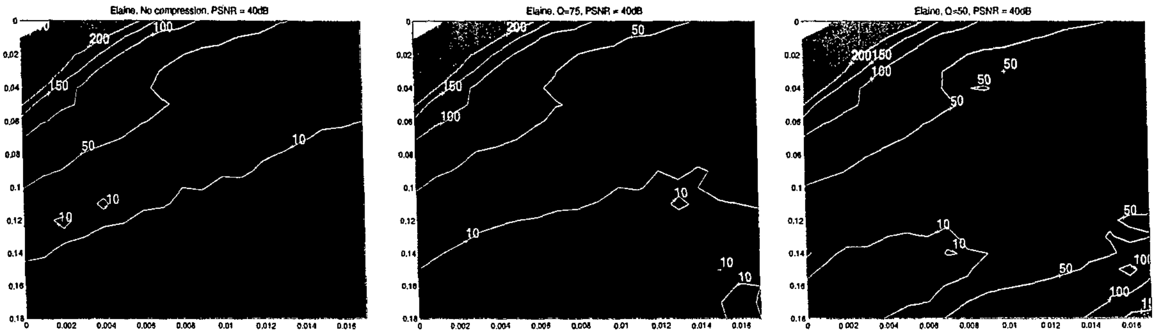
REFERENCES

1. I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multemedia," *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.
2. J. R. Hernandez, M. Amando and F. Perez-Gonzalez, "DCT-domain watermarking techniques for still images: detector performance analysis and a new structure," *IEEE Trans. Image Processing*, vol. 9, pp. 55–68, Jan. 2000.
3. F. Hartung and B. Girod, "Digital watermarking of uncompressed and compressed video," *Signal Process.*, vol. 66, pp. 283–301, May 1998.
4. M. Kutter and S. Winkler, "A vision-based masking model for spread-spectrum image watermarking," *IEEE Trans. Image Processing*, vol. 11, pp. 16–25, Jan. 2002.



(a) No compression. (b) With compression ($Q = 75\%$). (c) With compression ($Q = 50\%$).

Figure 12. BER contour for Lena.



(a) No compression. (b) With compression ($Q = 75\%$). (c) With compression ($Q = 50\%$).

Figure 13. BER contour for Elaine.

5. C. Honsinger and M. Rabbani, "Data embedding using phase dispersion," *Eastman Kodak*, 2000.
6. S. Stanković, I. Djurović, and I. Pitas, "Watermarking in the space/spatial-frequency domain using two-deimentional radon-wigner distribution," *IEEE Trans. Image Processing*, vol. 10, pp. 650–658, Apr. 2001.
7. F. Pérez-González, F. Balado, and J. R. H. Martin, "Performance analysis of existing and new methods for data hiding with known-host information in additive channels," *IEEE Trans. Image Processing*, vol. 51, pp. 960–980, April 2003.
8. I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," *Proc. IEEE*, vol. 87, no. 7, pp. 1127–1141, July 1999.
9. M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 439–441, May 1983.
10. L. Cohen, *Time-Frequency Analysis*. Upper Saddle River, NJ: Prentice Hall, 1995.
11. G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking digital image and video data," *IEEE Signal Processing Magazine*, Sep. 2000.
12. C. I. Podilchuk, and E. J. Delp, "Digital watermarking: algorithms and applications," *IEEE Signal Processing Magazine*, July 2001.
13. H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform: with Applications in Optics and Signal Processing*. New York, NY: John Wiley, 2001.
14. V. Katkovnik, "Discrete-time local polynomial approximation of the instantaneous frequency," *IEEE Trans. Signal Processing*, vol. 46, no. 10 , pp. 2626–2637, Oct. 1998.

Rotationally Robust Watermarking Using Tunable Chirps

Electrical and Computer Engineering Department and
Center for Advanced Communications
Villanova University

By
Christopher E. Fleming

Under the Direction of Dr. Bijan G. Mobasseri

Table of Contents

List of Figures

List of Tables

1. Introduction

The Chirp watermarking algorithm is notable for its ability to avoid the effects of JPEG compression [1]. However, it is still vulnerable to geometric attacks such as rotation, scaling and translation (RST). RST is quite common and found in most digital image software packages. These three image-processing tools are discussed because of their negative effects on the chirp's detection. Because the watermark is implemented in the spatial domain, it is directly susceptible to these attacks.

Rotation and Translation were implemented to the chirp's detection algorithm using MATLAB. Scaling was not examined further because previous research showed the chirp was reasonably resilient to scaling [1]. The experiment was carried out using a smaller viewing portion of the image. This "Viewing Window" was synchronized in multiples of the block size to eliminate extra variants in the tests. An example of the viewing window can be seen in the figure below.

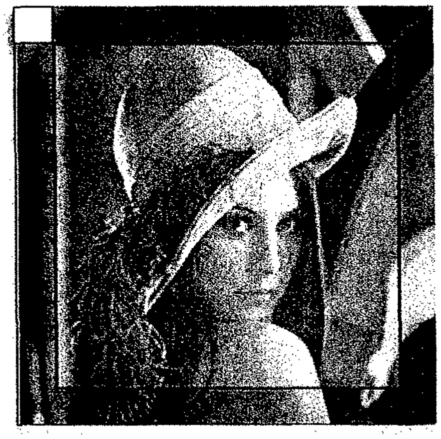


Figure 1: View Window

The results of these tests illustrated the chirp's inability to withstand both rotation and translation. Even though the effects of translation were destructive to the detector, in comparison, rotation's effect on the detector was worse. Therefore the remainder of this paper will focus on eliminating rotation's influence on the chirp. Figure 2 displays the bit error rate (BER) curve for rotation.

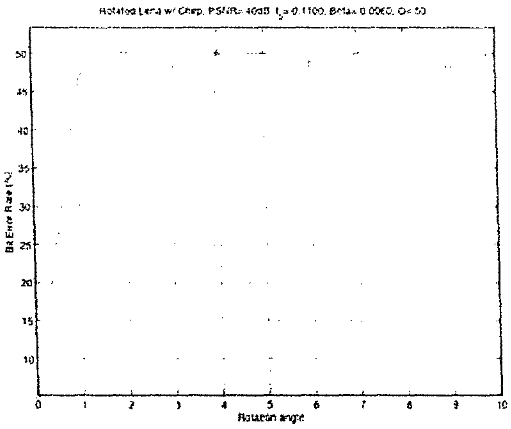


Figure 2: Rotation BER plots

The effects of rotation are troublesome with a 43% BER for only one degree of rotation. This is especially worrisome since detection is unpredictable when the BER is 50%. Statistically the results are poor, and the visual results of the effects of rotation are not even noticeable. The effects of rotation are displayed in the figures below.

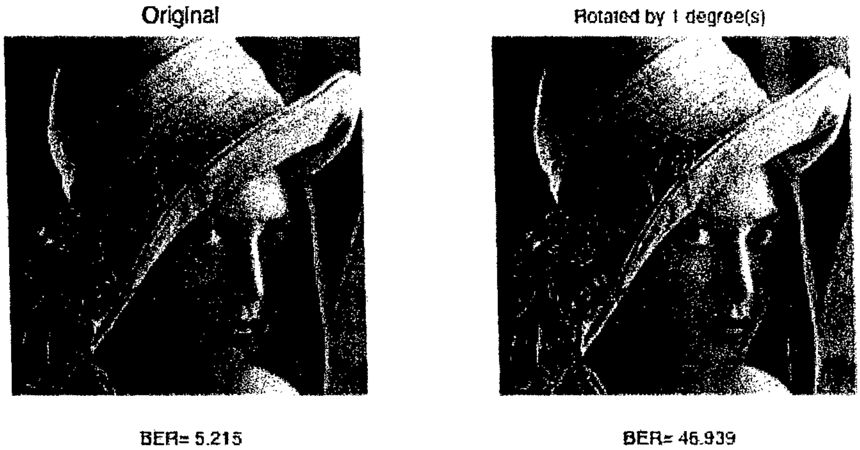


Figure 3: Visually Effects of Rotation

The distinctions between the two images are subtle even though one has been rotated. Even more troublesome, is the 47 percent increase in BER for only one degree of rotation. The vulnerability of this watermark against rotation is quite evident.

Rotations involved in this project will only range from 0-15 degrees, because large rotations are visually noticeable and can be corrected in most images. The following will discuss the use of an rotation-invariant domain and a circular watermark to add rotational resilience to the chirp-watermarking scheme.

2. Fourier-Mellin Transform

Rotation's effect at the chirp watermark detector is destructive, but algorithms using rotation-invariant domains can address them. The invariant-domain method used in this paper is the Fourier-Mellin Transform (FMT) [6]. This transform is known for it's ability to be invariant to RST, and its ability to damage image quality with interpolation [7]. The FMT can be broken down into two complex problems: the Log-Polar Mapping (LPM) and the complex modulus of the Discrete-Fourier Transform (DFT).

2.1. LPM

LPM is a process where a linear Cartesian coordinate system is converted into a logarithmic polar coordinate system. The variable, 'n', is equivalent to length pixels on the on side of the image. For example, a 512x512 image would have a 'n' value of 512. "N" then helps determine the center points (X_{center}, Y_{center}), the logarithmic scale (ρ), and angular scale (θ). The x_{lp} and y_{lp} are the mapping positions that the Cartesian coordinates (x,y) are interpolated too. This process is described in the equations below [6][8].

$$\begin{aligned}
&d = (n-1)/2, \quad \textit{even number of pixels} \\
&X_{center} = Y_{center} = (n+1)/2 \\
&\rho = \{0 \ldots \log_{10}(d)\} \\
&\theta = \{0 \ldots 2\pi\} \\
&x_{lp} = e^{\rho} \cdot \cos(\theta) \\
&y_{lp} = e^{\rho} \cdot \sin(\theta)
\end{aligned}$$

Equation 1: LPM Procedure

LPM is used in the FMT because it turns scaling and more importantly rotation into translation. Therefore a clockwise (cw) rotation will be a shift right, and a counter-clockwise (ccw) rotation will be a shift left. The shift in the LPM after rotation can be seen in figure 4.

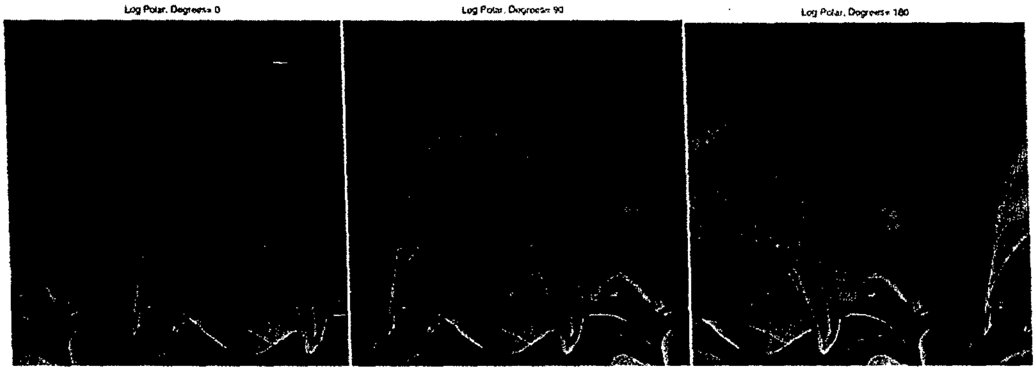


Figure 4: LPM Representations of Lena

The figures also show the image distortions mentioned earlier by the nonlinear operations of the LPM. This translation produced by the LPM allows the image and the chirp to become rotationally invariant with the Discrete Fourier Transform.

2.2. Discrete-Fourier Transform

The Discrete Fourier Transform (DFT) and its properties are one of the most widely known mathematical principals in signal processing. One of the properties of the DFT is the delay property. In this property a delay in time or a shift in space is converted into a shift in phase. Both the DFT and the delay property can be seen in the equations below.

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\cdot e^{jk\frac{2\pi n}{N}}, \quad DFT$$

$$x[n-k] = X(w)\cdot e^{-jwk}, \quad Delay\ property$$

Equation 2: DFT and Delay Property

Applying this equation with the complex modulus or absolute value removes the phase from the equation; making the image invariant to translation in the LPM. The original image data is therefore invariant to rotation and scaling, because translation in the Log-Polar domain is equivalent to scaling and rotation.

$$x[n-k] = X(w)\cdot e^{-jwk} \Rightarrow \quad \left|X(w)\cdot e^{-jwk}\right| = \left|X(w)\right|$$

Equation 3: Complex Modulus of the DFT

Previous algorithms used the original unmarked image to successfully detect a signal after RST using the FMT [6][8][9]. Even though it was effective, it is not possible to maintain this same algorithm for the chirp, because it is a blind watermarking scheme that is embedded spatially and the reverse LPM's distortion are too severe. An example of this is seen in the figure below.

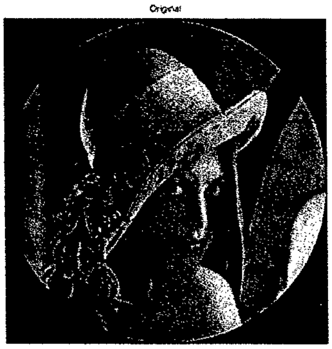


Figure 5: Reverse LPM

3. Chirp Watermark with FMT

In order to make the block-based chirp watermark rotationally invariant but compression resilient, a different implementation of the FMT is necessary. Therefore, the embedding process needs to remain unchanged. Using this knowledge about the chirp watermark's embedding process, some variations were applied to a previous flowchart to implement the FMT [9]. The outcome of these variations produced the flow chart in the figure below.

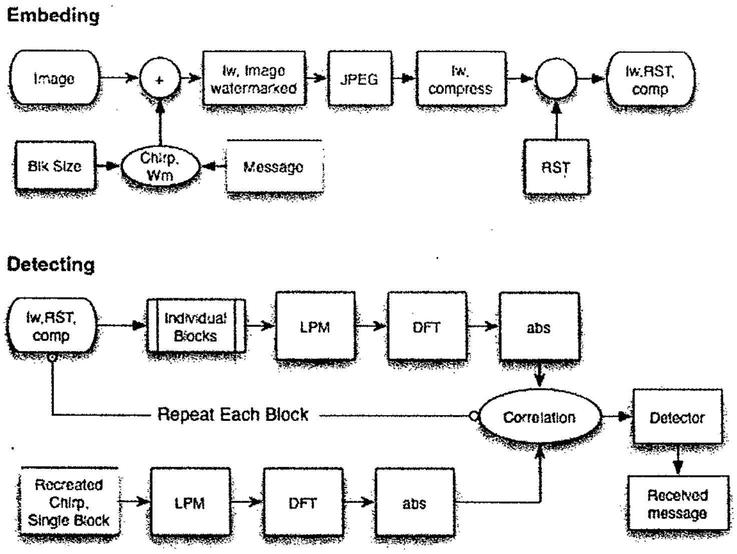


Figure 6: Embedding/Detection Flowchart

Following the chart above, the message bits are now ones and zeros instead of positive and negative ones. This is because the absolute value used in the FMT would remove polarity from the original message bits.

The detection process unlike its counterpart is extremely different from the original procedure [9]. Each image block at the beginning of the detection process is individually sent through the FMT and correlated with the FMT output of a recreated chirp block. This knowledge of the embedding process is also known as communication with side information [3]. While embedding occurs in the spatial domain, the detector operates in the spectral domain because of the use of the DFT. In light of this, a spectral correlation from Cox et al [2] is applied. After all of the correlations are performed, the total correlation's mean is subtracted form the correlation data to set the detectors threshold to zero. The message bits are then recreated using a threshold decision. All values greater than or equal to zero are ones, and all other values are zeros.

3.1. Results

The resulting output of this system with a Peak-Signal-to-Noise Ratio (PSNR) at 40dB and rotations 0-15° was not promising. The best results were from the whole image watermark that yielded perfect results. The trends for the other blocks continually worsen as the block size decreased. The pattern is generally consistent except for certain block sizes. The BER vs. Rotation plots are displayed in the figure below.

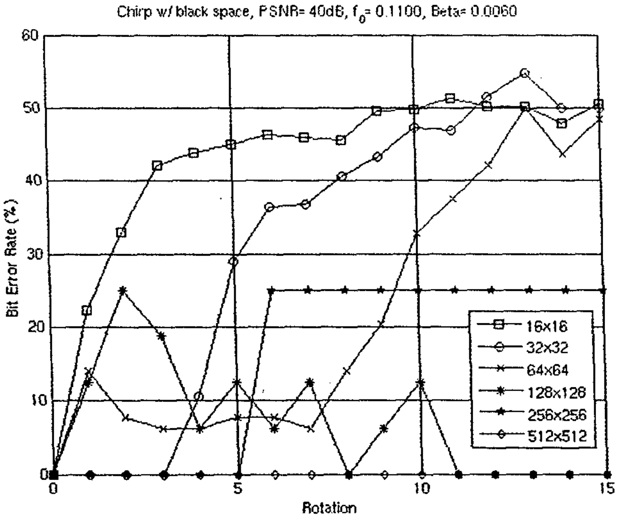


Figure 7: BER Plot of the Chirp with FMT

The lack of performance involving the chirp's block-based watermarks can be attributed to rotational displacement. Rotational displacement involves two points set at different distances rotating around the origin by the same angle, the one further from the center will move a greater distance. This can be seen in the figure below where point number 2 moves further than point number 1.

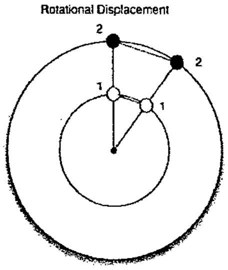


Figure 8: Rotational Displacement Simulation

Blocks farther from the image's origin will move a greater distance and remove or loss its hidden data after rotation. The detector is unaware of this and will still extract erroneous information from that original position. Depending on both the image size and block size, certain blocks can have all of there data removed or lost with only a slight amount of rotation. This is presented in the figure below as the highlighted block loses all of its original data after being rotated 16° (64x64 block size and 512x512 image).

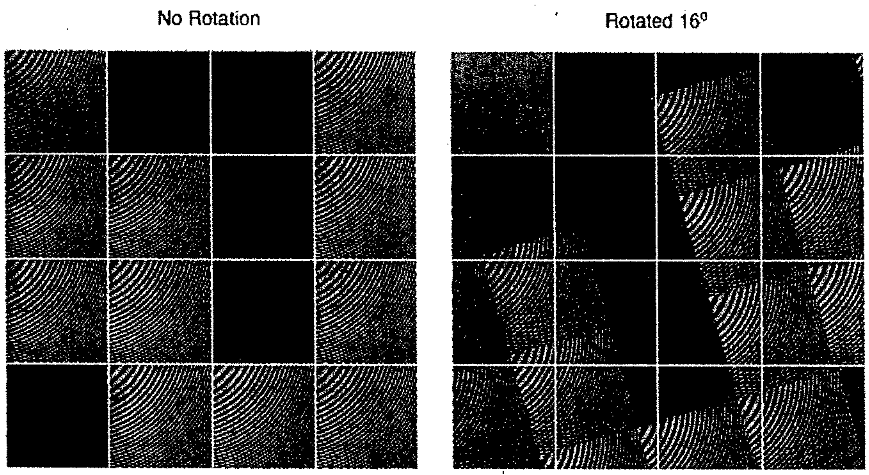


Figure 9: Block Rotational Displacement

The corner blocks are the first to lose their original information from rotation. The table below displays block sizes and the amount of rotation it takes to remove at least one entire blocks data from its original position for a 512x512 image.

Table 1: Rotation Induced Unrecoverable Data

Block Size	Rotation Angle
16x16	4
32x32	8
64x64	16
128x128	37
256x256	90
512x512	Never

After these values have been exceeded in most cases, excluding certain uses of symmetry, repetitive information, and full rotation, the detector will never recover all of the hidden information. Therefore to achieve complete message bit recovery for all block sizes in this scheme, the image cannot be rotated more than 4 degrees.

3.2. Proposed Solutions

There are several possible solutions to this problem, and the first involves creating a whole image watermark and zeroing out only the watermark coefficients specific to blocks representing zero bits. Another possible solution would suggest manipulating the chirp's tunable variables to better align adjacent watermarked blocks and removing any seams that might cause inter-symbol interference. If the problems of the FMT algorithm are further examined, then it is clear that this rotationally invariant domain algorithm is flawed. The FMT algorithm attempted to implement the invariant domain algorithm at the detector. Even though the possible solutions focus on rotational induced chirp interference, the core of the problem involves rotational displacement.

The effects of rotational displacement not only manipulates each individual blocks data, it also removes it. This combination of effects is not counteracted by the FMT algorithm at the receiver for block sizes smaller than the cover image. The algorithm's dependence on the origin of rotation is the problem. It is necessary that the image's center of rotation and the center of a rotational invariant chirp block are the same. These are the key guidelines for the circular chirp algorithm.

4. Circular Chirp

Using the previous guidelines for compression and in particular rotation, the block-based chirp is recreated in a circular form. A circular watermark would correct the previous algorithm's errors, since its geometric shape is conducive to rotation. The circular watermark's center of rotation is also equivalent to the image. Unlike the block-based watermark that loses its information after rotation, a circular watermark's data never leaves the image after rotation. This property alone gives a circular watermark a sizeable advantage over a block-based watermark. In order to maintain the circular watermark's ability to withstand compression, the chirp equations with its tunable values are converted to a circular form. This Cartesian to polar coordinate conversion can be seen in the equation below.

$$W(x,y) = W_{mag} e^{j\pi\beta\left(x^2+y^2\right)+j2\pi f(x+y)}$$

$$x = \rho\cos(\theta), \quad y = \rho\sin(\theta)$$

$$W(\rho,\theta) = W_{mag} e^{j\pi\beta\left(2\rho^2\right)+j2\pi f\rho\left(\cos(\theta)+\sin(\theta)\right)}$$

Equation 4: Circular Chirp Conversion

Theoretically the output of this conversion meets the guidelines of the previous algorithms. Using the two tunable values, the circular chirp will avoid compression, while geometric properties of the circular design will avoid the effects of rotation. This is showen in the figure below.

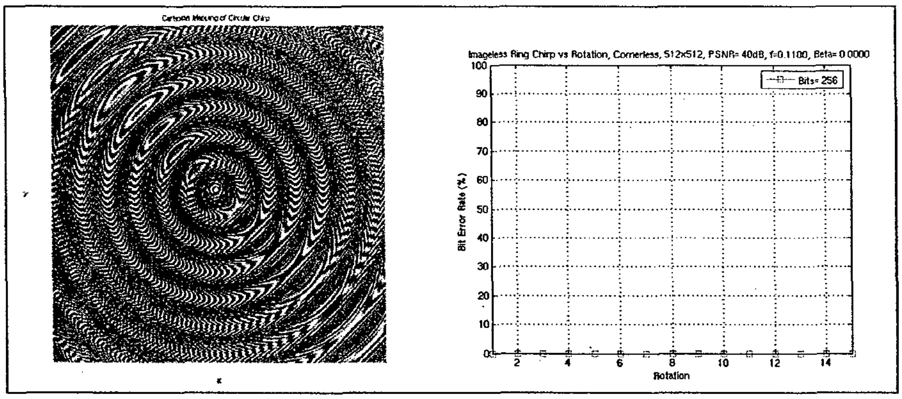


Figure 10: Circular Chirp and Ideal Performance

The BER graph on the right is generated using an imageless-circular chirp similar to the one on the left, and placing it through a series of rotations. This graph supports the previous claim that the circular chirp is rotationally resilient. The circular chirp is also compression resilient and is proven in the results section where a compressed image is added.

4.1. Variables

Like the block-based chirp before, the circular chirp has many variables. These variables can be broken up into two categories: watermark and image variables. The watermark variables consist of the PSNR, ring width, message, and the tunable-value range. The image variables consist of the image's pixel values and quantization variable.

These variables all serve particular purposes that ultimately yield a robust and secure watermark for any gray-scale image. The PSNR controls the watermark strength and is determined by the equation below.

$$W_{mag} = 255 \times \sqrt{2} \times 10^{(-PSNR/20)}$$

Equation 5: Watermark Strength

The circular chirp's ring width dictates the individual message bit's pixel size, which is similar to the chirp's block-size. The combined rings can be seen in the previous figure. Also, the message's binary scheme is a series of positive and negative ones or polar coding. Following the methods in Mobasseri et al [1], the two-tunable values are used to avoid JPEG compression, but instead they both use only 26 values. The variable f is sampled at 100 Hz and ranging from 0 to 0.25, while β is sampled at 1 KHz over a range from 0 to 0.025. Using all of these variables the circular chirp generates the BER for every tunable variable that can be represented as a contour plot similar to the figure below.

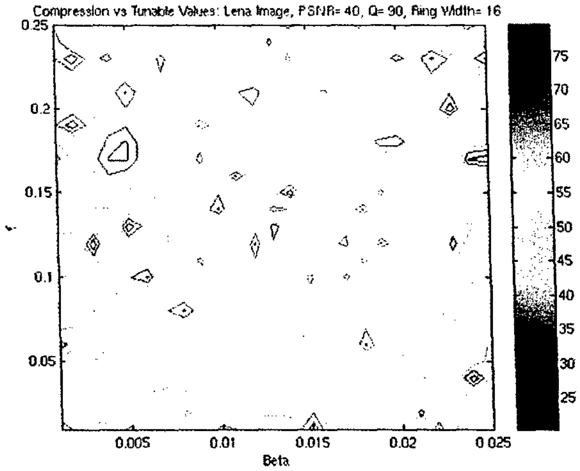


Figure 11: Contour Plot for Q= 90

This plot like the block-based chirp's contours determines the optimal tunable values for specific JPEG quantization values of a cover image.

4.2. Security

While encryption algorithms are arguably the most popular form of security for electronic transmissions, they are removed at a point and leaves its data vulnerable. Watermarks on the other hand, are apart of the image for its existence and can have properties similar to encryption algorithms [10]. The variables mentioned previously have a significant role in the circular chirp's security.

The least secure of the variables is the quantization value of the watermarked image. This can be determined using specialized programs with the quantization table used during compression. Otherwise there are 100 different quantization values that need to be checked. The ring width is more secure than the quantization value and has a total of 240 possibilities. The actual values of the two tunable values and there discrete range are more secure then the previous two variables. This is because of the volatility of the discrete range. This can be seen in the figure below where the sampling rate of the range is pivotal in the creation of contour plots.

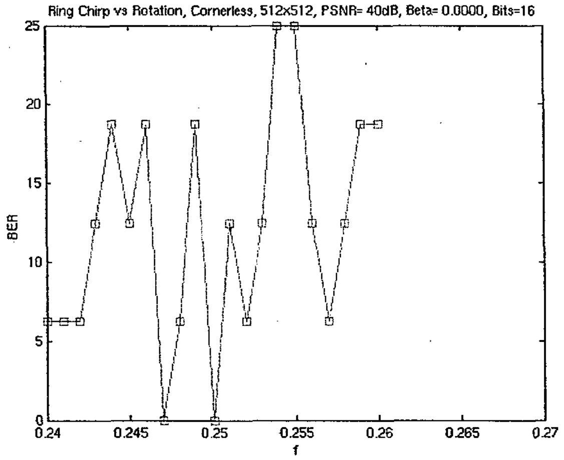


Figure 12: Volatile Tunable Values

If a hacker would sample at a lower rate to decrease the search time, then it is very likely that they would overlook possible key tunable values. If they did determine the discrete range, then they would still have to test 676 combinations (26x26) for the current range with an undetermined amount of optimal values. Finally, the most secure variable is the message itself, which is also necessary to determine the contour plots and has 2^{240} combinations for a 512x512 image (dependent on image size). The total combinations can be determined from the equation below.

$$C = TotalRW \times TotalQ \times Total\beta \times Totalf \times 2^{TotalRW}$$

Equation 6: Total Variable Combinations

For example, an attacker with knowledge of all variables but the exact tunable values, and the message would yield 2.87x10^{77} combinations. This situation is a worst-case scenario, since the ring width and the tunable values are exchanged via a secret key exchange.

4.3. Embedding

Under the guidelines of the proposed solutions, the circular chirp must be embedded before it is transmitted to insure its resilience to both compression and rotation. The flow chart for embedding the circular chirp is displayed below.

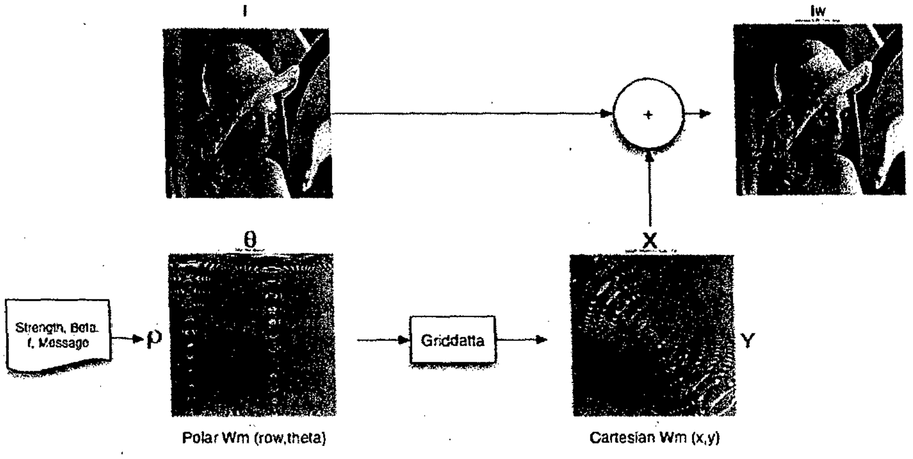


Figure 13: Embedding Flow Chart

The circular chirp is first generated in the polar domain with its watermark variables. The polar mapped circular chirp is then converted to a Cartesian coordinate system. The watermark is then spatial added to the image.

$$W(\rho,\theta)\xrightarrow{Interpolation}W(x,y)+I(x,y)=I_w(x,y)$$

Equation 7: Embedding Process

Interpolation is necessary in this process because of the image's polar coordinates do not sample directly into the Cartesian mapping. This process is presented in the figure below where a particular bit is highlighted.

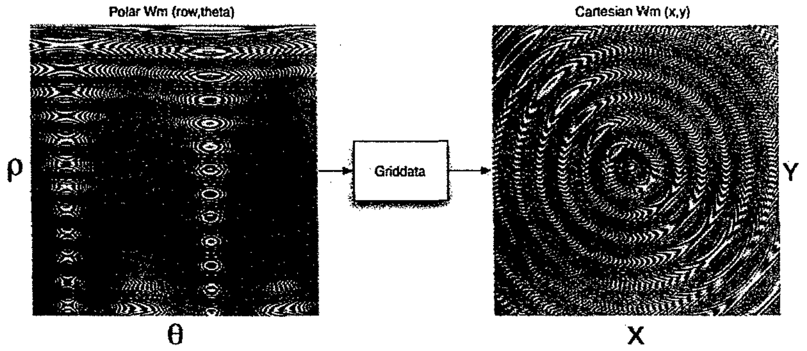


Figure 14: Polar to Cartesian Interpolation

If the polar coordinates were mapped directly the Cartesian representation, then the pixels would become a block like average of the polar values. These block-like patterns would then be exposed to similar results as the block-based chirp. Therefore instead of a general averaging, a more effective non-uniformed cubic interpolation (Griddata) is used to give the circular chirp a smooth representation in the Cartesian domain.

4.4. Detection

During transmission the image is both compressed with JPEG and rotated either accidentally or maliciously by an unknown user. At the receiver, the intended user receives the watermarked image, ring width, and tunable values. The tunable values (β, *f*) and ring width are obtained via a secret key exchange. This detection process is then broken up into two sub-processes that output both the recreated and received rings in the polar domain. The two corresponding rings are then correlated and a threshold is applied to determine each bit of the hidden message. The threshold used is zero, since the binary data implements polar coding. The detection process is illustrated in the figure below.

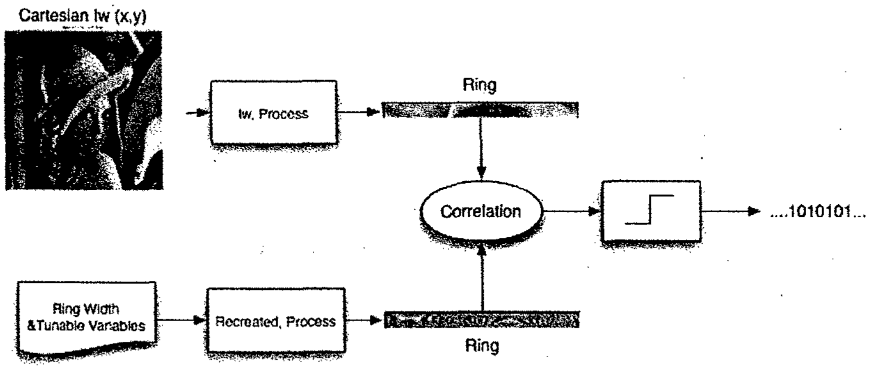


Figure 15: Detection Process

The initial sub-process at the receiver converts the received watermarked image, Iw, into a polar representation. This can be seen in the following figure.

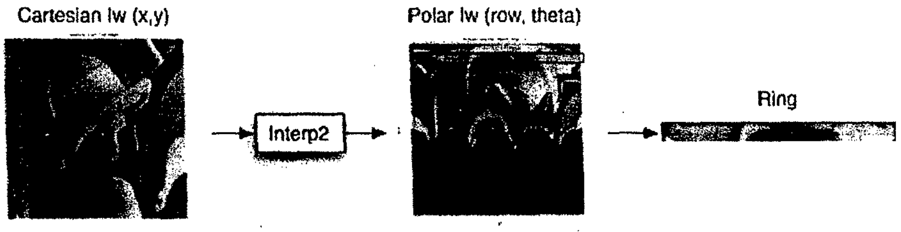


Figure 16: Watermarked Image Sub-Process

Converting to polar also allows the detector to organize the image's ring data into a linear fashion so it is correlated with its equivalent recreated ring. Otherwise the image's watermarked values would not correspond with the recreated chip's values properly.

The other sub-process at the receiver is to recreate the circular chirp using the undisclosed ring width and tunable values. This sub-process can be seen in the figure below.

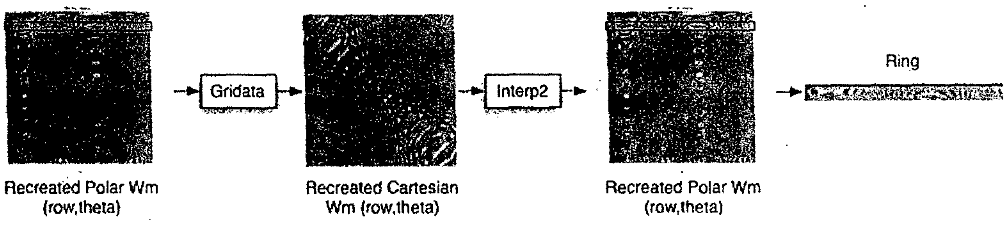
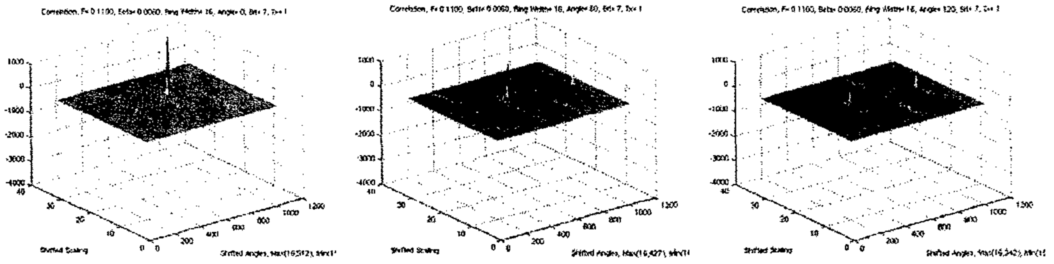

Recreated Polar Wm (row,theta)
Gridata
Recreated Cartesian Wm (row,theta)
Interp2
Recreated Polar Wm (row,theta)
Ring

Figure 17: Recreated Chirp Sub-Process

Following the watermarked image's development to the receiver, the recreated circular chirp begins in the polar domain, is then converted into Cartesian, and finally remapped in the polar domain. Even though the recreated watermark begins in the polar domain, it is still placed through the same amount of interpolation as the watermarked image to alleviate interpolation's effects.

4.4.1. Correlation Optimization

Working with in the polar domain also has benefits outside of organizing correlation data. The polar domain also converts Cartesian rotation into polar translation. The figures below illustrate the ideal results of cross-correlation with rotation.



0° rotation (left), 60° rotation (center), and 120° rotation (right)

Figure 18: Cross-Correlation

It is noticeable from the figures above that the correlation's peaks are shifting with respect to the image's rotation angle. Using its own characteristics, cross-correlation can take advantage of polar translation. Some of cross-correlation's characteristics with respect to this algorithm are operating in the spatial domain and possible rotation estimation. Operating in the spatial domain provides better correlation data that would otherwise be manipulated into a convoluted form. This corruption of data is common in pursuit of a RST resilient algorithm, and is discussed in papers regarding the FMT [7][8][9].

4.4.2. Corner-less Optimization

The detectors performance can also be affected by the location in which you detect the watermark. Even in the previous work with rotationally invariant domains it became apparent that the corners of the image are very flawed. The rotation of the image causes its corners to become cropped and consequently lost data. Avoiding this area would therefore increase the overall performance of the watermark. The BER plot below describes the effects of an imageless circular chirp's performance after rotation with and without corners.

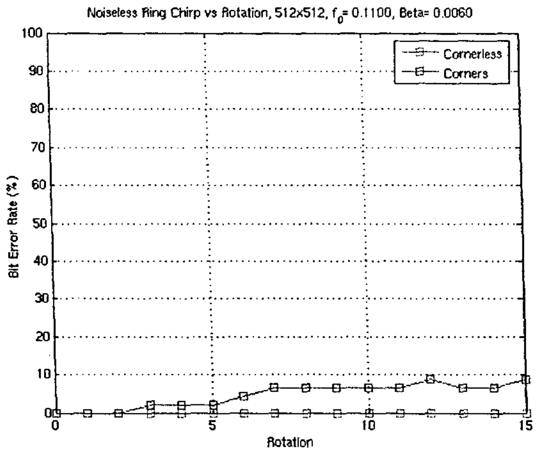


Figure 19: Corner and Corner-less BER Plots

Even in the ideal case, the performance of the circular chirp that attempts to detect the corner rings steadily decreases. On the other hand, the corner-less detector's performance is perfect.

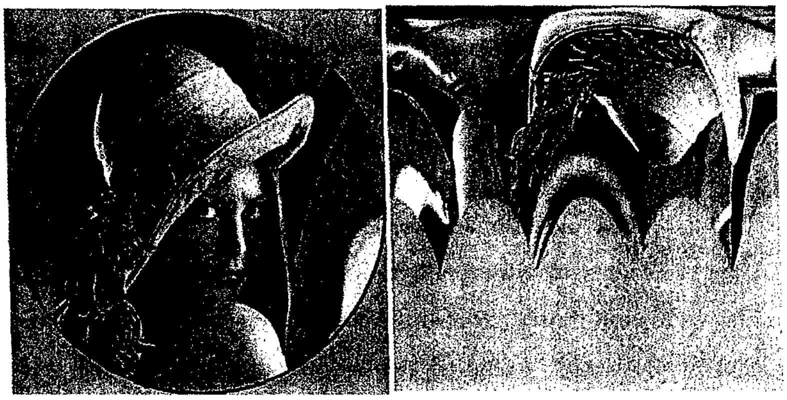


Figure 20: Corner Detection Avoidance

This lack of performance at the corners attributes to avoiding the highlighted regions in the figure above. In order to avoid the corners, the largest detectable ring width cannot exceed half the distance of the image's smallest side.

4.5. Results

In order to obtain superlative results, the previous procedures are implemented over every tunable value in the discrete range for a specific quantization value. As described earlier, the algorithm will output a BER for every tunable variable combination. An optimal set of tunable values is then determined based on the user's application. The image is then watermarked with its optimal variables and the watermarked image is transmitted wherever necessary.

The circular chirp's performance is best determined by comparing it to the previous algorithms' results. The plots in the figure below support these findings. Since the original chirp's rotational BER plot is the worst-case scenario, around 50 percent, then the circular chirp and the block-based chirp using the FMT both outperform the original chirp. Also, there is a distinct difference between the circular chirp's rotational results and the block-based chirp using the FMT. In the figures below, the circular chirp has no error for ring widths 32 to 256 pixels and a average BER of 7 percent for a ring width of 16 pixels, while the only perfect results from the previous algorithm occurs when the block size is equivalent to the entire image (512x512).

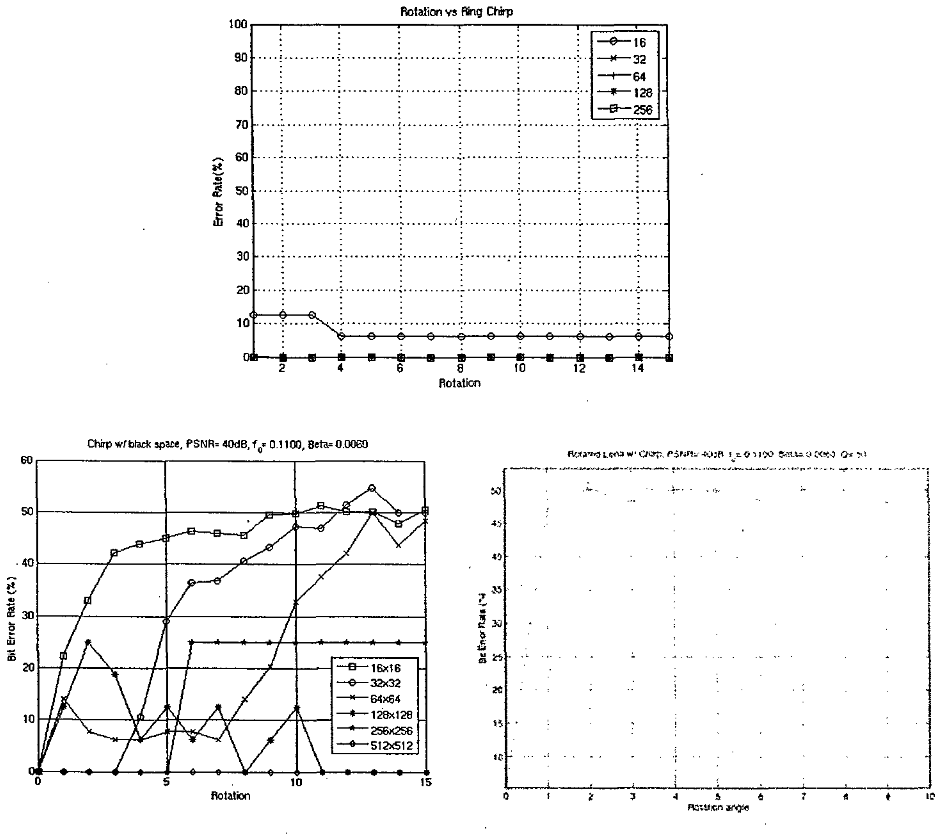


Figure 21: Various Chirp Rotational BER Plots

The results after rotation are therefore certain; the circular chirp's performance is far superior to both the original chirp and the chirp using the FMT.

The goal of the circular chirp though was to be robust to both rotation and compression. Compression's effects were tested on the circular chirp using JPEG quantization values 10 to 100. The results are displayed in the figure below.

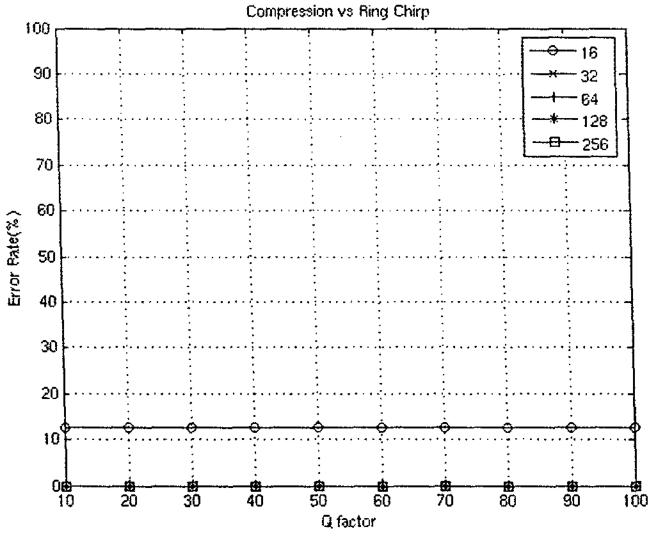


Figure 22: BER Plot Circular Chirp vs. Compression

The compression results for 32 to 256 pixel ring widths are once again without errors and a ring width of 16 pixels has a consistent BER of 12 percent. These therefore conclude that the circular chirp is robust to both compression and rotation. The optimal tunable values for rotation, compression, and several ring widths are in Table 3 and 4 of the appendix.

Interesting enough, it turns out that the tunable values' effects are almost independent of rotation. Therefore, the circular chirp's rotational resilience is exclusively dependent on its circular design. The tables of optimal tunable values in the appendix and the previous BER plots support this outcome. All of the tunable values in the compression table are also in the table for rotation. The compression BER plot's performance is equal to the worse BER of the rotation plot. The only noticeable effects of rotation are within the first 3 degrees of rotation, but these are currently negligible with respect to compression's BER. As a result, the amount of image compression is the main factor in what limits the optimal tunable values' performance. These findings have also increased the total efficiency of this algorithm by decreasing the total amount of necessary computations since each degree of rotation does not need to be tested.

4.6. Future Work

Even though the circular chirp is completely resilient to both compression and rotation for a range of ring widths, it still has a consistent amount error for a 16-pixel ring. Further examination of this and other problems have led to a series of future works.

The consistent BER with a 16-pixel ring width was determined as a smearing of data towards the center of the image. This is due to the large amount of interpolated data towards the center of the image. An example of this convoluted data is in the figure below.

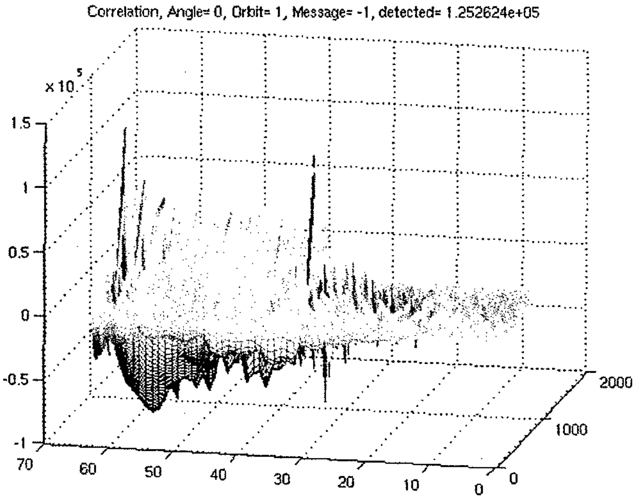


Figure 23: Center Point Correlation Error

In order to fix this problem the center point of the watermarked image must be avoided. In future research, several trials will be run in order to determine the shortest distant from the center point were the first ring can be consistently detected.

Examining a larger range of the tunable values is also being worked on. This should increase the performance of the circular chirp since extra optimal values should be outside of the current range. An increased range will also strengthen the algorithm's security in regard to Equation 10.

The over all capacity in comparison to the block-based chirp is low. This can be increased using rotation estimation and sectoring. As described earlier, cross-correlation is used for detect and its correlation peak shifts with rotation. The relative angle of rotation can be determined with this correlation characteristic. If the angle of rotation is reasonably accurate, then increasing the algorithm's capacity is feasible with sectoring. Sectoring is an idea that involves cutting up each ring into sections or slices, and therefore increasing the watermarks capacity dramatically. This can be seen in the following figure.

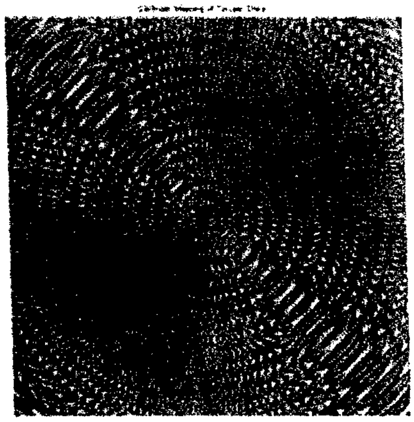


Figure 24: Twelve Sector Circular Chirp

A 16-ring watermark with 10 slices easily becomes 160 bits instead of 16. The only way this will happen though is if there is reasonable accurate rotation estimation. Without this the neighboring sectors would interfere with each other causing inter-symbol interference. Ultimately this future idea is dependent on the accuracy of the rotation estimation and the sector size.

5. Conclusion

In this paper, the problem of rotation and compression resistant watermarking algorithms was explained. While block-based watermarking has many beneficial traits when combating the effects of JPEG compression, these same traits also hinder the watermark's ability to avoid the effects of rotation. Research has explained that the geometric design of the block-based watermark is naturally flawed with respect to rotation. Even rotational invariant algorithms like the FMT cannot completely combat the effects of rotation without severely corrupting the image's data. In light of this evidence, it is apparent that a rotational design is necessary.

The circular chirp watermark combines the chirp's ability to avoid JPEG compression and a circular design to combat rotation. The noiseless results of the circular chirp alone show that it out performs the block-based chirp. In addition when JPEG compression and rotation were applied, it not only outperformed the previous algorithm its results were perfect for the majority of its ring widths. Even though the chirp gives up some capacity to the block-based design, the circular chirp watermark is a solution to the block-based watermark's inability to avoid both JPEG compression and rotation.

6. References

[1] Bijan G. Mobasseri, Yimin Zhang, Moeness G. Amin, Behzad M. Dogahe, "Designing Robust Watermarks Using Polynomial Phase Exponentials", *ICASSP*, vol. II, Philadelphia, PA, March 18-23, 2005, pp. 833-836.

[2] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking of Multimedia," *IEEE Transactions on Image Processing*, 6, 12, 1673-1687, 1997.

[3] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," Proc. IEEE, vol. 87, no. 7, pp. 1127-1141, July 1999.

[6] J. J. K. O'Ruanaidh, and T. Pun, "Rotation, Translation and Scale Invariant Spread Spectrum Digital Image Watermarking", Signal Processing, 66, pp.303-318, 1998.

[7] Woo, Cheng-Seng, Jiang Du, Binh Phan, "Geometrically Rubost Digital Image Watermarking using Scale Normalization and Flowline Curvature," *IS&T/SPIE Conference*, Jan. 2005.

[8] Kelly, Roberto M., H. S. Salguero, and E. S. Salguero, "Image Recognition Using the Fourier-Mellin Transform," *Instituto Politeco Naciopnal Mexico,* Mexico.

[9] Zheng, Dong, Jiying Zhao, and Abdulmotaleb El Saddik, "RST Invariant Digital Image Watermarking Based Log-Polar Mapping and Phase Correlation," IEE Trans. Circuits and Sys., vol. 20, no. 9, September 2003

[10] Jian Ren, Tongtong Li, "A Cryptographic Secure Image Watermarking Scheme", MILCOM, Atlantic City, NJ, October 17-20, 2005.

7. Appendix

Table 2: Rotation Optimal Tunable Values

Rotation vs. Tunable Values			
LENA.tiff	All Rotation: 0-1-15		
Ring Width	**Natural Frequency(*f*)**	**Frequency Rate (β)**	**AVG BER(%)**
16	0.15	0.014	7.421875
32	0.1	0.004	0
	0.25	0.006	0
	0.15	0.008	0
	0.21	0.014	0
64	0.04	0	0
	0.05	0	0
	0.06	0	0
	0.07	0	0
	0.09	0	0
	0.19	0	0
	0.21	0	0
	0.22	0	0
	0.23	0	0
	0.05	0.001	0
	0.12	0.001	0
	0.16	0.001	0
	0.07	0.002	0
	0.15	0.002	0
	0.25	0.002	0
	0.09	0.003	0
	0.15	0.003	0
	0.16	0.004	0
	0.15	0.005	0
	0.18	0.005	0
	0.23	0.005	0
	0.14	0.007	0
	0.13	0.008	0
	0.22	0.008	0
	0.25	0.009	0
	0.16	0.011	0
	0.19	0.011	0
	0.2	0.011	0
	0.25	0.011	0
	0.03	0.012	0
	0.2	0.013	0
	0.21	0.013	0

	0.23	0.015	0
	0.22	0.016	0
	0.23	0.017	0
	0.2	0.019	0
	0.25	0.019	0
	0.05	0.02	0
	0.25	0.02	0
	0.06	0.022	0
	0.07	0.022	0
	0.03	0.023	0
	0.22	0.023	0
	0.03	0.024	0
	0.22	0.024	0
	0.23	0.024	0
	0.05	0.025	0
	0.11	0.025	0
	0.19	0.025	0
	0.24	0.025	0

Table 3: Compression Optimal Tunable Values

Compression vs. Tunable Values			
LENA.tiff	All Q: 10-10-100		
Ring Width	Natural Frequency(ƒ)	Frequency Rate(β)	AVG BER(%)
16	0.15	0.014	12.5
	0.04	0.024	12.5
32	0.1	0.004	0
	0.25	0.006	0
	0.15	0.008	0
64	0.04	0	0
	0.05	0	0
	0.07	0	0
	0.19	0	0
	0.22	0	0
	0.05	0.001	0
	0.16	0.001	0
	0.07	0.002	0
	0.15	0.002	0
	0.25	0.002	0
	0.15	0.003	0
	0.15	0.005	0
	0.18	0.005	0
	0.14	0.007	0
	0.13	0.008	0
	0.22	0.008	0
	0.25	0.009	0
	0.18	0.01	0
	0.16	0.011	0
	0.19	0.011	0
	0.2	0.011	0
	0.03	0.012	0
	0.2	0.013	0
	0.23	0.015	0
	0.22	0.016	0
	0.23	0.017	0
	0.2	0.019	0
	0.25	0.019	0
	0.05	0.02	0
	0.25	0.02	0
	0.06	0.022	0
	0.07	0.022	0
	0.03	0.023	0
	0.22	0.023	0

	0.22	0.024	0
	0.19	0.025	0
	0.22	0.024	0
	0.19	0.025	0

Rotation Invariant Chirp Watermark: An Exploratory Study

An Independent Study
Presented to the Faculty of
The Department of Electrical and Computer Engineering
Villanova University

In Partial Fulfillment of the Requirements for the Degree of
Master Electrical Engineering

By
Christopher E. Fleming
May 13, 2005
Under the Direction of Dr. Bijan G. Mobasseri

Abstract

The following paper is an exploratory study on digital watermarking that focuses on making a compression robust 2-D chirp watermark rotational invariant. Tests conducted in this paper show that this data-hiding algorithm while robust to compression, is fragile to geometric attacks such as translation and particularly rotation.

The necessity for instilling rotational invariance in the chirp watermark is due to its successful performance over a majority of compressed images. While this paper focuses on invariance to geometric attacks, it also overviews basic digital watermarking principles and algorithms that accentuates the chirp watermark's capabilities. In this overview, the chirp is compared with two traditional watermarking algorithms: a spatial and a Spread-Spectrum (SS) watermark that both use a PN-sequence. The SS watermark was a thoroughly researched algorithm that used JPEG quantization principles to optimize its performance through compression. The chirp watermark in comparison surpassed both algorithms over a majority of quantization factors.

The chirp watermark retains its compression resilience at the encoder by implementing a rotation, scaling, and translation (RST) invariant algorithm in the decoder. The Fourier-Mellin Transform (FMT) is this algorithm and is notable for both its RST invariance and the destructive image corruption during the Inverse FMT (IFMT). Having knowledge of the signal shape at the decoder and avoiding the use of the IFMT, makes detection possible with the FMT. This actual detection process is achieved by correlating a recreated individual chirp block after the FMT with the received chirp blocks converted by the FMT.

Since the chirp watermark is a block-based spatial watermark, the watermarked image can only be rotated a small amount before a majority of its data is lost and/or has inter-watermark interference corrupting the block's data. Because of the limitations of block-based spatial watermarking to rotational invariance, the rotation will range from 0-15 degrees, therefore covering rotational unnoticeable regions.

The implementation of this range with the FMT was successful for block sizes equivalent to the entire image with no bit error, but all smaller block sizes results gave worse bit errors and inconsistent data. The study of these inconsistencies lead to the realization that rotation was causing induced chirp interference.

A rotation induced chirp inference occurs when the watermarked image is rotated and watermark blocks rotate into neighboring blocks. This interference only occurs between adjacent blocks hiding the data equivalent to a one, or in other words, a chirp block. The rotational induced chirp interference was determined after all other variables, except for the watermark, were eliminated from the study. The exact cause is still under research, but current results point towards the lack continuity between adjacent blocks.

Acknowledgements

This project would not have possible without the guidance and support of Bijan G. Mobasseri, Bezhad M. Dogahe, Michael Marcinak, and the US Office of Naval Research. This Research was supported in part by the grant #N00014-04-1-0630 awarded by the US Office of Naval Research.

TABLE OF CONTENTS

1. Introduction

The creation of digital media and the rise of the Internet have caused concern over digital rights management. An authentication technique was necessary to determine ownership for digital media. This concern was met with the creation of digital watermarks. A digital watermark is a visually unnoticeable mathematical signature or spreading function within digital media that is implemented in the spatial or spectral domain.

Digital watermarks were initially designed to spread across an entire image for the purpose of digital rights management. Watermarking also has the ability to transmit information as metadata in digital media. When variations of a watermark algorithm are repeated in sequence across an image metadata is created. This covert communications watermarking just as authentication watermarking can be implemented in various domains, but the method investigated in this paper is a spatial watermarking. This watermark uses a tunable 2-D chirp signal for its spreading function in blocks across the image creating metadata.

Dogahe [1] applied this 2-D chirp watermark method and it was designed to be specifically robust against JPEG compression. As mentioned before it is block based, and the detection scheme effectively embeds 1 to 1024 bits of information into digital media. This algorithm while it is robust to compression it is not resilient to geometric attacks such as rotation and translation. The main focus of this paper will be the adaptation of this algorithm for rotational resilience.

2. Watermarking

As described in the introduction, a digital watermark is a visually unnoticeable mathematical signature or spreading function within digital media. To create a simple watermarking algorithm it must take into account its spreading function, the type and quality of cover media it is using, and how and where it will embed and detect the spreading function. The structure of the original technique examined in this paper, the spreading function is a 2-D chirp, the cover media used are digital images, and the embedding and detection occurs in the spatial domain.

2.1. Cover Media

It is important when watermarking to make sure that the quality of your cover media (image) is high so it does not contain any artifacts caused by compression. This will eliminate variables that affect the performance your overall result. For an image-watermarking scheme, an uncompressed standard like TIFF is preferable. Several TIFF images are used for embedding this watermark, so the results are independent from any specific image. The 512x512 images below were implemented in this research.

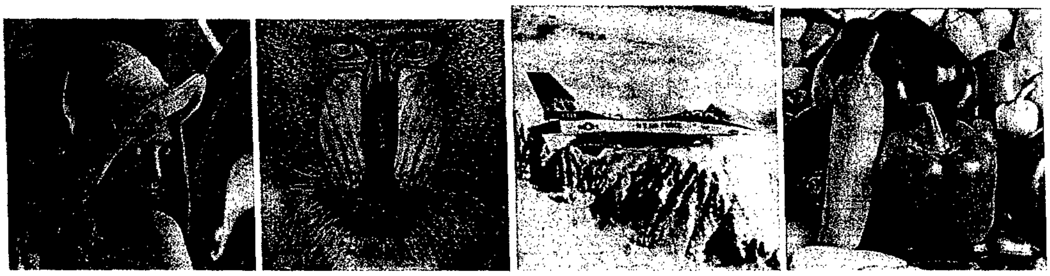


Figure 1: Standard Cover Images

They can be found on the University of Southern California's standard database for watermarking images[1]. Overall using uncompressed images will help create a strong foundation for a watermarking algorithm.

2.2. Embedding/Detecting

Embedding a watermark occurs for most cases in one of two domains: the spatial, or the spectral. Determining which is used is usually dependent on the application. The spatial domain embedding occurs directly in the pixels across the entire image. This is usually done by either adding or subtracting to these pixels, since they only produce small distortions. Spectral implementations use a frequency representation of the image, which are also altered by a spreading function. The ability to watermark in the spectral domain was made famous by Cox [2] in 1997 for his Spread-Spectrum technique. Since the impact of altering the spectral values outside of the DC value are small, mathematical processes outside of adding and subtracting can be implemented easier. The DC value carries the most information in spectral domain with respect to the image and is seldom altered. An example of a spectral embedding scheme is multiplying a Gaussian signal with a weighting factor to an image's spectral coefficients. This procedure was used by Cox [2] and is in the equation below.

$$\begin{aligned}&i = 1,2,....N\\&N = Gaussian\ signal\ length\\&Coefficient_{M_i} = Coefficient_i(1 + a * signal)\end{aligned}$$

Equation 1: Cox's Spread-Spectrum Watermark

An example of a spatial embedding scheme is used in this paper and was used in the work of Dogahe [1]. This can be seen in section 3.

[1] "http://sipi.usc.edu/database/", USC image database.

Detection for watermarking is similar to detection for digital communications, which involves a form of correlation and decision criteria. In Cox et al [3] the authors parallel communications, especially spread-spectrum (SS) with watermarking, and refers to watermarking as, "communications with side information." In most cases in watermarking the signal shape is known and is used to correlate with the received data. This is known as a matched filter in communications. Using the output of the matched filter and knowledge of the signal to noise ratio, the detector can use methods like MAP or maximum likelihood to generate an optimal threshold. The detector then uses this optimal threshold to regenerate the original message. The effectiveness of this process varies with the magnitude of the signal-to-noise ratio.

3. Chirp Overview

The chirp watermark is a 2-D block-based spatial watermark for data hiding. The spreading function design of Dogahe [1] was based off the prior work of Stankovic [4] and his use of a chirp as a digital watermark. This spreading function was designed to be tunable to avoid the effects of JPEG compression. The 2-D chirp spreading function is illustrated in the equation below.

$$W(x,y) = e^{j\pi\beta(x^2+y^2)+j2\pi f(x+y)}$$

Equation 2: Chirp Spreading Function

3.1. Tunable Design

The ability to tune this watermark is a unique and powerful aspect that allows it to work with a variety of compressed images. These tunable values are ß, the chirp rate, and f, the chirp natural frequency. Using these values, the chirp can change the location of its watermark strength and avoid effects of JPEG compression. This can be seen in the graphs below.

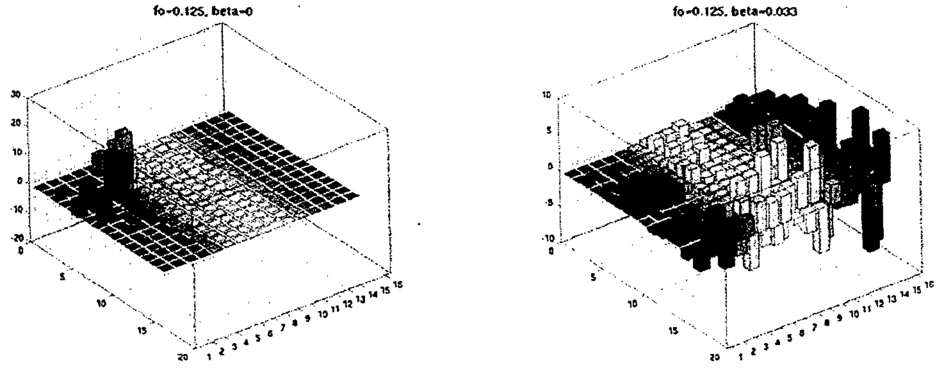


Figure 2: Chirp Energy Adjustment

The optimal tunable values are dependent on both the image and the amount it is compressed, and therefore, each image needs to run tests before it can be properly watermarked. These tests are performed by running a range of both values through the embedding and detection algorithm and determining the probability of error for each combination of the tunable values in the range. The outputs of these tests are contour plots such as the ones in the figure below.

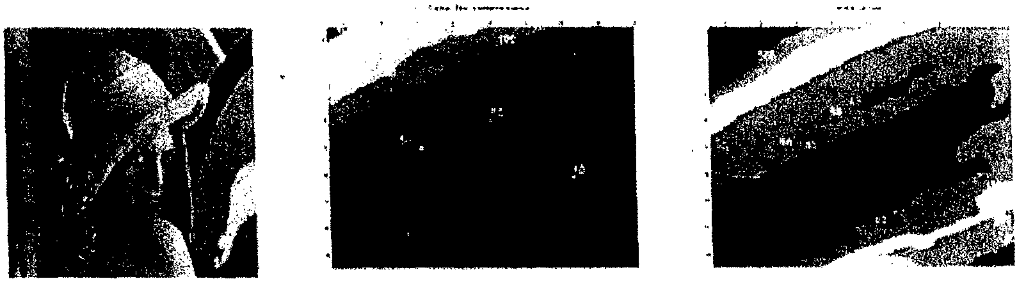


FIGURE 3: LENA CONTOUR PLOTS

The tunable values are then chosen from these graphs and implement with respect to the applications requirements.

3.2. Chirp Watermark Procedure

Adding the spreading function or chirp and its message mask with a weighting factor, directly to the pixel values, is the basic application methodology of the chirp watermark to an image. The message mask is a matrix with a size equivalent to the quantity of blocks used with the cover media.

This matrix is filled with positive and negative ones to represent a binary message (hidden message). The weighting factor or watermark magnitude controls the strength of the watermark. The implementation of these three components to the cover media is shown in the equation below.

$$
\begin{aligned}
&I_w(m,n,x,y) = I(m,n,x,y) + k \cdot \mathrm{Re}[d[m,n] \cdot W(x,y)] \\
&k = \textit{weighting factor} \\
&d[m,n] = \textit{message mask}
\end{aligned}
$$

Equation 3: Chirp Watermark Embedding

The watermark magnitude's strength directly affects the peak-signal-to-noise ratio (PSNR) that is used as a perceptual metric for visual cover media quality after watermarking and signal strength. An expectable industry standard for an effective watermarking scheme is capable of detecting its watermark with a PSNR of 40dB or higher.

The detection process has two separate procedures one for encryption and one with out. To encrypt the chirp all blocks must use a different combinations of the tunable variables, and those variables are sent to the receiver through a secret key exchange. The unencrypted version uses the same tunable variables for all blocks, and determines the tunable variables by summing all of the watermarked blocks together.

After the tunable variables are determined, each block of the watermarked image is correlated with the conjugate of the recreated chirp. The output of this correlation removes the phase and yields the magnitude. This magnitude is then decoded into ones and zeros by applying a threshold at zero, and the resulting sequence is a recreation of the encoder's hidden message. The mathematical equations for this correlation are below.

$$
\begin{aligned}
&C(m,n,\beta,f) = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} I_w(m,n,x,y) W^*(m,n,x,y) \\
&= \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} I(m,n,x,y) W^*(m,n,x,y) + \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} k \,\mathrm{Re}[d(m,n) W(x,y)] W^*(m,n,x,y) \\
&\bar{d}(m,n) = \begin{Bmatrix} 1 & \mathrm{Re}\{C(m,n,\beta,f)\} \geq 0 \\ 0 & \mathrm{Re}\{C(m,n,\beta,f)\} < 0 \end{Bmatrix}
\end{aligned}
$$

Equation 4: Chirp Watermark Detection

4. JPEG Compression Watermark Comparisons

The chirp watermark's purpose was to use its tunable characteristics to avoid the effects of JPEG compressions and transmit a hidden message. To give an idea of the exceptional performance of this scheme, it needed to be compared against block-based watermarking algorithms that are both well studied and effectively survive compression. The two traditional watermarking algorithms compared against the chirp are a simple spatial implementation of a PN sequence and a representation of Hernandez et al [5] spectral implementation with a PN sequence. The spatial PN sequence was implemented in the same manner as the chirp, but the chirp's spreading function was switched with a PN-sequence. The spectral PN-sequence watermark on the other hand, was recreated using optimized SS techniques [2,3].

4.1. SS Watermarking with Discrete-Cosine Transform Coefficients

The Discrete-Cosine Transform (DCT) takes discrete variables and converts them into a real-valued frequency representation of equal size. The equation for this procedure can be seen below.

$$x[n] = \mathrm{Re}\left[\frac{1}{N}\sum_{k=0}^{N-1} X(k)\cdot e^{jk\frac{2\pi n}{N}}\right], \quad DCT$$

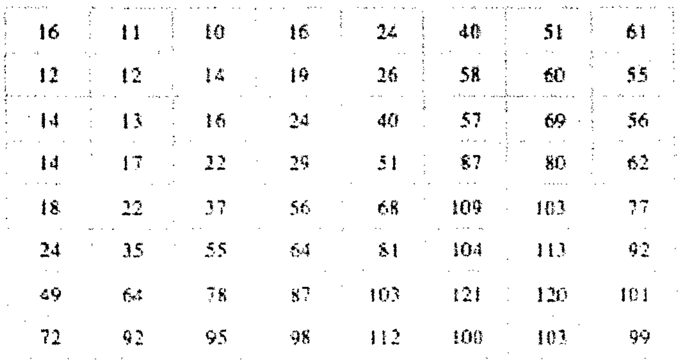
Equation 5: DCT

This same procedure is also used in SS watermarking. Taking the image's pixel values and calculating the real part of the Discrete Fourier Transform executes this process. The resulting output of this equation on a two-dimensional image is called the DCT coefficients. Embedding a SS watermark is achieved by altering a series of DCT coefficients, and using the Inverse DCT (IDCT) on altered coefficients creates a watermarked image.

4.2. DCT Coefficients and JPEG Compression

The DCT coefficients are significant part of JPEG compression; because JPEG uses a luminance quantization table (Grayscale) to reduce the DCT coefficients for each non-overlapping 8x8 DCT block in an image. This reduction of the DCT coefficients also decreases the amount of information stored by the image; therefore compressing the file. These altered DCT coefficients from the standard JPEG quantization table are also the least likely to cause general image distortion. This general image distortion is determined from the image's the most commonly used frequencies. This is one of many ways the JPEG standard manipulates digital images to reduce storage size with the least amount of image distortion.

The JPEG quantization tables also categorize the most commonly used frequency coefficients in an 8x8 DCT block. The DCT coefficients can be sectioned into three categories of frequency: the low, middle, and high frequency coefficients. From the JPEG quantization tables, the low frequency coefficients cause the most visual distortion when altered, because they are changed the least. The lowest frequency coefficient is the most fragile, and it is called the DC coefficient. It contains the most common visual information and is not watermarked because of the amount visual distortion it causes. As the frequency coefficients move further away from the DC coefficient, the amount visual distortion decreases. Therefore, the high frequency coefficients barely distort the image and the middle coefficients slightly affect the image. This is illustrated in the values of the standard quantization table, as the lower quantities represent the visually most fragile coefficients, and the higher quantities represent the more visually insignificant coefficients. An example of a JPEG quantization table can be seen in the figure below.



16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figure 4: Luminance Quantization Table

In the JPEG standard, the 8x8 DCT block coefficients are organized in a zig-zag structure. This zig-zag scan can be seen in the figure below.

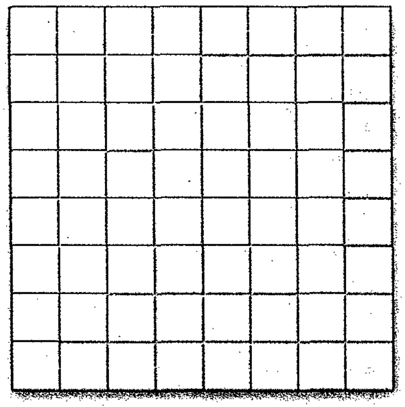


Figure 5: Zig-Zag Scan

The lower frequency coefficients of the DCT block are in the top left corner, the high frequency coefficients are in the bottom right, and the middle coefficients are in the middle diagonals of the block. This zig-zag structure is also useful in locating and watermarking specific frequencies in a DCT block.

4.3. Optimal SS Watermarking with DCT Coefficients

To increase the performance of the SS watermarking algorithm against JPEG compression, the location to embed the watermark in the DCT block coefficients was tested [5]. Utilizing the knowledge about JPEG's quantization tables, an optimal watermarking location can be determined. The high frequency DCT coefficients would barely distort the image but are removed too easily as compression increases. The low frequency DCT coefficients on the other hand would survive compression very well but greatly distort the image. The middle frequency DCT coefficients have the right range of acceptable compression resistance and only a slight image distortion. After a comparative analysis with Hernandez et al [5] and several combinations of middle frequency coefficients, it was apparent that the low-middle frequency DCT coefficients, 7-37, were the best DCT coefficients to watermark for all block sizes.

Another variation to Hernandez et al [5] was the use of a PN-sequence instead of a Gaussian signal. The reason a PN- sequence was used over a Gaussian signal was because it performed better after compression in the detector. The PN-sequence while pseudorandom has better correlation output properties that a Gaussian signal. The combination of both the PN-sequence and watermarking of the low-middle frequency DCT coefficients actual outperformed the compression performance plots by Hernandez et al [5].

4.4. Chirp Comparison to Traditional Watermarks

The overall efficiency of the chirp algorithm is best proven in a comparison to other noteworthy algorithms such as the SS Watermarking discussed previously and a spatial PN-sequence. The results of this comparison can be seen in the figure below[2].

[2] Graph code in section 8.1

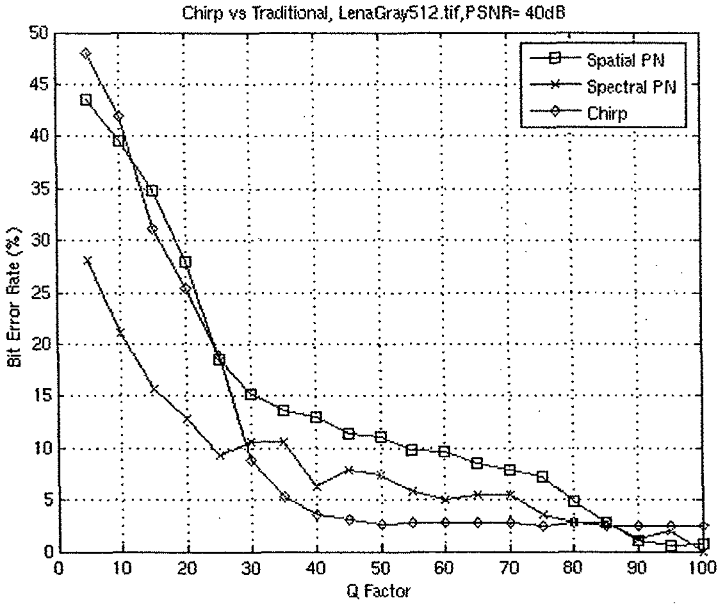


Figure 6: Chirp vs. Traditional Methods

As you can see the chirp noticeably outperforms both of these watermarking algorithms over various compression values of JPEG compression. In JPEG compression the more commonly used quantization factors are between 50 and 100, and in that range the chirp outperforms both watermarking schemes. The results from the figure above and the fact that the chirp can obtain these results consistently with other images, more than enforces the overall superiority of the chirp's performance against compression in the world of watermarking.

5. Geometric Attacks

The image-processing tools rotation, scaling, and translation (RST) are quite common, and are found in most photo or image software packages. While the chirp is resistant to JPEG compression it is vulnerable to these image-processing tools, or in other words "geometric attacks." These three image-processing tools are the subjects of this section because of their negative effects on the chirp's detection. Because the watermark is implemented in the spatial domain, it is directly susceptible to these attacks.

Two of these attacks were implemented to the chirp's detection algorithm with Matlab. The experiment was carried out using a smaller viewing portion of the image. This "Viewing Window" was also synchronized or in multiples of the block size being applied to eliminate extra variants in the tests. An example of the viewing window can be seen in the figure below.

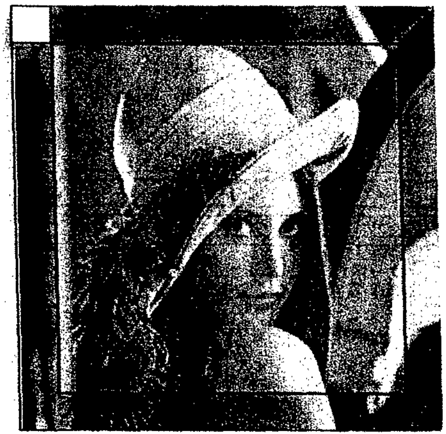


Figure 7: View Window

The results of these test illustrated the chirp's inability to withstand both rotation and translation. The figures below display the BER curves for both rotation and translation. Scaling was not examined because previous research showed the chirp reasonably resilient to scaling [1].[3]

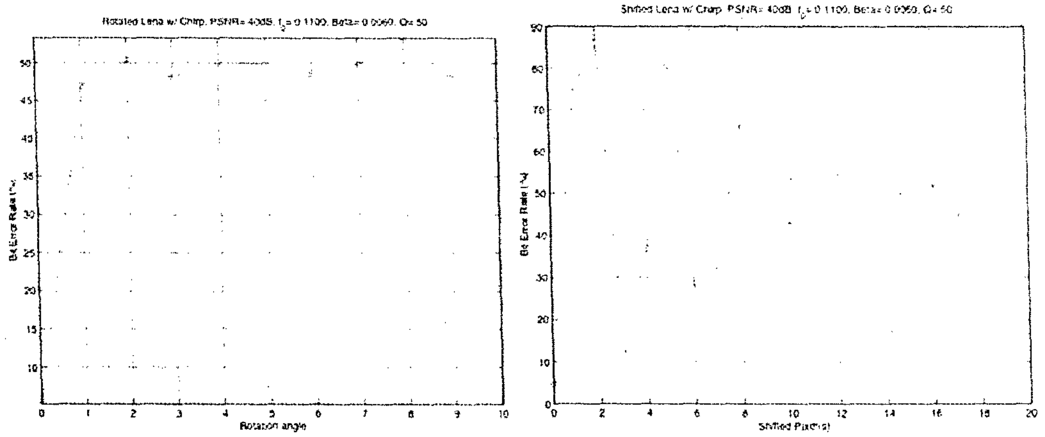


Figure 8: Translation and Rotation BER plots

The effects of rotation are troublesome with a 43% BER for only one degree of rotation. This is especially worrisome since detection is unpredictable when the BER is 50%. Translations effects are not as severe, but they are still poor. After being shifted 8 pixels, the watermark converges towards a BER of 50%.

Statistically the results are poor, and the visual results of the effects are not even noticeable. The effects of both rotation and translation are displayed in the figures below.

[3] Graphs code in sections 8.2 and 8.3

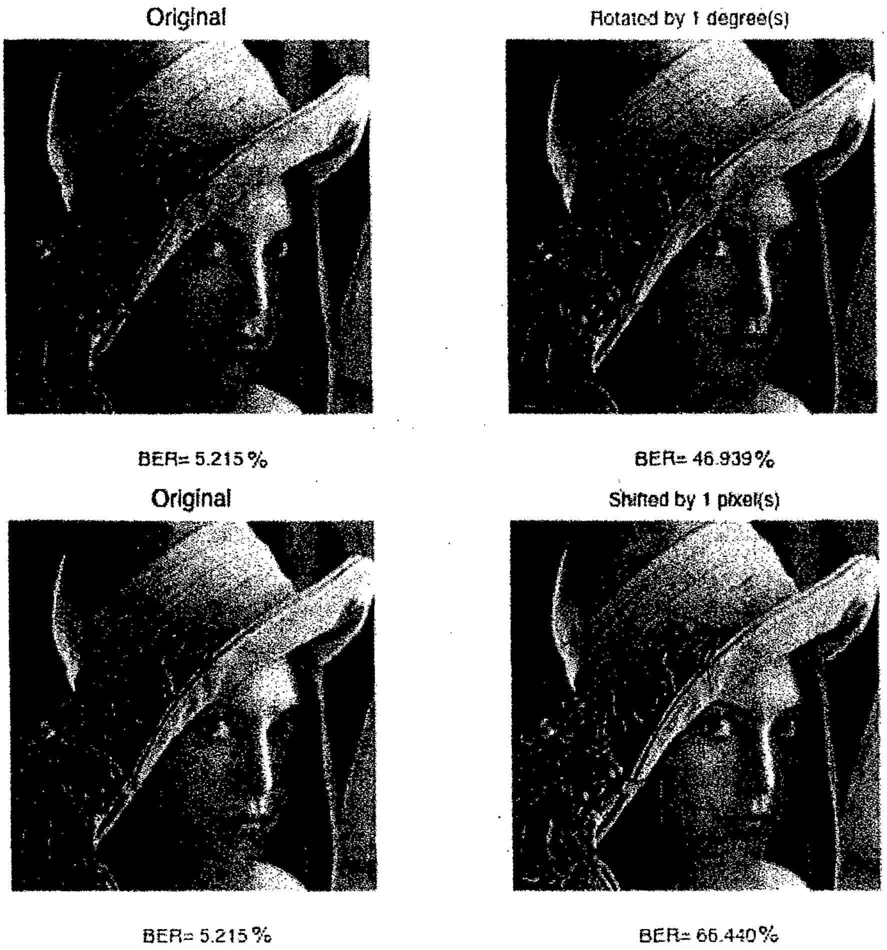


Figure 9: Visually Effects of Rotation and Translation

The figures speak for themselves; the one figure is rotated only one degree and has a BER of 47%, while the figure shifted by one pixel has a BER of 66%. The vulnerability of this watermark to such attacks is quite apparent. The following will discuss the use of an invariant-domain approach to add geometric resilience to this watermarking scheme.

The common geometric attacks of RST were described previously, and their effects were significant to the chirp watermark's detector. Even though the effects of translation were destructive to the detector, in comparison, rotation's effect on the detector was worse. Therefore the remainder of this paper will focus on using an invariant-domain method to eliminate rotation's influence on the chirp. The rotations involved in this project will only range from 0-15 degrees, because large rotations are visually noticeable and can be corrected in most images. This range was also influenced by the optimal recovery criteria describe in section 5.5.1.

6. Rotational Invariance by Way of the FMT

The invariant-domain method used in this paper is the FMT [6]. This transform is known for it's ability to be invariant to RST, but its ability to damage image quality is also known [7]. The FMT can be broken down into two complex problems the Log-Polar Mapping (LPM) and the complex modulus of the Dicrete-Fourier Transform (DFT). Image quality is lost because of interpolation that occurs in the Log-Polar Mapping, since the Fourier Transform is reversible and lossless.

6.1. LPM

Log-Polar Mapping is a process where a linear Cartesian coordinate system is converted into a logarithmic polar coordinate system. The value of variable, 'n', is equivalent to length pixels on the on side of the image. For example, a 512x512 image would have a 'n' value of 512. The value of n then helps determine the center points (X_{center}, Y_{center}), the logarithmic scale (ρ), and angular scale (θ). This process is described in the equations below [6, 8].

$$
\begin{aligned}
&d = (n-1)/2, \quad \textit{even number of pixels} \\
&X_{center} = Y_{center} = (n+1)/2 \\
&\rho = \{0 \ldots \log_{10}(d)\} \\
&\theta = \{0 \ldots 2\pi\} \\
&x_{lp} = e^{\rho} \cdot \cos(\theta) \\
&y_{lp} = e^{\rho} \cdot \sin(\theta)
\end{aligned}
$$

Equation 6: LPM Procedure

The x_{lp} and y_{lp} are the mapping positions that the Cartesian coordinates (x,y) are interpolated too. This occurs because rotation is continuous while the pixel positions are discrete and can be positioned between pixels. The interpolation used for log polar mapping is bicubic. This is displayed in the figure below from [8]:

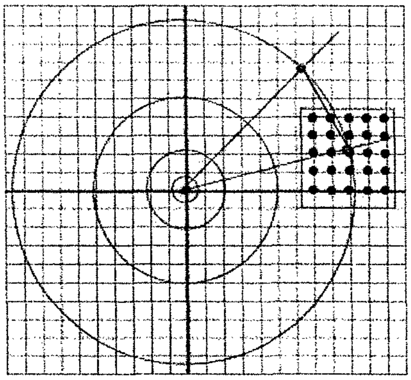
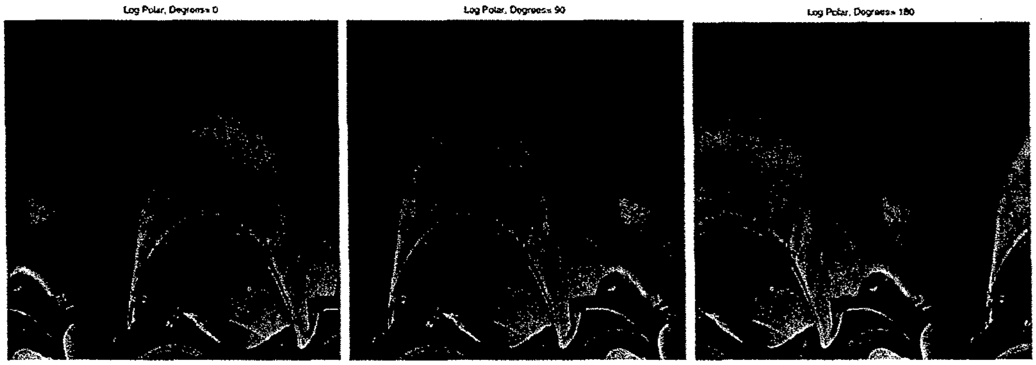


Figure 10: Pixel Interpolation

LPM is used in the FMT because it turns scaling and more importantly rotation into translation. Therefore a rotation clockwise (cw) will be a shift right, and a rotation counter-clockwise (ccw) will be a shift left (scaling causes vertical translation). The shift in the LPM after rotation can be seen in the figures below[4].



Log Polar, Degrees= 0

Log Polar, Degrees= 90

Log Polar, Degrees= 180

Figure 11: LPM Representations of Lena

The figures also show the image distortions mentioned earlier by the nonlinear operations of the LPM. This translation produced by the LPM allows the image and the chirp to become rotationally invariant with the Discrete Fourier Transform.

6.2. Discrete-Fourier Transform

The Discrete Fourier Transform (DFT) and its properties are one of the most widely known mathematical principals in signal processing. One of the properties of the DFT is the delay property. In

[4] Code for these images are in section 8.4

this property a delay in time or a shift in space is converted into a shift in phase. Both the DFT and the delay property can be seen in the equations below.

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\cdot e^{jk\frac{2\pi n}{N}}, \quad DFT$$

$$x[n-k] = X(w)\cdot e^{-jwk}, \quad Delay\ property$$

Equation 7: DFT and Delay Property

Applying this equation with the complex modulus or absolute value removes the phase from the equation; making the image invariant to translation in the LPM. The original image data is therefore invariant to rotation and scaling, because translation in the Log-Polar domain is equivalent to scaling and rotation.

$$x[n-k] = X(w)\cdot e^{-jwk} \Rightarrow \quad \left|X(w)\cdot e^{-jwk}\right| = \left|X(w)\right|$$

Equation 8: Complex Modulus of the DFT

The authors of [6][8][9] used the original unmarked image to successfully detect a signal after RST using the Fourier-Mellin transform. Even though it was effective in there case, it is not possible to maintain this same algorithm for the chirp, because it is a blind watermarking scheme that is embedded spatially and the reverse LPM's distortion are too severe. An example of this is seen in the figure below[3]:

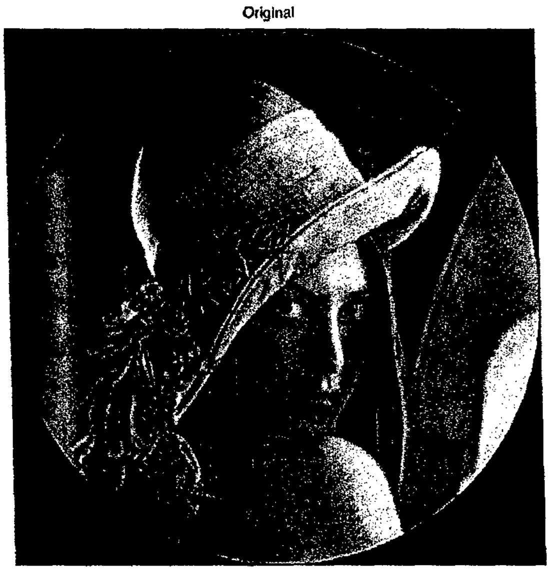


Figure 12: Reverse LPM

7. Rotation Invariant Chirp Watermark by the Way of the FMT

7.1. Rotation Invariant Chirp Watermark Method

The rotational invariant chirp Watermark must implement a different algorithm then previously used with the FMT to maintain the chirp's robustness to JPEG compression. To accomplish this task, the embedding process needs to remain unchanged. Using this knowledge about the chirp watermark's embedding process, some variations were applied to a previous flowchart to implement the FMT [9]. The outcome of these variations produced the flow chart in the figure below.

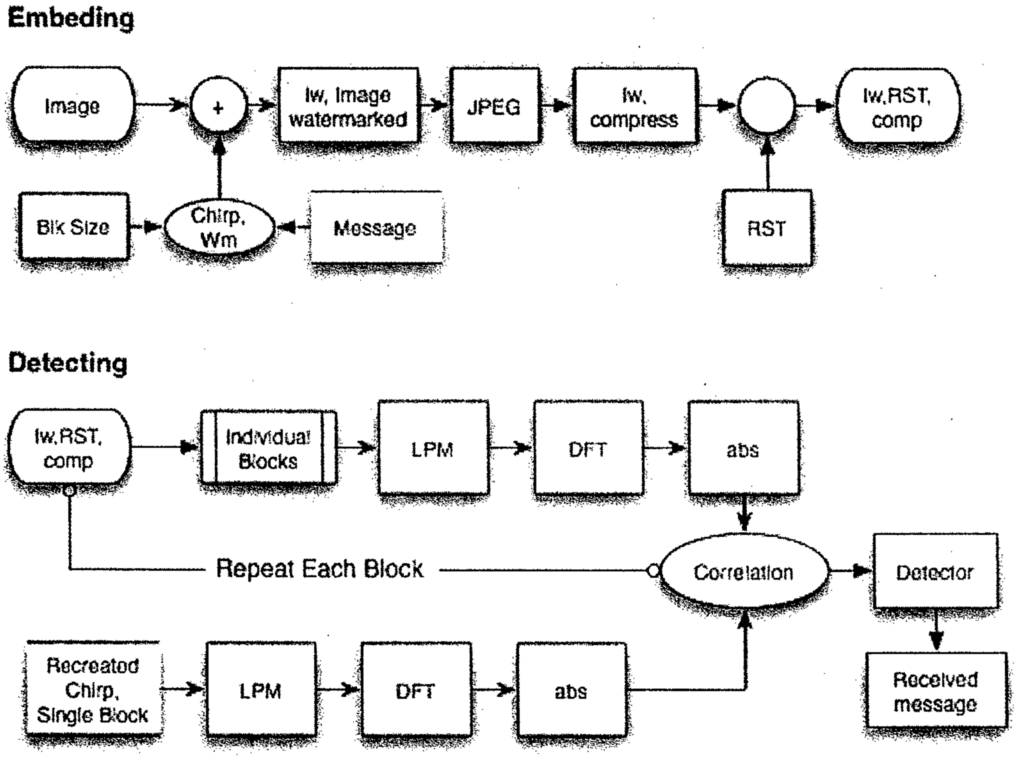


Figure 13: Embedding/Detection Flowchart

Following the chart above, the embedding scheme would remain unchanged with the exception of the message bits. The message bits would now be ones and zeros instead of the positive and negative ones used previously. This is because the absolute value used in the FMT removes polarity from the message bits, making it impossible to determine one signal from another. The implementation of the RST attack is applied during embedding section of the flow chart for coding purposes. In real-world situations, The RST attack occurs during the transmission of the watermarked image.

The detection process unlike the embedding process is extremely different from the original procedure. Each image block at the beginning of the detection process is individually sent through the FMT and correlated with the single-block-size-recreated chirp. This recreated chirp also has to go through the FMT to accurately correlate with the received blocks. While embedding occurs in the spatial domain, the detector is applied in the spectral domain because of the use of the DFT in the FMT algorithm. Since the data at the detector is spectral, the correlation used for the rotation invariant chirp watermark detection is identical to the correlation applied to SS watermarking [2]. To perform correlation with the received data, X^*, and the single-block-size-recreated chirp, X, the dot product is performed on the received data and the single-block-size-recreated chirp, and then divided by the square root of the dot product of the received data with itself. This correlation equation is displayed below.

$$Correlation = (X^* \cdot X')/\sqrt{X^* \cdot (X^*)'}$$

Equation 9: Rotation Invariant Chirp Watermark Correlation

After all of the block's correlation is performed, the mean of the correlation output is determined and subtracted form the correlation data. Doing this allows the threshold for detection to be zero. The message bits are then recreated by implementing a zero threshold on the zero-mean correlation output. The decision criteria is as follows, all values greater than or equal to zero are ones, and all values less than zero are zeros.

7.2. Rotation Invariant Chirp Watermark Results

The resulting output of this system was mixed. The whole image watermark gave perfect results yielding a 0% BER with a PSNR at 40dB for rotations 0-15 degrees. The trend for the other blocks with a PSNR at 40dB would continually worsen as the block size decreased. The BER vs. Rotation plots are displayed in the figure below[5]:

[5] Code for this graph is in section 8.9

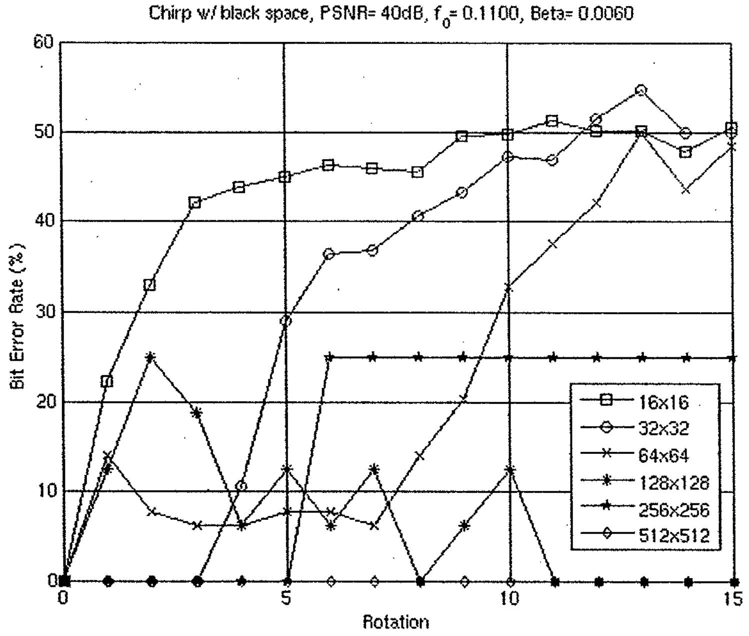


Figure 14: BER Plot of the Rotation Invariant Chirp Watermark Detection

This figure describes the worsening effect of BER to rotation as block size decreases. The pattern is generally consistent except for certain block sizes. This phenomenon will be described later in this section dealing with rotation induced chirp interference. All of the BER graphs generated in this section do not embed the watermark into an image or compress it. This is done to eliminate signal-to-noise ratio as a variable, therefore the image consists of only pure signal.

The lack of performance involving the chirp's block-based watermarks can be attributed to rotational displacement. Rotational displacement in general involves separate points at set distances moving circularly around a center point. When comparing two points set at different distances rotating around the origin by the same angle, the one further from the center will move a greater distance. This can be seen in the figure below where point number 2 moves further than point number 1.

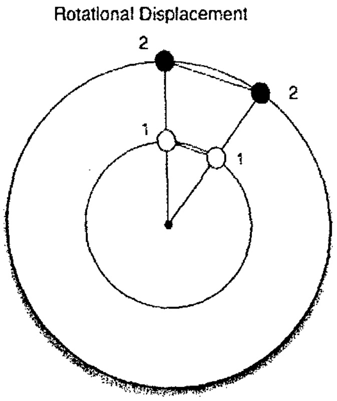


Figure 15: Rotational Displacement Simulation

This also holds true for blocks that are farther from the center of the image. This means blocks farther from the image's origin will move a distance that would remove its hidden data from its original position after rotation. The problem with this is the detector will still attempt to extract information from that original position thinking the correct data is there. Depending on both the image size and block size, certain blocks can have all of there data moved or lost with only a slight amount of rotation. This is presented in the figure below as the highlighted block loses all of its original data after being rotated 16^0 (64x64 block size and 512x512 image).

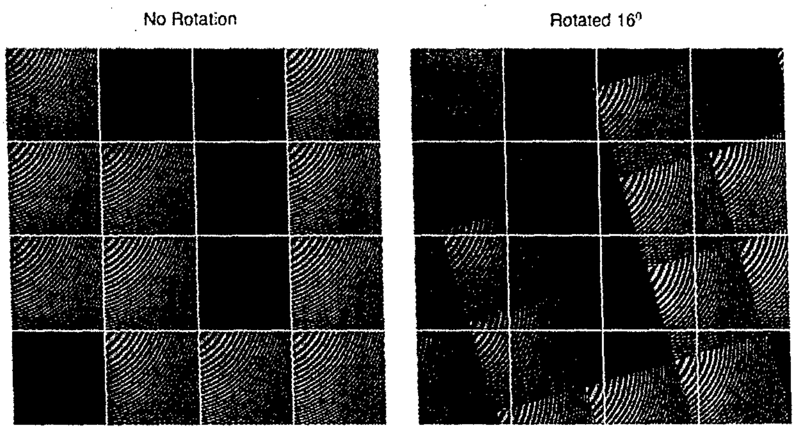


Figure 16: Block Rotational Displacement

The corner blocks are the first to lose their original information from rotation. The table below displays block sizes and the amount of rotation it takes to remove at least one blocks data from its original position for a 512 by 512 image.

Block Size	Rotation Angle
16x16	4
32x32	8
64x64	16
128x128	37
256x256	90
512x512	Never

Table 1: Rotation Induced Unrecoverable Data

After these values have been exceeded in most cases, excluding certain uses of symmetry, repetitive information, and full rotation, the detector will never recover all of the hidden information. Therefore to achieve complete message bit recovery for all block sizes in this scheme, the image cannot be rotated more than 4 degrees.

7.3. Regional Capacity Performance

In an attempt to lessen the effects of rotational displacement, the image's capacity is decreased in order to increase performance came. This would occur by removing the outer layers of blocks or "block region" from the image. Since the outer most block region is subject to the most rotation displacement, removing it would increase performance. This procedure is only possible for block sizes that have more than one region. For example, a 512x512 image cannot use this technique for block sizes 512x512 and 256x256. A description of the outer region removal is shown in the figure below.

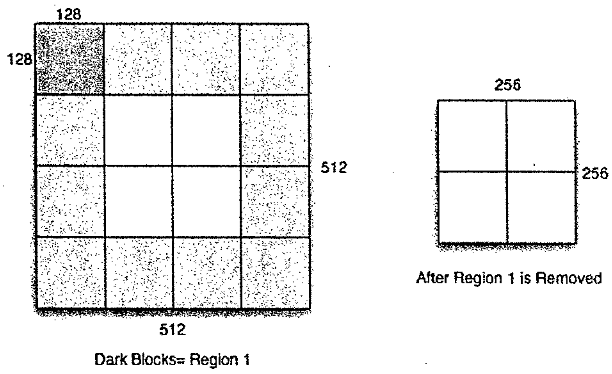


Figure 17: Regional Capacity Example

The block regions quantity value increases for each layer of blocks that approaches the center of the image. The last block region for any block size contains the four blocks surrounding the image's center. For example, the dark outer region in figure 17 is block region 1, and then the inner region is referred to as region 2. Since block region 2 in this example consists of four blocks surrounding the center of the image, it is also the final block region for this block size. The total amount of regions that can be removed per block size and the equation to determine these values are displayed below.

$$
\begin{aligned}
&\mathrm{Re}\,movable\ \mathrm{Re}\,gions = [(n \cdot 2^{-1})/m] - 1 \\
&n = amount\ of\ image\ columns \\
&m = amount\ of\ block\ size\ columns
\end{aligned}
$$

Equation 10: Total Removable Block Regions

Block Size	**Removable Regions**
128x128	1
64x64	3
32x32	7
16x16	15

Table 2: Total Removable Block Regions per Block Size

This operation was performed on block sizes 128x128 to 16x16, and the results of the four block sizes are in the figures below[6].

[6] Code for these graphs are in section 8.6

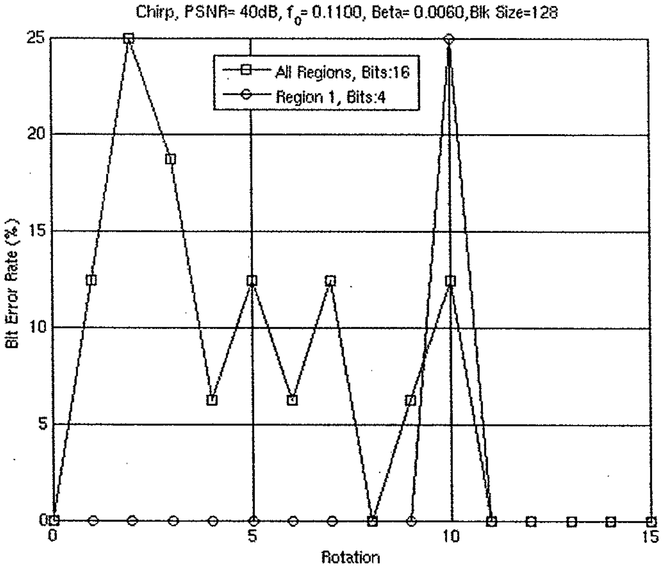


Figure 18: Regional Capacity BER Plots for 128x128 Blocks

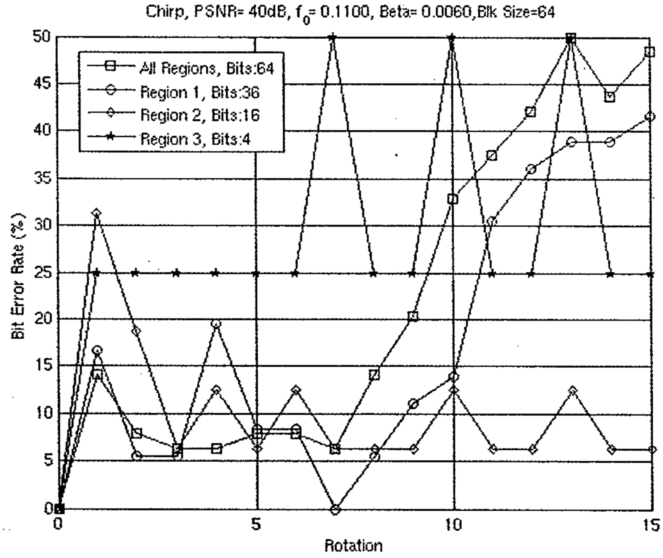


Figure 19: Regional Capacity BER Plots for 64x64 Blocks

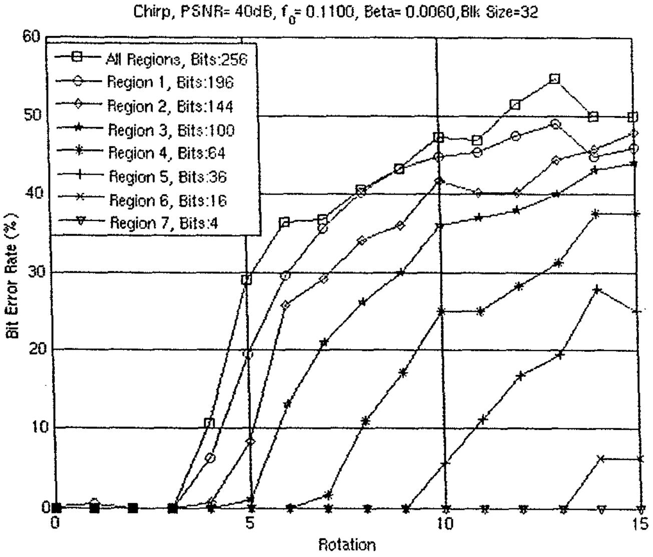


Figure 20: Regional Capacity BER Plots for 32x32 Blocks

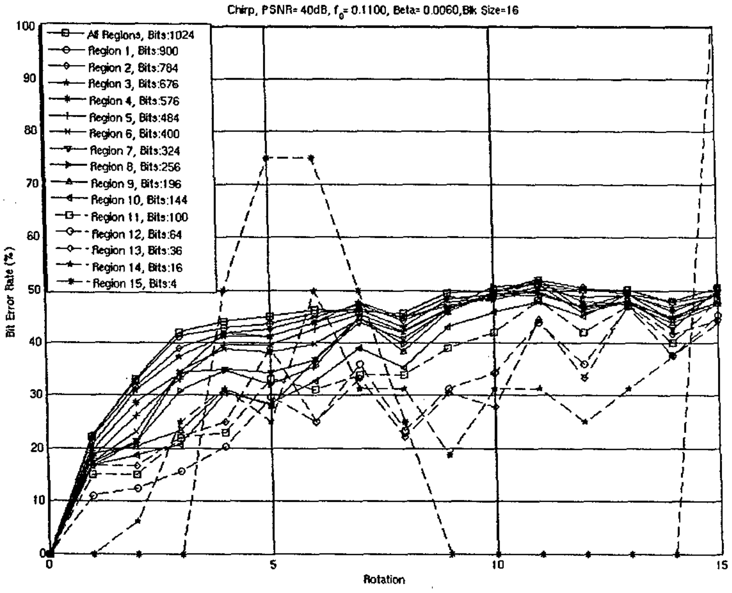


Figure 21: Regional Capacity BER Plots for 16x16 Blocks

All of the block sizes in the figures above have exhausted all possible removable regions. The resulting graphs display consistent improvement for block sizes 128x128 and 32x32 converging to a BER of 0%, while the other two block sizes where vastly different. This shows that the diminishing Regional Block Capacity does work, but there is another variable that is causing inconsistent bursts like the one for the 128x128 block rotated by 10^0. These phenomena also occurred in the original BER vs Rotation graph in Figure 14, which means these inconsistencies are a product of the chirp watermark's structure.

7.4. Rotation Induced Chirp Interference

The chirp's watermark structure is not problematic when referring to the results of the whole image watermark. This means that chirp watermark alone has none of these phenomena, and therefore, must involve the interaction of the blocks with each other during rotation.

To limit the variables the following test results will involve a 512x512 image with a block sizes of 256x256. This will produce 4 blocks around the image. The BER output from the BER plot in Figure 14 shows that this block structure was receiving perfect results until it was rotated 6 degrees, so the figure below will display the watermark image rotated by 6 degrees[7].

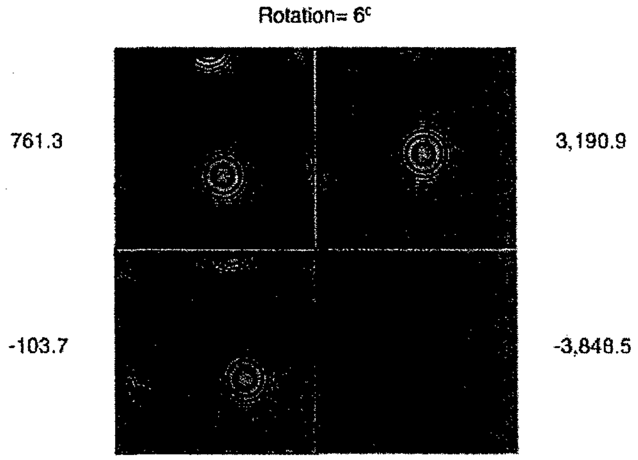


Figure 22: Chirp Rotated 6 Degrees with Correlation Output

The image above also displays the correlation outputs next to its corresponding block after the mean was removed. From the correlation output with respect to the detector, it is apparent that the lower left block is the first block to lose correlation. The upper left block has also lost some significant correlation, but not as severe as the lower left block. The difference between these two blocks and the others is another chirp block rotates into them. This means that when a block with a chirp rotates into another; the resulting block loses more correlation than if a blank block rotated into it. The most

[7] Code for this image is in section 8.5

probable cause for this loss in correlation is the highlighted seams between the chirp blocks seen in the figure below[7].

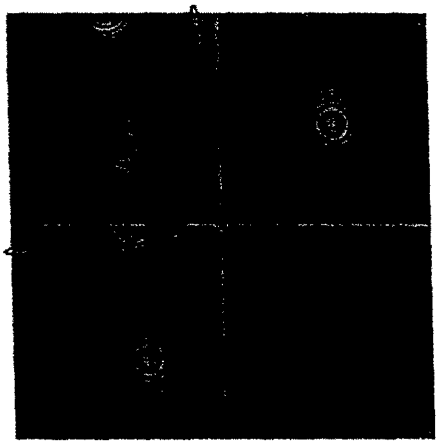


Figure 23: Chirp Rotation Block Seams

To investigate this problem further a block with a chirp in it was divided into four cases: Chirp-Chirp, Chirp-Blank, Blank-Chirp, and Blank-Blank. A Chirp-Chirp is when the block rotating in and the block the current chirp is rotating into are both chirp blocks. A Chirp-Blank is when the block rotating in is a chirp and the block the current chirp is rotating into is blank. A Blank-Chirp is when the block rotating in is blank and the current chirp is rotating into a chirp block. A Blank-Blank is when the block rotating in and the block the current chirp is rotating into are both blank blocks. It turns out that the seams between the chirp blocks are having an effect on the correlation, because the cases involving a chirp rotating perform the worst. The table below describes lists all of the cases form best to worst case.

Chirp Cases	Performance
Blank-Blank	Best
Blank-Chirp	Middle Best
Chirp-Chirp	Middle Worst
Chirp-Blank	Worst

Table 3: Chirp Interference Performance

When these same cases are applied to a blank block, all blocks perform perfectly within the 0-15 degree range. This means the detector for this test has a tendency to get only false-negatives and no false-positives. It also helps justify the theory that these chirp seams are causing the loss in correlation, since the blank blocks have no discontinuities like the chirp blocks and it performs perfectly.

8. Proposed Solutions

There are several possible solutions to this problem, and the first involves creating a whole image watermark and zeroing out only the watermark coefficients specific to blocks representing zero bits. Performing this method would recreate the block-based chirp watermark structure, but it would remove the seams created by the individual block implementations. This first possible solution alone will continue to help justify if the seams created by the rotation of two adjacent chirp blocks into each other are a major contributor to this loss in correlation. The only draw back of the first implementation is it will most likely lose the option of the chirp's natural encryption, because it will only have one set of tunable values.

Another possible solution would suggest manipulating the chirp's tunable variables to better align adjacent watermarked blocks and removing the seams. This idea comes from the regional capacity result of block size 128x128, which avoided chirp interference after significant rotation. These partially aligned chirp blocks can be seen the figure below.

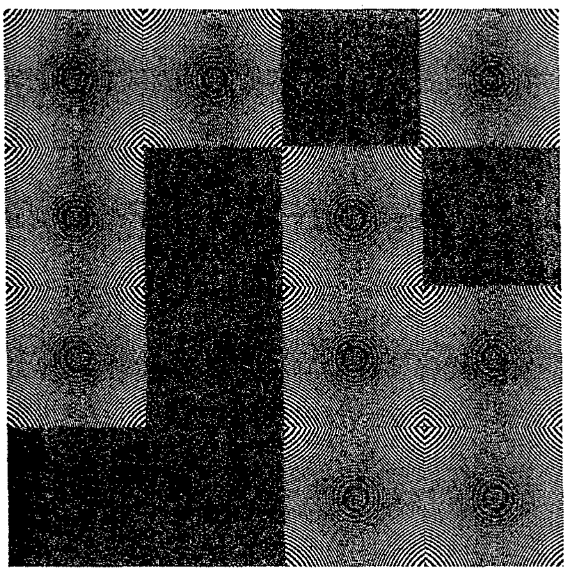


Figure 24: Partially Aligned Chirp Watermarks

The second approach is preferable because it still allows encryption while the first case does not. The only draw back is that manipulating the tunable values to align the chirps would limit the total possible combinations of tunable values. Future work will look into this problem of rotation induced chirp interference and the implementation of the theories mention previously.

9. Conclusions

The status of the project in relationship to its goals is ongoing. Even though a rotational resistant approach for a block-based spatial watermark has not been concluded, the research performed in this paper is optimistic in developing a solution to this problem. The successful rotational invariance of an authentication chirp watermark shows the effectiveness of the FMT approach. The information gathered from the study of rotational displacement will act as a helpful benchmark for the performance of this algorithm. The identification of rotation induced chirp interference and its effects on the loss of correlation is also significant evidence to a solution. These essential and hopeful findings will lead to a successful rotationally invariant data-hiding algorithm for visually unnoticeable rotations.

10.References

[1] Dogahe, Behzad M. "Applications of time-frequency signal representation to watermarking and over-the-horizon radar", Villanova University, 2004.

[2] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking of Multimedia," *IEEE Transactions on Image Processing*, 6, 12, 1673-1687, 1997.

[3] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," Proc. IEEE, vol. 87, no. 7, pp. 1127-1141, July 1999.

[4] Stankovic', I. Djurovic, and I. Pitas, "Watermarking in the space/spatial-frequency domain using two-dimensional radon-wigner distribution," *IEE Trans. Image Processing*, vol. 10, no. 4, pp. 650-658, Apr. 2001

[5] J. R. Hernadez, M. Amando and F. Perez-Gonzalez, "DCT-domain watermarking techniques for still images: detector performance analysis and a new structure," *IEEE Trans. Image Processing*, vol. 9, pp. 55-68, Jan. 2000.

[6] J. J. K. O'Ruanaidh, and T. Pun, "Rotation, Translation and Scale Invariant Spread Spectrum Digital Image Watermarking", Signal Processing, 66, pp.303-318, 1998.

[7] Woo, Cheng-Seng, Jiang Du, Binh Phan, "Geometrically Rubost Digital Image Watermarking using Scale Normalization and Flowline Curvature," *IS&T/SPIE Conference*, Jan. 2005.

[8] Kelly, Roberto M., H. S. Salguero, and E. S. Salguero, "Image Recognition Using the Fourier-Mellin Transform," *Instituto Politeco Naciopnal Mexico,* Mexico.

[9] Zheng, Dong, Jiying Zhao, and Abdulmotaleb El Saddik, "RST Invariant Digital Image Watermarking Based Log-Polar Mapping and Phase Correlation," IEE Trans. Circuits and Sys., vol. 20, no. 9, September 2003.

11. Appendix

11.1. TradCompare.m

```
clc
clear all
close all

load ChirpBER2D chirpErr
load ChirpBER2D pnErr
load ChirpBER2D im
load lenaTradSpectral err16x16
% load ChirpBER2D compChirpImage
% load ChirpBER2D compPNimage

qfactor= 5:5:100;
nmChirpErr= chirpErr./1024*100;%Normalized
nmPNspat= pnErr./1024*100;%Normalized
nmPNspect= err16x16*100;

figure(1)
plot(qfactor,nmPNspat,'-bs',qfactor,nmPNspect,'-gs',qfactor,nmChirpErr,'-rs')
title(sprintf('Chirp vs Traditional, %s,PSNR= 40dB',im))
xlabel('Q Factor')
ylabel('Bit Error Rate (%)')
legend('Spatial PN','Spectral PN','Chirp')
grid on
```

11.2. RotationBER40dB.m

```
clear all
close all

load vu_seal
seal = vu_seal;

eps=1.e-06;
cmprate = 50;

%Variables: sync, psnr, qfactor
sync= 1;
qfactor= 50;
PSNR = 40;%38.59;
wmag = 255*10^(-PSNR/20)*sqrt(2);

beta_range = [0:.001:.025];
f0_rng = [0:0.01:.5]; %2D range

N = 512; % size of picture (one side)
M = 32;  % size of wm (one side)
R = 8;   % size of pixel block (one side)

K = N/R/M; % number of blocks / wm (linear: actual number of blocks is K^2)
KR = K * R;
KR2 = (K*R)^2;  % total size of the wm block
```

```
M0=M;

x_tot = zeros(M0*KR);

rand('seed',100);

minpair = zeros(16,6);

t1=[0:15].';
t2 = t1;

% figure
% imagesc(abs(specwm))

% figure
% imagesc(real(specwm))
%
% figure
% imagesc(imag(specwm))

[T1,T2]=meshgrid(0:15,0:15);

varerror = ones(14,13)*10000000;

%errormat = zeros(19,18);
%errormatcmp = zeros(19,18);


f0 = f0_rng(12);
beta = beta_range(7);

w1= exp(j*pi*beta*(T1.^2 + T2.^2) + j*2*pi*f0*(T1 + T2))*wmag;
w1 = real(w1);
w1 = round(w1 - mean(mean(w1)));

10*log10(255^2/(sum(sum(abs(w1).^2))/256))

%     figure(1)
%     subplot(121)
%     imagesc(w1)
%     colormap(gray)
%     title('Chirp Watermark')
%     axis square
%     axis off
%     xlabel('16')
%     ylabel('16')
%     subplot(122)
%     imagesc(wpn)
%     colormap(gray)
%     title('PN Watermark')
%     axis square
%     axis off
%     xlabel('16')
%     ylabel('16')

%load lena_gray
```

```
% image = imread('LenaGray512.tif');
% image = double(image);
image= zeros(N,N);
% image = image - round(mean(mean(image)));

  iblock = 0;
  for iloop=1:M0;
    for jloop=1:M0;

    iblock = iblock + 1;

    tmp01 = (iloop-1)*KR+[1:KR];
    tmp02 = (jloop-1)*KR+[1:KR];

    s = image(tmp01,tmp02);  %original picture block

    y = tfrdc_2Dnewchirp(s,t1,beta,f0);
    spec(iblock) = y;

    %w = w * seal(iloop,jloop);
    %w = real(w1*seal(iloop,jloop));
    %w = round(w - mean(mean(w)));

    wChirp = w1*seal(iloop,jloop);
    xChirp = s + wChirp;

    y = tfrdc_2Dnewchirp(xChirp,t1,beta,f0);
    specadd(iblock) = y;

    xChirp_tot(tmp01,tmp02) = xChirp;
    %tf2(iloop,jloop) = sign(real(tfrdc_2Dnewchirp(x,t1,beta,f0)))/2+0.5;

     end
   end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%
 pheta= 0:1:10;
 blkS=16;
 if(sync==1)
   sideBS= 81;
 else
   sideBS= 76;
 end

 newM0= floor((N-(2*sideBS))/blkS);
 limitBS= newM0*blkS+sideBS-1;
 for counter=1:length(pheta)
   m= imrotate(xChirp_tot,pheta(counter),'bicubic','crop');
   nonBS(:,:,counter)= m(sideBS:limitBS,sideBS:limitBS);
 %    figure(1)
 %    imagesc(nonBS)
 %   colormap(gray)
 %   axis square
 %   axis off
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%
```

```
  imwrite(uint8(nonBS(:,:,counter)),'lenaRotateChirp.jpg','jpeg','Quality',qfactor)
  compChirpImage(:,:) = double(imread('lenaRotateChirp.jpg'));

  for iloop=1:newM0
    for jloop=1:newM0

    tmp01 = (iloop-1)*KR+[1:KR];
    tmp02 = (jloop-1)*KR+[1:KR];

    compChirpBlock = compChirpImage(tmp01,tmp02);
    tfChirp(iloop,jloop) = sign(real(tfrdc_2Dnewchirp(compChirpBlock,t1,beta,f0)))/2+0.5;
    end
  end
  if(sync==1)
    chirpErr(counter)= sum(sum(abs((seal(6:26,6:26) > 0)*1 - tfChirp(:,:))));%Sychronized
  else
    chirpErr(counter)= sum(sum(abs((seal(6:27,6:27) > 0)*1 - tfChirp(:,:))));%desychronized
  end
end%angle rotation loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nmChirpErr= chirpErr./(newM0*newM0)*100;%Normalized

% Displaying rotated images in pairs
% for picO=0:2:8
% for picI=1:2
% figure(1)
% subplot(1,2,picI)
% imshow(nonBS(:,:,picO+picI),[])
% axis square
% axis off
% end
% pause
% end

% % Displays original image vs 1 degee of rotation
% figure(1)
% subplot(1,2,1)
% imshshow(nonBS(:,:,1),[])
% axis square
% axis off
% xlabel(sprintf('BER= %2.3f',nmChirpErr(1)))
% title('Origianl')
% subplot(1,2,2)
% imshow(nonBS(:,:,2),[])
% axis square
% axis off
% xlabel(sprintf('BER= %2.3f',nmChirpErr(2)))
% title(sprintf('Rotated by %d degree(s)',1))


figure(1)
plot(pheta,nmChirpErr,'-rs')
% set(gca,'XTick',[0:15:360])
set(gca,'XTick',[0:1:10])
```

```
axis tight
title(sprintf('Rotated Lena w/ Chirp, PSNR= 40dB, f_0= %2.4f, Beta= %2.4f, Q=
%d',f0,beta,qfactor))
xlabel('Rotation angle')
ylabel('Bit Error Rate (%)')
grid on
set(gca,'GridLineStyle','-')

% temp= [1:11]*(nan)
% for a=1:11
% figure(1)
% temp(a)= nmChirpErr(a);
% plot(pheta,temp,'-rs')
% axis([0 10 0 55])
% title(sprintf('Rotated Lena w/ Chirp, PSNR= 40dB, f_0= %2.4f, Beta= %2.4f, Q=
%d',f0,beta,qfactor))
% xlabel('Rotation angle')
% ylabel('Bit Error Rate (%)')
% grid on
% set(gca,'GridLineStyle','-')
% pause
% end

% if(sync==1)
%     save rotateBERsync50;%Sychronized
% else
%     save rotateBERdsync50;%desychronized
% end

% save ChirpBER chirpErr, pnErr, compChirpImage, compPNimage;
```

11.3. ShiftLenaBER.m

```
%Christopher Fleming
%This program takes the Lena image and shifts it determine its effects on BER
clear all
close all

% load vu_seal
% seal = vu_seal;

%random seal
seal= rand(32);
seal(find(seal>=.5))=1;
seal(find(seal<.5))=-1;

eps=1.e-06;
cmprate = 50;

%Variables: psnr, qfactor
qfactor= 50;
PSNR = 40;%38.59;
wmag = 255*10^(-PSNR/20)*sqrt(2);

beta_range = [0:.001:.025];
f0_rng = [0:0.01:.5]; %2D range
```

```
N = 512; % size of picture (one side)
M = 32;  % size of wm (one side)
R = 8;   % size of pixel block (one side)

K = N/R/M; % number of blocks / wm (linear: actual number of blocks is K^2)
KR = K * R;
KR2 = (K*R)^2;  % total size of the wm block
M0=M;

rand('seed',100);

t1=[0:15].';
t2 = t1;

[T1,T2]=meshgrid(0:15,0:15);
f0 = f0_rng(12);
beta = beta_range(7);

w1= exp(j*pi*beta*(T1.^2 + T2.^2) + j*2*pi*f0*(T1 + T2))*wmag;
w1 = real(w1);
w1 = round(w1 - mean(mean(w1)));

%Disp PSNR in cmd window
10*log10(255^2/(sum(sum(abs(w1).^2))/256))

%     figure
%     subplot(121)
%     imagesc(w1)
%     colormap(gray)
%     title('Chirp Watermark')
%     axis square
%     axis off
%     xlabel('16')
%     ylabel('16')
%     subplot(122)
%     imagesc(wpn)
%     colormap(gray)
%     title('PN Watermark')
%     axis square
%     axis off
%     xlabel('16')
%     ylabel('16')

%load lena_gray
image = imread('LenaGray512.tif');
image = double(image);
% image = image - round(mean(mean(image)));

%%%%%%%%%%%%%%%%%%% Create Watermark
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  iblock = 0;
  for iloop=1:M0;
    for jloop=1:M0;

    iblock = iblock + 1;
```

```
        tmp01 = (iloop-1)*KR+[1:KR];
        tmp02 = (jloop-1)*KR+[1:KR];

        s = image(tmp01,tmp02);  %original picture block

        y = tfrdc_2Dnewchirp(s,t1,beta,f0);
        spec(iblock) = y;

        %w = w * seal(iloop,jloop);
        %w = real(w1*seal(iloop,jloop));
        %w = round(w - mean(mean(w)));

        wChirp = w1*seal(iloop,jloop);
        xChirp = s + wChirp;

        y = tfrdc_2Dnewchirp(xChirp,t1,beta,f0);
        specadd(iblock) = y;

        xChirp_tot(tmp01,tmp02) = xChirp;
        %tf2(iloop,jloop) = sign(real(tfrdc_2Dnewchirp(x,t1,beta,f0)))/2+0.5;

        end
      end

    %%%%%%%%%%%%%%%%%%%% Shift Image
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    shiftMax= 70;
    shift= 0:1:shiftMax;
    blkS=16;

    sideBS= 81;
    newM0= floor((N-(2*sideBS))/blkS);
    limitBS= newM0*blkS+sideBS-1;
    % scaleIm= image(sideBS:limitBS+shiftMax+1,sideBS:limitBS+shiftMax+1);
    % scaleIm= image(sideBS:limitBS,sideBS:limitBS);
    % [sR,sC]= size(scaleIm);
    % shiftM= ones(sR,sC,length(shift));

    for shiftC= 1:length(shift)
    %     shiftIm=
    scaleIm(1+shift(shiftC):limitBS+shift(shiftC),sideBS+shift(shiftC):limitBS+shift(shiftC));
       shiftIm(:,:,shiftC)= xChirp_tot(
    (sideBS+shift(shiftC)):(limitBS+shift(shiftC)),(sideBS+shift(shiftC)):(limitBS+shift(shiftC)) );
    %     figure
    %     imshow(shiftM,[])
    %   axis square
    %   axis off

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%
       imwrite(uint8(shiftIm(:,:,shiftC)),'lenaShiftChirp.jpg','jpeg','Quality',qfactor)
       compChirpImage = double(imread('lenaShiftChirp.jpg'));

       for iloop=1:newM0
          for jloop=1:newM0
      tmp01 = (iloop-1)*KR+[1:KR];
      tmp02 = (jloop-1)*KR+[1:KR];

      compChirpBlock = compChirpImage(tmp01,tmp02);
      tfChirp(iloop,jloop) = sign(real(tfrdc_2Dnewchirp(compChirpBlock,t1,beta,f0)))/2+0.5;
      end
   end%detection loop
   chirpErr(shiftC)= sum(sum(abs((seal(6:26,6:26) > 0)*1 - tfChirp(:,:))));%Sychronized
end%shift loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nmChirpErr= chirpErr./(newM0*newM0)*100;%Normalized

% figure(1)
% subplot(1,2,1)
% imshow(shiftIm(:,:,1),[])
% axis square
% axis off
% xlabel(sprintf('BER= %2.3f',nmChirpErr(1)))
% title('Origianl')
% subplot(1,2,2)
% imshow(shiftIm(:,:,2),[])
% axis square
% axis off
% shiftDisp=1;
% xlabel(sprintf('BER= %2.3f',nmChirpErr(shiftDisp+1)))
% title(sprintf('Shifted by %d pixel(s)',shiftDisp))

figure(2)
plot(shift,nmChirpErr,'-rs')
% set(gca,'XTick',[0:10])
title(sprintf('Shifted Lena w/ Chirp, PSNR= 40dB, f_0= %2.4f, Beta= %2.4f, Q=
%d',f0,beta,qfactor))
xlabel('Shifted Pixel(s)')
ylabel('Bit Error Rate (%)')
grid on
set(gca,'GridLineStyle','-')

% figure(3)
% for shiftC= 1:5:length(shift)
%    imshow(shiftIm(:,:,shiftC),[]);
%    pause
% end

% save shiftBERq50randseal0_50;
% save ChirpBER chirpErr, pnErr, compChirpImage, compPNimage;
```

11.4. LogPolarMapping.m

```
function output =logpolarmap(m,mode)
%logpolarmap.m computes either forward or inverse log-polar map. The Fourier-Mellin
%transform can then be obtained by running FFT2 on the LPM array. LPM array is saved
%in a file for further processing. If rotation, translation or scaling is applied to input image
```

```
%pad the image first.
%     m: input image(square).
%     mode: string to specify forward or inverse LPM.
%     usage: logpolarmap(image_array,'forward') for forward LPM and
%     logpolarmap(lpm_image,'inverse') for inverseLPM. For consistent
%     results use logpolarmap.m and invert_lpm.m together.
%(c) 2005 Bijan G. Mobasseri & Christopher E. Fleming

%normalize image
m=m./max(m(:));

%setup variables
n=length(m);
center=(n+1)/2;
d=(n-1)/2;
r= logspace(0,log10(d),n);
theta = linspace(0,2*pi,n+1); theta(end) = [];

%coordinate conversion produces nonuniform grid
x=r'*cos(theta)+center;
y=r'*sin(theta)+center;

if strmatch(mode,'forward')
   %do LPM
   %find pixel values on uniform grid
   output= interp2(m,x,y,'cubic');
   save logpolarForw
elseif strmatch(mode,'reverse')
   % Reverse Interpolation w/ griddata
   c= linspace(1,n,n);
   r= linspace(1,n,n);
   [yC,xC]= meshgrid(r,c);
   output= griddata(x,y,m,yC,xC,'cubic');
   %Add a NaN removal algorithm
   save logpolarRev
else
   error('Please Enter a String: reverse or Forward')
end

% %dispaly images
% subplot(211)
% imshow(m,[])
% title('ORIGINAL','fontsize', 14)
% subplot(212)
% imshow(m_lpm,[])
% title('LPM OF ORIGINAL','fontsize', 14)
% %horizontal is theta, vertical is rho
% %origin is at upper left left
```

11.5. ChirpOnlyRotDisp.m

```
%Christopher E. Fleming
% This program uses a watermark with no cover image, therefore it is the
% strongest possible signal. It displays a rotated version of the watermark
% dependent on its rotation

clear all
close all

blkSize= 256;
region = 0;
rotAngle= 6;

if(blkSize == 512 || blkSize == 256)
   region = 0;
end

fName= cat(2,'rotBER10Chirp40dB_Blk',num2str(blkSize),'Region',num2str(region));

load(fName)
% figure
% imshow(chirpOnly,[]);
% title('Original')
%
% figure
% gridIm= blkGrid(KR,rotChirp(:,:,rotAngle+1));
% imshow(gridIm,[]);
% title('Rotated, No Regions Removed')
%
% figure
% gridRegion= blkGrid(KR,nonBS(:,:,rotAngle+1));
% imshow(gridRegion,[]);
% title(sprintf('Grid Synchronized Looking Glass, Angle= %d, Blk= %d, Region=
%d',rotAngle,KR,region));

% Display any blocksize rotated with a grid
figure
gridRegion= blkGrid(KR,imrotate(chirpOnly,rotAngle,'bicubic','crop'));
imshow(gridRegion,[]);
```

11.6. BERRegionSingBlk.m

```
% Christopher E. Fleming
% April 2005
% Chirp Only BER Plots w/o Black Space. it uses synchronized regions of the
% image. Displays several region removals for a single blk size.
clear, close all

blk= 16;
angles= [0:1:15];

if(blk==128)
   load rotBER10Chirp40dB_Blk128Region0 chirpErr
   load rotBER10Chirp40dB_Blk128Region0 chirpOnly
   load rotBER10Chirp40dB_Blk128Region0 detSeal
   errReg0= chirpErr;
   chirpReg0= chirpOnly;
   [r,c,ang]= size(detSeal);
   reg0bits= r*c;
   load rotBER10Chirp40dB_Blk128Region1 chirpErr
   load rotBER10Chirp40dB_Blk128Region1 chirpOnly
   load rotBER10Chirp40dB_Blk128Region1 detSeal
   errReg1= chirpErr;
   chirpReg1= chirpOnly;
   [r,c,ang]= size(detSeal);
   reg1bits= r*c;

   load rotBER10Chirp40dB_Blk128Region1 PSNR
   load rotBER10Chirp40dB_Blk128Region1 f0
   load rotBER10Chirp40dB_Blk128Region1 beta

   figure
   plot(angles,errReg0*100,'-ks',angles,errReg1*100,'-rs')
   legend(sprintf('Region 0, Bits:%d',reg0bits),sprintf('Region 1, Bits:%d',reg1bits))
   title(sprintf('Chirp, PSNR= %ddB, f_0= %2.4f, Beta= %2.4f,Blk
Size=%d',PSNR,f0,beta,blk))
   xlabel('Rotation')
   ylabel('Bit Error Rate (%)')
   grid on
   set(gca,'GridLineStyle','-')
 elseif(blk==64)
   load rotBER10Chirp40dB_Blk64Region0 chirpErr
   load rotBER10Chirp40dB_Blk64Region0 chirpOnly
   load rotBER10Chirp40dB_Blk64Region0 detSeal
   errReg0= chirpErr;
   chirpReg0= chirpOnly;
   [r,c,ang]= size(detSeal);
   reg0bits= r*c;

   load rotBER10Chirp40dB_Blk64Region1 chirpErr
   load rotBER10Chirp40dB_Blk64Region1 chirpOnly
   load rotBER10Chirp40dB_Blk64Region1 detSeal
   errReg1= chirpErr;
   chirpReg1= chirpOnly;
   [r,c,ang]= size(detSeal);
   reg1bits= r*c;

   load rotBER10Chirp40dB_Blk64Region2 chirpErr
   load rotBER10Chirp40dB_Blk64Region2 chirpOnly
   load rotBER10Chirp40dB_Blk64Region2 detSeal
   errReg2= chirpErr;
   chirpReg2= chirpOnly;
   [r,c,ang]= size(detSeal);
   reg2bits= r*c;

   load rotBER10Chirp40dB_Blk64Region3 chirpErr
   load rotBER10Chirp40dB_Blk64Region3 chirpOnly
   load rotBER10Chirp40dB_Blk64Region3 detSeal
   errReg3= chirpErr;
   chirpReg3= chirpOnly;
   [r,c,ang]= size(detSeal);
   reg3bits= r*c;

   load rotBER10Chirp40dB_Blk64Region3 PSNR
   load rotBER10Chirp40dB_Blk64Region3 f0
```

```
  load rotBER10Chirp40dB_Blk64Region3 beta

  figure
  plot(angles,errReg0*100,'-ks',angles,errReg1*100,'-rs',angles,errReg2*100,'-
bs',angles,errReg3*100','-gs')
  legend(sprintf('Region 0, Bits:%d',reg0bits),sprintf('Region 1,
Bits:%d',reg1bits),sprintf('Region 2, Bits:%d',reg2bits),sprintf('Region 3, Bits:%d',reg3bits))
  title(sprintf('Chirp, PSNR= %ddB, f_0= %2.4f, Beta= %2.4f,Blk
Size=%d',PSNR,f0,beta,blk))
  xlabel('Rotation')
  ylabel('Bit Error Rate (%)')
  grid on
  set(gca,'GridLineStyle','-')
elseif(blk==32)
  load rotBER10Chirp40dB_Blk16Region0 PSNR
  load rotBER10Chirp40dB_Blk16Region0 f0
  load rotBER10Chirp40dB_Blk16Region0 beta

  load rotBER10Chirp40dB_Blk32Region0 chirpErr
  load rotBER10Chirp40dB_Blk32Region0 chirpOnly
  load rotBER10Chirp40dB_Blk32Region0 detSeal
  errReg0= chirpErr;
  chirpReg0= chirpOnly;
  [r,c,ang]= size(detSeal);
  reg0bits= r*c;

  load rotBER10Chirp40dB_Blk32Region1 chirpErr
  load rotBER10Chirp40dB_Blk32Region1 chirpOnly
  load rotBER10Chirp40dB_Blk32Region1 detSeal
  errReg1= chirpErr;
  chirpReg1= chirpOnly;
  [r,c,ang]= size(detSeal);
  reg1bits= r*c;

  load rotBER10Chirp40dB_Blk32Region2 chirpErr
  load rotBER10Chirp40dB_Blk32Region2 chirpOnly
  load rotBER10Chirp40dB_Blk32Region2 detSeal
  errReg2= chirpErr;
  chirpReg2= chirpOnly;
  [r,c,ang]= size(detSeal);
  reg2bits= r*c;

  load rotBER10Chirp40dB_Blk32Region3 chirpErr
  load rotBER10Chirp40dB_Blk32Region3 chirpOnly
  load rotBER10Chirp40dB_Blk32Region3 detSeal
  errReg3= chirpErr;
  chirpReg3= chirpOnly;
  [r,c,ang]= size(detSeal);
  reg3bits= r*c;

  load rotBER10Chirp40dB_Blk32Region4 chirpErr
  load rotBER10Chirp40dB_Blk32Region4 chirpOnly
  load rotBER10Chirp40dB_Blk32Region4 detSeal
  errReg4= chirpErr;
  chirpReg4= chirpOnly;
  [r,c,ang]= size(detSeal);
```

```
reg4bits= r*c;

load rotBER10Chirp40dB_Blk32Region5 chirpErr
load rotBER10Chirp40dB_Blk32Region5 chirpOnly
load rotBER10Chirp40dB_Blk32Region5 detSeal
errReg5= chirpErr;
chirpReg5= chirpOnly;
[r,c,ang]= size(detSeal);
reg5bits= r*c;

load rotBER10Chirp40dB_Blk32Region6 chirpErr
load rotBER10Chirp40dB_Blk32Region6 chirpOnly
load rotBER10Chirp40dB_Blk32Region6 detSeal
errReg6= chirpErr;
chirpReg6= chirpOnly;
[r,c,ang]= size(detSeal);
reg6bits= r*c;

load rotBER10Chirp40dB_Blk32Region7 chirpErr
load rotBER10Chirp40dB_Blk32Region7 chirpOnly
load rotBER10Chirp40dB_Blk32Region7 detSeal
errReg7= chirpErr;
chirpReg7= chirpOnly;
[r,c,ang]= size(detSeal);
reg7bits= r*c;

 figure
 plot(angles,errReg0*100,'-ks',angles,errReg1*100,'-rs',angles,errReg2*100,'-
bs',angles,errReg3*100','-gs',angles,errReg4*100','-ms',angles,errReg5*100,'-
ys',angles,errReg6*100,'-cs',angles,errReg7*100,'-kx')
  legend(sprintf('Region 0, Bits:%d',reg0bits),sprintf('Region 1,
Bits:%d',reg1bits),sprintf('Region 2, Bits:%d',reg2bits),sprintf('Region 3,
Bits:%d',reg3bits),sprintf('Region 4, Bits:%d',reg4bits),sprintf('Region 5,
Bits:%d',reg5bits),sprintf('Region 6, Bits:%d',reg6bits),sprintf('Region 7, Bits:%d',reg7bits))
  title(sprintf('Chirp, PSNR= %ddB, f_0= %2.4f, Beta= %2.4f,Blk
Size=%d',PSNR,f0,beta,blk))
  xlabel('Rotation')
  ylabel('Bit Error Rate (%)')
  grid on
  set(gca,'GridLineStyle','-')
elseif(blk==16)
  load rotBER10Chirp40dB_Blk16Region0 PSNR
  load rotBER10Chirp40dB_Blk16Region0 f0
  load rotBER10Chirp40dB_Blk16Region0 beta

  load rotBER10Chirp40dB_Blk16Region0 chirpErr
  load rotBER10Chirp40dB_Blk16Region0 chirpOnly
  load rotBER10Chirp40dB_Blk16Region0 detSeal
  errReg0= chirpErr;
  chirpReg0= chirpOnly;
  [r,c,ang]= size(detSeal);
  reg0bits= r*c;

  load rotBER10Chirp40dB_Blk16Region1 chirpErr
  load rotBER10Chirp40dB_Blk16Region1 chirpOnly
  load rotBER10Chirp40dB_Blk16Region1 detSeal
```

```
errReg1= chirpErr;
chirpReg1= chirpOnly;
[r,c,ang]= size(detSeal);
reg1bits= r*c;

load rotBER10Chirp40dB_Blk16Region2 chirpErr
load rotBER10Chirp40dB_Blk16Region2 chirpOnly
load rotBER10Chirp40dB_Blk16Region2 detSeal
errReg2= chirpErr;
chirpReg2= chirpOnly;
[r,c,ang]= size(detSeal);
reg2bits= r*c;

load rotBER10Chirp40dB_Blk16Region3 chirpErr
load rotBER10Chirp40dB_Blk16Region3 chirpOnly
load rotBER10Chirp40dB_Blk16Region3 detSeal
errReg3= chirpErr;
chirpReg3= chirpOnly;
[r,c,ang]= size(detSeal);
reg3bits= r*c;

load rotBER10Chirp40dB_Blk16Region4 chirpErr
load rotBER10Chirp40dB_Blk16Region4 chirpOnly
load rotBER10Chirp40dB_Blk16Region4 detSeal
errReg4= chirpErr;
chirpReg4= chirpOnly;
[r,c,ang]= size(detSeal);
reg4bits= r*c;

load rotBER10Chirp40dB_Blk16Region5 chirpErr
load rotBER10Chirp40dB_Blk16Region5 chirpOnly
load rotBER10Chirp40dB_Blk16Region5 detSeal
errReg5= chirpErr;
chirpReg5= chirpOnly;
[r,c,ang]= size(detSeal);
reg5bits= r*c;

load rotBER10Chirp40dB_Blk16Region6 chirpErr
load rotBER10Chirp40dB_Blk16Region6 chirpOnly
load rotBER10Chirp40dB_Blk16Region6 detSeal
errReg6= chirpErr;
chirpReg6= chirpOnly;
[r,c,ang]= size(detSeal);
reg6bits= r*c;

load rotBER10Chirp40dB_Blk16Region7 chirpErr
load rotBER10Chirp40dB_Blk16Region7 chirpOnly
load rotBER10Chirp40dB_Blk16Region7 detSeal
errReg7= chirpErr;
chirpReg7= chirpOnly;
[r,c,ang]= size(detSeal);
reg7bits= r*c;

load rotBER10Chirp40dB_Blk16Region8 chirpErr
load rotBER10Chirp40dB_Blk16Region8 chirpOnly
load rotBER10Chirp40dB_Blk16Region8 detSeal
```

```
errReg8= chirpErr;
chirpReg8= chirpOnly;
[r,c,ang]= size(detSeal);
reg8bits= r*c;

load rotBER10Chirp40dB_Blk16Region9 chirpErr
load rotBER10Chirp40dB_Blk16Region9 chirpOnly
load rotBER10Chirp40dB_Blk16Region9 detSeal
errReg9= chirpErr;
chirpReg9= chirpOnly;
[r,c,ang]= size(detSeal);
reg9bits= r*c;

load rotBER10Chirp40dB_Blk16Region10 chirpErr
load rotBER10Chirp40dB_Blk16Region10 chirpOnly
load rotBER10Chirp40dB_Blk16Region10 detSeal
errReg10= chirpErr;
chirpReg10= chirpOnly;
[r,c,ang]= size(detSeal);
reg10bits= r*c;

load rotBER10Chirp40dB_Blk16Region11 chirpErr
load rotBER10Chirp40dB_Blk16Region11 chirpOnly
load rotBER10Chirp40dB_Blk16Region11 detSeal
errReg11= chirpErr;
chirpReg11= chirpOnly;
[r,c,ang]= size(detSeal);
reg11bits= r*c;

load rotBER10Chirp40dB_Blk16Region12 chirpErr
load rotBER10Chirp40dB_Blk16Region12 chirpOnly
load rotBER10Chirp40dB_Blk16Region12 detSeal
errReg12= chirpErr;
chirpReg12= chirpOnly;
[r,c,ang]= size(detSeal);
reg12bits= r*c;

load rotBER10Chirp40dB_Blk16Region13 chirpErr
load rotBER10Chirp40dB_Blk16Region13 chirpOnly
load rotBER10Chirp40dB_Blk16Region13 detSeal
errReg13= chirpErr;
chirpReg13= chirpOnly;
[r,c,ang]= size(detSeal);
reg13bits= r*c;

load rotBER10Chirp40dB_Blk16Region14 chirpErr
load rotBER10Chirp40dB_Blk16Region14 chirpOnly
load rotBER10Chirp40dB_Blk16Region14 detSeal
errReg14= chirpErr;
chirpReg14= chirpOnly;
[r,c,ang]= size(detSeal);
reg14bits= r*c;

load rotBER10Chirp40dB_Blk16Region15 chirpErr
load rotBER10Chirp40dB_Blk16Region15 chirpOnly
load rotBER10Chirp40dB_Blk16Region15 detSeal
```

```
  errReg15= chirpErr;
  chirpReg15= chirpOnly;
  [r,c,ang]= size(detSeal);
  reg15bits= r*c;

  figure
  plot(angles,errReg0*100,'-ks',angles,errReg1*100,'-rs',angles,errReg2*100,'-
bs',angles,errReg3*100','-gs',angles,errReg4*100','-ms',angles,errReg5*100,'-
ys',angles,errReg6*100,'-cs',angles,errReg7*100,'-kx',angles,errReg8*100,'-
rx',angles,errReg9*100,'-bx',angles,errReg10*100,'-gx',angles,errReg11*100','-
mx',angles,errReg12*100,'-yx',angles,errReg13*100,'-cx',angles,errReg14*100,'-
ko',angles,errReg15*100,'-ro')
   legend(sprintf('Region 0, Bits:%d',reg0bits),sprintf('Region 1,
Bits:%d',reg1bits),sprintf('Region 2, Bits:%d',reg2bits),sprintf('Region 3,
Bits:%d',reg3bits),sprintf('Region 4, Bits:%d',reg4bits),sprintf('Region 5,
Bits:%d',reg5bits),sprintf('Region 6, Bits:%d',reg6bits),sprintf('Region 7,
Bits:%d',reg7bits),sprintf('Region 8, Bits:%d',reg8bits),sprintf('Region 9,
Bits:%d',reg9bits),sprintf('Region 10, Bits:%d',reg10bits),sprintf('Region 11,
Bits:%d',reg11bits),sprintf('Region 12, Bits:%d',reg12bits),sprintf('Region 13,
Bits:%d',reg13bits),sprintf('Region 14, Bits:%d',reg14bits),sprintf('Region 15,
Bits:%d',reg15bits))
   title(sprintf('Chirp, PSNR= %ddB, f_0= %2.4f, Beta= %2.4f,Blk
Size=%d',PSNR,f0,beta,blk))
   xlabel('Rotation')
   ylabel('Bit Error Rate (%)')
   grid on
   set(gca,'GridLineStyle','-')
else
   error('NO DATA')
end
```

11.7. InvarChirpOnly.m

```
%Christopher E Fleming

close all
clear all

NAME= 'Chirp';
PSNR= 40;

N = 512; % size of picture (one side)
R = 8;   % size of pixel block (one side)
% size of wm matrix(one side)
% M = 32; %16x16
% M = 16; %32x32
% M = 8;  %64x64
% M = 4;  %128x128
M = 2;  %256x256
% M = 1;  %512x512

K = N/R/M; % number of blocks / wm (linear: actual number of blocks is K^2)
KR = K * R;
KR2 = (K*R)^2;  % total size of the wm block
```

```
M0=M;

region=0;
%Only certain block sizes can specific regions
% region=0: all Blk(includes black space)
% region=1: 128x128- 16x16
% region=2-3: 64x64- 16x16
% region=4-7: 32x32- 16x16

rand('seed',100);
if KR== N
   seal=1;
   bits= '10';
else
   % load vu_seal
   % seal = vu_seal; % VU seal
   % seal= sign(rand(M,M) - .5); % random seal {1,-1}
   % bits= '1-1';
   seal= abs(sign(sign(rand(M,M) - .5)-1)); % random seal {1,0}
   bits= '10';
end


%Re-creates a block(16x16) of the chirp
beta_range = [0:.001:.025];
% f0_rng = [0:0.01:.5]; %2D range
f0_rng = [0:0.01:.25]; %better range
f0 = f0_rng(12);
beta = beta_range(7);

[T1,T2]=meshgrid(0:KR-1,0:KR-1);
%Image block Watermarks
wmag = 255*10^(-PSNR/20)*sqrt(2);
w= exp(j*pi*beta*(T1.^2 + T2.^2) + j*2*pi*f0*(T1 + T2))*wmag;
w = real(w);
w = round(w - mean(mean(w)));

%Comparison Watermark
w1= exp(j*pi*beta*(T1.^2 + T2.^2) + j*2*pi*f0*(T1 + T2))*1;%*wmag;
w1 = real(w1);
w1 = round(w1 - mean(mean(w1)));

lopoW= logpolarmap(w1,'forward');
lopoW(find(isnan(lopoW)))=0;
invaW= abs(fft2(lopoW));

figure(1);
subplot(121); imshow(w,[])
title(sprintf('Image Block Chirp, PSNR= %ddB', PSNR))
subplot(122); imshow(w1,[])
title(sprintf('Comparison Chrip'))


%%%%% Embeding
for iloop=1:M0;
   for jloop=1:M0;
```

```
  tmp01 = (iloop-1)*KR+[1:KR];
  tmp02 = (jloop-1)*KR+[1:KR];

  chirpOnly(tmp01,tmp02)= w*seal(iloop,jloop);
%    lopoChirpOnly(tmp01,tmp02)= logpolarmap(wChirp,'forward');%Log Polar Chirp only
%    lopoChirpOnly(find(isnan(lopoChirpOnly)))=0;
%    invaChirp(tmp01,tmp02)= abs(fft2(lopoChirpOnly(tmp01,tmp02)));% FFT
  end
end

figure(2); imshow(chirpOnly,[]); title(sprintf('Chirp, Blk= %d, PSNR= %ddB',KR,PSNR));

%%%%% Rotation & Detection
% angles= 360-[0:1:15];%CW
angles= [0:1:15];%CCW
for rot= 1:length(angles)
  posC=0;
  negC=0;
  rotChirp(:,:,rot)= imrotate(chirpOnly,angles(rot),'bicubic','crop');%Chirp Only Uncompressed

  %Removes Blackspace for specifc block sizes
  if(KR==512 || KR== 256)
    region=0;
    endL=M0;
    newM0= M0;
    nonBS(:,:,rot)=rotChirp(:,:,rot);
  else
    sideBS= KR*region;%81;%16x16
    newM0= floor((N-(2*sideBS))/KR);
    limitBS= newM0*KR+sideBS;
    nonBS(:,:,rot)= rotChirp(sideBS+1:limitBS,sideBS+1:limitBS,rot);
    endL= newM0;
%      [R,C]= size(rotCompChirp);
  end
% % Displays both versions of rotation: one as the normal version and the as the looking
glass.
%    figure
%    subplot(121); imshow(rotChirp(:,:,rot),[]);
%    title(sprintf('Synchronized Looking Glass, Angle= %d, Blk= %d, Region=
%d',angles(rot),KR,region));
%    subplot(122); imshow(blkGrid(KR,nonBS),[]);

% % Displays the rotated blocks with a grid
% figure
% gridIm= blkGrid(KR,nonBS);
% imshow(gridIm,[]);
% title(sprintf('Grid Synchronized Looking Glass, Angle= %d, Blk= %d, Region=
%d',angles(rot),KR,region));

  for iloop=1:endL
    for jloop=1:endL
      tmp01 = (iloop-1)*KR+[1:KR];
      tmp02 = (jloop-1)*KR+[1:KR];

      rotLopoChirp= logpolarmap(nonBS(tmp01,tmp02,rot),'forward');
```

```
          rotLopoChirp(find(isnan(rotLopoChirp)))=0;
          fRotLopoChirp= abs(fft2(rotLopoChirp));

          %Without Noise(aka Cover Image)
          [rTemp,cTemp]= size(fRotLopoChirp);
          cox(iloop,jloop,rot)=
dot((fRotLopoChirp),invaW(1:rTemp,1:cTemp))/sqrt(dot((fRotLopoChirp),(fRotLopoChirp)));

%           if(seal(iloop,jloop)==1)
%               posC= posC + 1;
%               posSeal(posC,rot)= cox(iloop,jloop,rot);
%          else
%               negC= negC + 1;
%               negSeal(negC,rot)= cox(iloop,jloop,rot);
%           end
%
%           wmCor= xcorr2(fRotLopoChirp,invaW);
%           maxCor(iloop,jloop,rot)= max(max(wmCor));
%           figure
%           mesh(wmCor(iloop,jloop,rot))
%           title(sprintf('Correlation On-off, Blk=%d, Bit=%d, Q=%d,
Rot=%d',KR,seal(iloop,jloop),qfactor,angles(rot)));
%           xlabel(sprintf('max= %d', max(max(wmCor(iloop,jloop,rot)))))
%           ylabel(sprintf('X= %d, Y= %d', iloop,jloop))
%           if(maxCor >= 5e+08)
%               detSeal(iloop,jloop)= 1;
%           else
%               detSeal(iloop,jloop)= 0;
%           end
        end%End of Column Loop
      end%End of Row Loop

      if(KR~=512)
         cox0(:,:,rot)= cox(:,:,rot)- mean(mean(cox(:,:,rot)));%Zero mean or Zero threshold
      else
         cox0(:,:,rot)= cox(:,:,rot);
      end

      detSeal(:,:,rot)= zeros(endL,endL);
      [posR,posC]= find(cox0(:,:,rot) > 0);
      for setB= 1:length(posR)
      detSeal(posR(setB),posC(setB),rot)= 1;
      end

      %Plot Distribution
    %   figure(5)
    %   subplot(121); hist(negSeal(:,rot),50); title('Negative Ones');
    %   xlabel(sprintf('%s %ddB, BLK=%d, lw Compressed',NAME,PSNR,KR))
    %   subplot(122); hist(posSeal(:,rot),50); title('Positive Ones');
    %   xlabel(sprintf('Both Distibutions, Angle= %d',angles(rot)))

    chirpErr(rot)=length(find((detSeal(:,:,rot) - seal(1+region:end-region,1+region:end-
    region))~=0))/(newM0*newM0);
    end% End Rotation Loop
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%
```

```
figure
plot(angles,chirpErr*100,'-rs')
% axis([0 330  0 100])
title(sprintf('%s, PSNR= %ddB, Blk=%d, f_0= %2.4f, Beta= %2.4f,
Region=%d',NAME,PSNR,KR,f0,beta,region))
xlabel('Rotation')
ylabel('Bit Error Rate (%)')
grid on
set(gca,'GridLineStyle','-')

save(cat(2,'rotBER',bits,NAME,num2str(PSNR),'dB_Blk',num2str(KR),'Region',num2str(regio
n)));
```

11.8. BlkGrid.m

```
function grdIm= blkGrid(blkSize,im)

% blkSize:  size of the one side of the square block
% im:       Image the grid will be located on

[R,C]= size(im);
r= R/blkSize;
c= C/blkSize;
gColor= max(max(im))-min(min(im));

for iloop=1:r;
   for jloop=1:c;

   tmp01 = (iloop-1)*blkSize+[1:blkSize];
   tmp02 = (jloop-1)*blkSize+[1:blkSize];

   im(tmp01(end),tmp02)= gColor;
   im(tmp01,tmp02(end))= gColor;
   end
end

grdIm= im;
```

11.9. BERregionsAllBlks.m

```
% Christopher E. Fleming
% April 2005
% Chirp Only BER Plots w/o Black Space. it uses synchronized regions of the image
clear, close all

region=1;
angles= [0:1:15];

if(region==1)
   load rotBER10Chirp40dB_Blk16Region1 chirpErr
   load rotBER10Chirp40dB_Blk16Region1 chirpOnly
   errBlk16= chirpErr;
   chirpBlk16= chirpOnly;
   load rotBER10Chirp40dB_Blk32Region1 chirpErr
   load rotBER10Chirp40dB_Blk32Region1 chirpOnly
   errBlk32= chirpErr;
   chirpBlk32= chirpOnly;

   load rotBER10Chirp40dB_Blk64Region1 chirpErr
   load rotBER10Chirp40dB_Blk64Region1 chirpOnly
   errBlk64= chirpErr;
   chirpBlk64= chirpOnly;

   load rotBER10Chirp40dB_Blk128Region1 chirpErr
   load rotBER10Chirp40dB_Blk128Region1 chirpOnly
   errBlk128= chirpErr;
   chirpBlk128= chirpOnly;

   load rotBER10Chirp40dB_Blk128Region1 PSNR
   load rotBER10Chirp40dB_Blk128Region1 f0
   load rotBER10Chirp40dB_Blk128Region1 beta

   figure
   plot(angles,errBlk16*100,'-rs',angles,errBlk32*100,'-bs',angles,errBlk64*100,'-
gs',angles,errBlk128*100,'-cs')
   legend('16x16','32x32','64x64','128x128')
   title(sprintf('Chirp, PSNR= %ddB, f_0= %2.4f, Beta=
%2.4f,Region=%d',PSNR,f0,beta,region))
   xlabel('Rotation')
   ylabel('Bit Error Rate (%)')
   grid on
   set(gca,'GridLineStyle','-')
 elseif(region==2)
   load rotBER10Chirp40dB_Blk16Region2 chirpErr
   load rotBER10Chirp40dB_Blk16Region2 chirpOnly
   errBlk16= chirpErr;
   chirpBlk16= chirpOnly;

   load rotBER10Chirp40dB_Blk32Region2 chirpErr
   load rotBER10Chirp40dB_Blk32Region2 chirpOnly
   errBlk32= chirpErr;
   chirpBlk32= chirpOnly;

   load rotBER10Chirp40dB_Blk64Region2 chirpErr
   load rotBER10Chirp40dB_Blk64Region2 chirpOnly
   errBlk64= chirpErr;
   chirpBlk64= chirpOnly;

   load rotBER10Chirp40dB_Blk64Region2 PSNR
   load rotBER10Chirp40dB_Blk64Region2 f0
   load rotBER10Chirp40dB_Blk64Region2 beta

   figure
   plot(angles,errBlk16*100,'-rs',angles,errBlk32*100,'-bs',angles,errBlk64*100,'-gs')
   legend('16x16','32x32','64x64')
   title(sprintf('Chirp, PSNR= %ddB, f_0= %2.4f, Beta=
%2.4f,Region=%d',PSNR,f0,beta,region))
   xlabel('Rotation')
   ylabel('Bit Error Rate (%)')
```

```
  grid on
  set(gca,'GridLineStyle','-')
elseif(region==3)
  load rotBER10Chirp40dB_Blk16Region3 chirpErr
  load rotBER10Chirp40dB_Blk16Region3 chirpOnly
  errBlk16= chirpErr;
  chirpBlk16= chirpOnly;

  load rotBER10Chirp40dB_Blk32Region3 chirpErr
  load rotBER10Chirp40dB_Blk32Region3 chirpOnly
  errBlk32= chirpErr;
  chirpBlk32= chirpOnly;

  load rotBER10Chirp40dB_Blk64Region3 chirpErr
  load rotBER10Chirp40dB_Blk64Region3 chirpOnly
  errBlk64= chirpErr;
  chirpBlk64= chirpOnly;

  load rotBER10Chirp40dB_Blk64Region3 PSNR
  load rotBER10Chirp40dB_Blk64Region3 f0
  load rotBER10Chirp40dB_Blk64Region3 beta

  figure
  plot(angles,errBlk16*100,'-rs',angles,errBlk32*100,'-bs',angles,errBlk64*100,'-gs')
  legend('16x16','32x32','64x64')
  title(sprintf('Chirp, PSNR= %ddB, f_0= %2.4f, Beta=
%2.4f,Region=%d',PSNR,f0,beta,region))
  xlabel('Rotation')
  ylabel('Bit Error Rate (%)')
  grid on
  set(gca,'GridLineStyle','-')
elseif(region==4)
  load rotBER10Chirp40dB_Blk16Region4 chirpErr
  load rotBER10Chirp40dB_Blk16Region4 chirpOnly
  errBlk16= chirpErr;
  chirpBlk16= chirpOnly;

  load rotBER10Chirp40dB_Blk32Region4 chirpErr
  load rotBER10Chirp40dB_Blk32Region4 chirpOnly
  errBlk32= chirpErr;
  chirpBlk32= chirpOnly;

  load rotBER10Chirp40dB_Blk32Region4 PSNR
  load rotBER10Chirp40dB_Blk32Region4 f0
  load rotBER10Chirp40dB_Blk32Region4 beta

  figure
  plot(angles,errBlk16*100,'-rs',angles,errBlk32*100,'-bs')
  legend('16x16','32x32')
  title(sprintf('Chirp, PSNR= %ddB, f_0= %2.4f, Beta=
%2.4f,Region=%d',PSNR,f0,beta,region))
  xlabel('Rotation')
  ylabel('Bit Error Rate (%)')
  grid on
  set(gca,'GridLineStyle','-')
else
```

```
     error('NO DATA')
   end
```

Applications of Two-Dimensional Chirps

Electrical and Computer Engineering Department and
Center for Advanced Communications
Villanova University

By
Behzad Mohammadi Dogahe

Under the Direction of Dr. Bijan G. Mobasseri

Acknowledgment

I would like to express my gratitude to my advisors, Prof. Moeness Amin and Prof. Yimin Zhang, for their productive guidance, understanding, and encouragement in my research throughout my Master's program. I would also like to thank Prof. Bijan Mobasseri for his guidance in the digital watermarking research. I would like to thank all the faculty members and staffs of Villanova Univeristy.

This work was supported by the Office of Naval Research. I would like to express my gratefulness to ONR for supporting this project. Also, I would like to thank the Villanova University and Prof. Amin for granting me the opportunity of being a part of this project.

Thesis Outline

This thesis considers two problems in time-frequency signal representation applications. In both problems, the linear frequency modulated (LFM) signals or chirp signals are focused.

In Chapter 1, a novel linear time-frequency representation method is proposed for source detection and classification in over-the-horizon radar (OTHR) systems. Of particular interest is the estimation and identification of the multi-paths of maneuvering targets described as multi-component time-Doppler signatures in the presence of strong clutter. By approximating the time-Doppler signatures as chirp signals in each block of time period, chirp signal analysis is used to estimate the chirp parameters (chirp rates and center frequencies) of the clutter and target signal components. Clutter components, which are localized around the low chirp rates and frequencies, are effectively suppressed through subspace projections. The target signals are then dechirped, which is followed by either conventional DFT transform or high-resolution spectrum estimation methods for center frequency estimation.

In Chapter 2, the concept of nonstationary signal processing developed for OTHR is extended to the two-dimensional (2-D) signal waveforms for the applications of digital watermarking. The hidden information is embedded into a picture using the phase information of a 2-D chirp prototype. The original picture is properly partitioned into several blocks. In each block, one or several 2-D chirp watermark waveforms are embedded. The parameters of the chirp watermark are selected in order to achieve low bit error rate (BER) detection of the watermark using the 2-D chirp transform. Chirp parameter optimization incorporates the 2-D chirp transform of the image blocks. The use of multiple, rather than a single, chirp waveforms for each image block provides robustness to the randomness of the image and the distortion caused by compression and various attacks. Experimental results compare the proposed methods and support their effectiveness.

The above two problems constitute Chapters 1 and 2. Each chapter has its own introduction and references. At the end of this thesis, the publication associated with this research work is provided.

This research work was funded by a fund from the Office of Naval Research (ONR).

Chapter 1
Time-Frequency Analysis for Maneuvering Target Detection in Over-the-Horizon Radars

I. Introduction

Over-the-horizon radar (OTHR) systems, operating at the high-frequency (HF) bands to exploit the reflective and refractive nature of radio propagation through ionosphere, perform wide-area surveillance at long range well beyond the limit of the horizon and provide advance warning which directly translates into reaction time [1], [2], [3].

An important problem in OTHR is robust high-resolution Doppler processing of targets. This resolution problem is complicated for accelerating or decelerating targets where the Doppler characteristics are time-varying, which arises during aircraft and ship target maneuver and during observations of rockets during boost phase and mid-course flight.

As reported in [4] and will be briefly discussed in Section 2, the complex Doppler signatures presented in these cases reveal important information about the target. For accelerating/decelerating targets, classical Doppler processing techniques introduce smearing in Doppler spectrum which reduces resolution and can obscure important multi-component Doppler features [5]. The purpose of this chapter is to achieve high-resolution of time-varying Doppler spectrum using linear time-frequency representation techniques.

Because of the existence of strong clutter, however, direct application of various time-frequency distributions (TFDs) often fails to achieve desirable resolution and concentration. In [4], we have proposed a novel estimation method based on effective clutter suppression and data-dependent bilinear TFDs, combined with robust high-resolution analysis. Inspired by the linear time-frequency signal representation methods proposed in [5] for weak signal detection and Doppler signature estimation, we use in this chapter the linear chirp transform and decomposition in place of bilinear transforms. We take full advantage of distinct chirp rates and frequency shifts characteristics between the target signals and the clutter. The proposed method provides improved clutter suppression performance and robust target signal characteristics estimation over the existing methods.

II. Signal Model

The received OTHR signal, after waveform dechirping at the receiver, is expressed as

$$y(t) = x(t) + u(t), \tag{1}$$

where $x(t)$ is the single- or multi-component return signal from the target, and $u(t)$ is the clutter which also includes the additive noise.

In a typical OTHR scenario, in addition to the path directly reflected from the ionosphere, there is a multipath due to additional reflections from the ground or sea near the target. Denote l_1 and l_2 as the propagation distance of the two paths, respectively, and d_t and d_r as the respective one-way slant range between the transmitter and the target and between the target and the receiver, respectively. Then, d_t takes the value of either l_1 and l_2 and so does d_r (In a backscattering OTHR system, the range of a target is slightly different when viewed from the transmitter and the receiver. However, without loss of generality, we assume identical values for notation simplicity). Therefore, the received signal consists of four combination paths which result in the following three multipath components:

$$x(t) = A_1 e^{-j\omega 2l_1/c} + A_2 e^{-j\omega 2l_2/c} + A_3 e^{-j\omega(l_1+l_2)/c}, \tag{2}$$

where c denotes the speed of light, $\omega = 2\pi f_c$ is the carrier radian frequency, and A_1, A_2, and A_3 are the corresponding path losses.

In this chapter, we consider a well encountered scenario of a maneuvering aircraft as an example. In this case, the target makes a 180° turn within a $T = 30.72$ second duration to change the height and direction. The time-Doppler signatures is plotted in Fig. 1, where the parameters used in the analysis and simulations are listed in Table 1. All the multipath signals are considered to fall in the same range cell.

The dominant Doppler component is proportional to the target velocity in the slant range direction, and the small Doppler difference between the three paths is proportional to the ascending velocity of the target. The Doppler difference between the three paths, therefore, reveals important information on how the target moves in the elevation direction. The maximum one-side Doppler difference corresponding to the maximum ascending speed 1500 m/min = 25 m/s is about 1.17 Hz.

In this chapter, we consider a more challenging scenario than that considered in [4]. The signal level is reduced by 6 dB whereas the clutter and noise remain unchanged. The fourth revisit (block 4) consisting of 256 samples from the 5.12 second period between $t = 15.36$ and $t = 20.48$ seconds. In the raw clutter data used in this chapter, the nominal clutter frequency being about 1 Hz (refer to Fig. 2). Therefore, block 4 is more significantly jammed by the clutter.

TABLE I

MAJOR PARAMETERS

Parameter	Notation	Value
initial range	$R(0)$	2000 km
ionosphere height	H	350 km
aircraft initial height	$h(0)$	10000 m
maximum range speed	v	500 km/hr
maximum climbing speed	v_c	1500 m/min
carrier frequency	f_c	20 MHz
repetition frequency	f_s	50 Hz
samples per block	N	256 samples

III. Clutter Suppression

The clutter is suppressed in two phases. The first phase considers the clutter as an autoregressive (AR) model. Considerable part of the clutter energy can be mitigated by the AR pre-whitening approach [4]. To further reduce the clutter, we propose to use chirp decomposition to estimate and subtract the residual clutter component.

A. AR Pre-Whitening

We point to the fact that the clutter is highly localized in low frequencies and can be well modelled as an autoregressive (AR) process. Denote P as the order of the AR model, the AR polynomial parameters $a(t), t = 0, ..., P$ can be estimated via the modified covariance method [6].

Filtering the received signal $y(t)$ through a finite impulse filter (FIR), constructed using the AR polynomial parameters, results in the pre-whitened signal:

$$z(t) = y(t) * a(t) = z_x(t) + z_u(t), \tag{3}$$

where "*" denotes the convolution operator.

In this chapter, the target signal modeled in Section 2 is overlayed on real OTHR clutter data. The order of the AR model should be chosen to maximize the signal-to-clutter ratio (SCR). We set the order of the AR model to a unit value (P=1). The spectrogram of block 4 before and after the AR pre-whitening is shown in Fig. 2. It is seen that, the AR pre-whitening substantially suppresses the clutter by more than 40 dB, and the target signal is also affected with different levels, depending on how close its Doppler frequency is to the nominal frequency of the clutter.

When the signal level is moderate over the noise floor, AR pre-whitening with higher order than one may undesirably mitigate the target signal. Therefore, the AR pre-whitening usually brings the clutter level comparable to that of the target signal. Such clutter level might still be considered high for proper estimation of the parameters of the target signals, particularly when the target signals have close signature to the clutter.

In the following, further and key improvements in resolutions of the target signature components can be accomplished using the chirp decomposition techniques.

B. Chirp Decomposition and Projection

A. Chirp Transform

The chirp transform of signal $x(t)$ is defined as

$$X(\beta, f) = \int_{-\infty}^{\infty} x(t)w(t)e^{-j\beta(t-t_0)^2/2-j2\pi f(t-t_0)}dt \tag{4}$$

where β is the chirp rate of the chirp transform, and $w(t)$ is the window function. In Eq. (4), t_0 is used to shift the time origin. For the finite samples used in the computation, we choose t_0 as half of the length of the data samples, so that the frequency shift is evaluated at the center of each chirp signal component. Moreover, the Hamming window is applied in the computations.

When applied to a chirp signal with chirp rate β_0 and initial frequency shift f_0, the transform $X(\beta, f)$ will show a high peak at the corresponding chirp rate β_0 and center frequency shift f_0. The location of the peak is used to estimate the chirp rate and center frequency shift of both the clutter components and the target signals. In the following, we use the chirp transform magnitude with respect to the chirp rate β and the center frequency f, to illustrate the clutter suppression procedure.

B. Chirp Transform Based Clutter Suppression

To remove the clutter, we take advantage of the fact that the clutter is highly localized in both low frequencies and low chirp rates. The chirp transform magnitude of the OTHR signal after pre-whitening is plotted in Fig. 3. The localized clutter is evident in this figure. The existence of clutter often obscures the identification of target signals as well as the estimation of chirp rate. To mitigate the clutter components, therefore, we proceed to find the strongest chirp signal component by searching the peak of the chirp transform magnitude. The basis function of the associated chirp signal component is then estimated as

$$\hat{v}(t) = e^{j\hat{\beta}(t-t_0)^2/2+j2\pi\hat{f}(t-t_0)} \tag{5}$$

where $\hat{\beta}$ and $\hat{f}$ are the estimated chirp rate and initial frequency, respectively. The strength and phase of the associated chirp signal component is obtained by projecting the data signal onto the signal subspace spanned by $\hat{v}(t)$. This process is expressed in the following vector form,

$$\mathbf{x}_v = (\hat{\mathbf{v}}^H\hat{\mathbf{v}})^{-1}\hat{\mathbf{v}}\hat{\mathbf{v}}^H\mathbf{x}. \tag{6}$$

This signal component is then subtracted from the data signal,

$$\mathbf{x}_r = \mathbf{x} - \mathbf{x}_v, \tag{7}$$

which is used for further chirp decomposition. By repeating the above process described by Eqs. (4)–(7), we can obtain the dominant chirp components from the data signal.

We define a small window around the point $(f = 0, \beta = 0)$ and the chirp signals falling in this window will be considered as clutter components. Therefore, the signal after clutter removal becomes the sum of the residual signal component and all estimated chirp signal components except those being classified as clutter, i.e.,

$$\mathbf{x}_0 = \mathbf{x}_r + \sum_{\text{not clutter}} \mathbf{x}_v. \tag{8}$$

In Fig. 4, we show the reconstructed signal after ten iterations of chirp decompositions with the clutter components removed. It is evident that, the clutter is suppressed considerably. The maximum value of the chirp transform magnitude of the reconstructed signal in Fig. 4 clearly provides the chirp rate, as indicated by the horizontal line.

In order to resolve the center frequencies of the signal components sharing the same chirp rate, we plot the spectrum of Fig. 4 along the horizontal line indicated. The results are shown in Fig. 5 and reveal two neighboring peaks to the highest value in the plot. The above combined information of the chirp rate and the center frequencies can easily translate to the time-varying Doppler signatures of the three target signal components shown in Fig. 1.

IV. Conclusion

In this chapter, a novel method has been proposed for high-resolution time-Doppler signature localization applied to over-the-horizon radar systems. By combining AR prewhitening and chirp decomposition for effective clutter suppression, the proposed method provides robust estimation of time-varying Doppler signature in low signal-to-clutter ratio (SCR) scenarios.

References

[1] J. M. Headrick and M. I. Skolnik, "Over-the-horizon radar in the HF band," *Proc. IEEE*, vol. 62, pp. 664–673, 1974.

[2] G. D. McNeal, "The high-frequency environment at the ROTHR Amchitka radar site," *Radio Science*, vol. 30, pp. 739–746, May–June 1995.

[3] J. M. Headrick and J. F. Thomason, "Naval applications of high frequency over-the-horizon radar," *Nav. Eng. J.*, vol. 108, pp. 353–359, May 1996.

[4] Y. Zhang, M. G. Amin, and G. J. Frazer, "High-resolution time-frequency distributions for maneuvering target detection in over-the-horizon radars," in *Proc. ICASSP*, Hong Kong, April 2003.

[5] G. Wang, X.-G. Xia, B. Root, V. Chen, Y. Zhang, and M. G. Amin, "Manuvering target detection in over-the-horizon radar by using adaptive chirplet transform and subspace clutter rejection," in *Proc. ICASSP*, Hong Kong, April 2003.

[6] S. M. Kay, *Modern Spectral Estimation: Theory and Applications*, Englewood Cliffs, NJ: Prentice Hall, 1998.

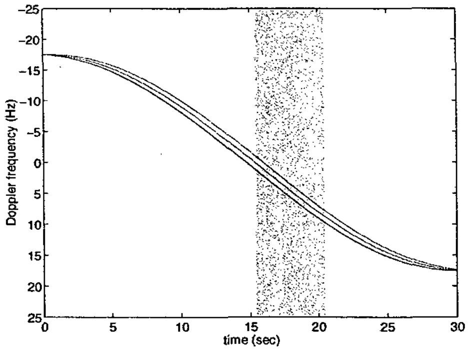


Fig. 1. Time-Doppler signature (block 4 is shaded).

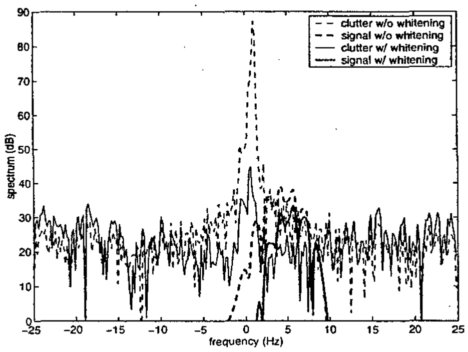


Fig. 2. Block-wise spectrogram of the received signal before and after AR pre-whitening.

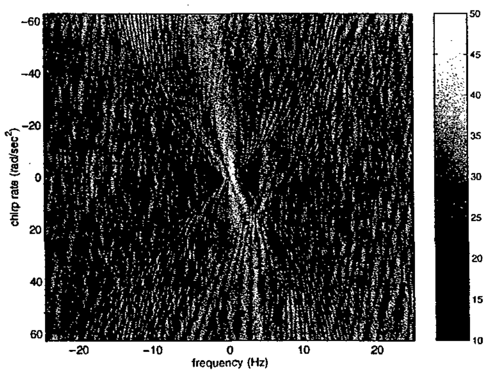


Fig. 3. Chirp transform magnitude of OTHR Signal after AR pre-whitening.

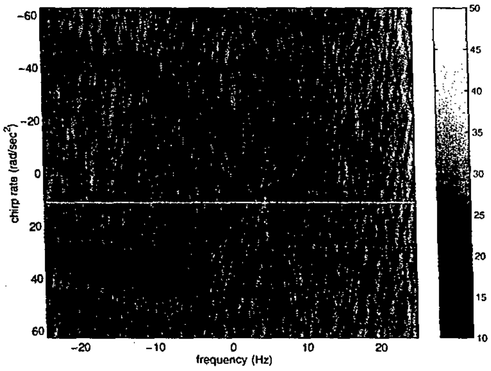


Fig. 4. Chirp transform magnitude of OTHR signal after 10 iterations of chirp decomposition and clutter removal. The line shows the estimated chirp rate of the target signals.

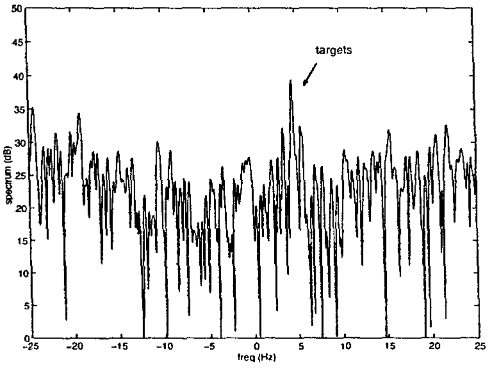


Fig. 5. Spectrum of OTHR after clutter suppression and dechirping.

Chapter 2
Digital Watermarking Using Two-Dimensional Chirps

I. Introduction

Digital watermarking is the process of securely embedding invisible signatures within a cover media with no perceptual impact. Depending on the application, this process is also referred to as data hiding or steganography. Data hiding, if used as a means for covert communications, may require a heavier embedding capacity than digital watermarking. The cover media is of primary interest in watermarking whereas in data hiding the cover media is only useful to the extent that it provides a container for hidden communications. Developing any new watermarking algorithm requires the definition of five components, i.e., 1) cover media, 2) watermark, 3) embedding and extraction, 4) perceptual metric, and 5) resilience and security criteria.

Watermarking has been performed in spectral as well as spatial domain. Arguably, the best known watermarking technique is spread spectrum (SS). SS watermarking has been used in several different contexts. One of the earliest references to SS watermarking is due to Cox [1]. The watermark is drawn from a Gaussian source and additively modifies the full frame discrete Fourier transform (DFT) transform coefficients of the image. Watermarked portions of the DFT consist of the perceptually significant transform coefficients. Hernandez et. al [2] have applied the same idea to 8 × 8 block discrete cosine transform (DCT) transform of images, closely following the JPEG standard. First, the watermark is mapped to a one-dimensional (1-D) binary vector. Then, a 2-D binary mask is generated by an expansion process by repeating each bit of the 1-D vector in different subsets of DCT coefficients. The strength of watermark is driven by a perceptual mask. This approach is similar to spatial domain SS watermarking proposed earlier by Hartung and Girod [3] which did not resort to masking models.

The concept of SS can be applied equally in spectral as well as spatial domains. Case in point is the watermarking model proposed by Kutter and Winkler [4]. The watermark consists of a binary array. Each bit of the array is spread by a 2-D modulation function and added to nonoverlapping sets of image pixels driven by a density metric. This is similar to Honsinger and Rabbani's phase dispersion method [5]. Both Kutter and Honsinger use the same model to spread watermark bits. The former uses a pseudo-random (PN) sequence for the job whereas the latter designs a carrier with flat spectrum but PN phase. The amplitude of the carrier is driven by a perceptual masking profile. SS watermarking model, therefore, is an exercise in selecting the optimum spreading function for a given task.

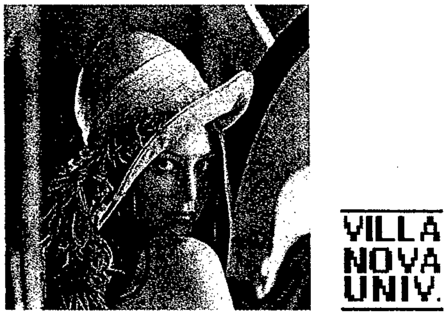
PN spreading sequences are among the earliest spreading functions used in digital watermarking. Although PN sequences provide respectable robustness against malicious attacks through the processing gain, they provide little in terms of spectral shaping. The ability to spectrally shape the watermark allows us to design the spreading function with as little overlap as possible with image data. As importantly, spectral shaping allows for circumventing compression in general and JPEG in particular. Since JPEG compression profile is already known, it is possible to shape the watermark in order to avoid frequency-selective JPEG compression.

In this work we propose a 2-D wideband signal as our choice for spreading function and implement a block-based digital watermarking algorithm. We then evaluate the performance of this function for a special case of 2-D chirp signals. In a prior work, Stankovic et al [6] used chirps as digital watermarks too. They added a chirp to the entire frame then used energy-concentrating property of Radon-Wigner transform to establish the presence of the watermark by peak-searching. This algorithm is best suited to copyright and ownership verification applications where a binary decision is sufficient to establish the presence or absence of the watermark. The ability to embed and detect different chirps per image block, however, allows for data hiding applications where the extracted watermark may be an information-bearing bitstream. Another point of departure from [6] is exploitation of known-host-state-methods [7]. This approach was first suggested by Cox as communication problem with side-information [8] which was in turn based on Costa's dirty paper writing [9]. We have incorporated this idea into our work and show that it is possible to achieve zero BER by exploiting knowledge of host signal at the encoder. This observation is in contrast to SS watermarking and others where watermark structure is unrelated to host signal statistics.

II. Watermark Embedding

Consider a problem that a digital watermark containing N-bit information is to be embedded in a gray-scale image. The image is partitioned into non-overlapping blocks whose sizes depend on the picture size and the amount information to hide. If N_p is the number of bits that a block can host, then $\lceil N/N_p \rceil$ blocks are needed, where $\lceil x \rceil$ denotes the minimum integer equal to or larger than x. In addition, when JPEG image compression is considered, it is preferred that the size of each block be 8×8 or its multiples.

As an example, we consider in this chapter a 32×32 binary seal, which is the name of the authors' affiliation, to be embedded into the 512×512 gray-scale Lena picture (Fig. 1). If the watermark is embedded through binary phase modulations, each block hosts one bit of information. Therefore, we can partition the image into $32 \times 32 = 1024$ blocks, with each block consisting of $16 \times 16 = 256$ pixels.



(a) Gray-scale Lena image (512×512). (b) Binary seal (32×32).

Fig. 1. Original image and binary seal.

In each block (m, n), $m, n = 0, ..., 31$, the watermarked image $G(x, y)$ is expressed as

$$G(m, n, x, y) = I(m, n, x, y) + \mathcal{Q}\{k\Re[s(m, n)W(x, y, \Theta_0)]\} \tag{1}$$

where $I(m, n, x, y)$ is the original image at block (m, n), extending over the spatial axes, x and y, where $x,\ y = 0, \cdots, 15$. In (1), $W(x, y, \Theta_0)$ represents the employed complex 2-D FM waveform basis with Θ_0 representing a set of parameters that defines the 2-D FM waveform basis, and $s(m, n)$ is the information to be mapped into the 2-D waveform basis in block (m, n). When binary phase data modulation is used, $s(m, n)$ takes value of either +1 or −1, corresponding to either 0 (black) or 1 (white) of the seal pixels. The parameter k is introduced to control the image-to-watermark ratio, which is usually referred to as the peak signal-to-noise ratio (PSNR). Moreover, $\Re[\cdot]$ denotes the real-part operator, emphasizing the fact that while the original 2-D FM basis waveform is complex, the hidden information in the image is real. $Q[x] = \lfloor x + 0.5 \rfloor$ denotes quantization operation, where $\lfloor \cdot \rfloor$ stands for rounding down to the nearest integer.

III. Watermark Detection and Recovery

A. Watermark Detection and Parameter Estimation

We consider blind decoding of the watermarked image, that is, the unmarked image is not used in the detection. When the parameters Θ_0 that define the 2-D FM basis waveform are not available at the detector, they must be estimated before detection can be made. On the other hand, when the waveform parameters are known at the detector, this step can be skipped.

The detector first estimates $\hat{\Theta}_0$ by maximizing the following criterion,

$$\hat{\Theta}_0 = \arg\max_{\Theta} |C(m,n,\Theta)|, \tag{2}$$

where

$$\begin{aligned} C(m,n,\Theta) &= \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} G(m,n,x,y)W^*(x,y,\Theta) \\ &= \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} I(m,n,x,y)W^*(x,y,\Theta) + \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} Q\{k\Re[s(m,n)W(x,y,\Theta_0)]\}W^*(x,y,\Theta) \\ &= C_I(m,n,\Theta) + C_W(m,n,\Theta) \end{aligned} \tag{3}$$

In (3),

$$C_I(m,n,\Theta) = \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} I(m,n,x,y)W^*(x,y,\Theta)$$

is the output of the matched filter corresponding to the original image, whereas

$$C_W(m,n,\Theta) = \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} Q\{k\Re[s(m,n)W(x,y,\Theta_0)]\}W^*(x,y,\Theta)$$

is the output corresponding to the watermark.

It is noted that, unlike the conventional communications where the data is often zero-mean, the image in its original format is all non-negative. To avoid any potential bias in the detection, therefore, it is important to remove the DC component from the image before he watermark detection, and it is desirable that the waveform basis is designed to be zero-mean.

Because of the different signatures between the image and the watermark waveform basis when they are projected into the Θ domain, the waveform basis achieves much higher gain through the matched filtering at the detection. When the watermark has enough energy such that $|C_W(m, n, \Theta_0)| > |C_I(m, n, \Theta_0)|$, the waveform parameters Θ_0 may be detected by locating the peak of $C(m, n, \Theta)$.

In practical watermarking applications, however, the low probability of detection is important. For this purpose, the embedded information usually does not have enough strength such that the waveform parameters can be estimated in each partitioned block. When the hidden information is embedded such that

$$|\tilde{C}_W(\Theta_0)| \gg |\tilde{C}_I(\Theta_0)|, \tag{4}$$

in the vicinity of the watermark, away from the region where the image energy is concentrated, then the existence of watermark can be detected and reliable estimation of $\hat{\Theta}_0$ can be made by maximizing the following criterion,

$$\hat{\Theta}_0 = \arg\max_{\Theta} \tilde{C}(\Theta), \tag{5}$$

where

$$\tilde{C}(\Theta) = \sum_m \sum_n |C(m, n, \Theta)|, \tag{6}$$

and $\tilde{C}_I(\Theta)$ and $\tilde{C}_W(\Theta)$ are defined similarly.

B. Watermark Recovery

When the watermark is detected and waveform parameters Θ_0 are known or reliably estimated, the watermark information at block (m, n) can be recovered from the phase information of the matched filter output, i.e., $C(m, n, \Theta_0)$. In particular, when the binary phase modulation is used, the embedded information is estimated as

$$\hat{s}(m,n) = \begin{cases} 1, & \text{if } \Re[C(m,n,\Theta_0)] \geq 0 \\ 0, & \text{if } \Re[C(m,n,\Theta_0)] < 0. \end{cases} \tag{7}$$

IV. Watermark Waveform Design

A. Chirp Transform

In this chapter the means of detecting the watermark prototype, which is a 2-D chirp signal, is 2-D chirp transform. The 2-D chirp transform of a signal $x(t_1, t_2)$, $0 \leq t_1, t_2 \leq T-1$ is defined as

$$X_c(\beta_0, f_0) = \sum_{t_1=0}^{T-1} \sum_{t_2=0}^{T-1} x(t_1, t_2) e^{-j\pi\beta_0(t_1^2+t_2^2)-j2\pi f_0(t_1+t_2)} \tag{8}$$

In this definition we can choose the resolution of β_0 and f_0 based on our desired resolution and there is no limitation as far as the number of points of β_0 and f_0 that we can have. This property of chirp transform is the main reason that we prefer to use for our application as opposed to using discrete chirp Fourier transform (DCFT) [15]. DCFT is defined as

$$X_c(k,l) = \frac{1}{T} \sum_{t_1=0}^{T-1} \sum_{t_2=0}^{T-1} x(t_1, t_2) e^{-j\frac{2\pi}{T}l(t_1^2+t_2^2)-j\frac{2\pi}{T}k(t_1+t_2)}, \qquad 0 \leq k, l \leq T-1 \tag{9}$$

It is clear from the definition that the resolutions of k and l are bounded to the number of points, T.

B. 2-D Chirp Waveform

In this section, we consider the design of watermark waveforms. A 2-D chirp signal is used as a simple example of 2-D FM basis waveform. The 2-D chirp waveform basis, $W(x, y, \beta_x, \beta_y, f_x, f_y)$, in the complex format, is expressed as [10], [6]

$$W(x, y, \beta_y, f_x, f_y) = e^{j\pi(\beta_x x^2+\beta_y y^2)+j2\pi(f_x x+f_y y)}, \tag{10}$$

where β_x and β_y are the chirp rates at the x and y axes, and f_x and f_y are the respective initial frequencies of the chirp signal. These four variables form the waveform parameters, i.e., $\Theta_0 = (\beta_x, \beta_y, f_x, f_y)$. For notational simplicity and without loss of generality, we

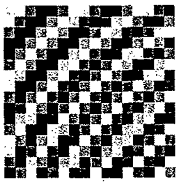


Fig. 2. Quantized 2-D chirp waveform.

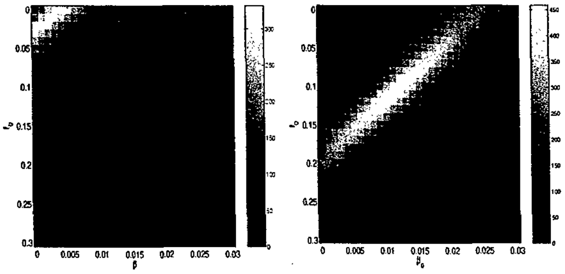


Fig. 3. Chirp transform spectra of a picture block $C_I(\beta, f)$ (left) and the 2-D chirp FM waveform $C_W(\beta, f)$ (right).

consider symmetric cases and denote $\beta_x = \beta_y = \beta_0$, and $f_x = f_y = f_0$, and Θ_0 simplifies to $\Theta_0 = (\beta_0, f_0)$. Accordingly, the 2-D chirp basis function becomes

$$W(x, y) = e^{j\pi\beta_0(x^2+y^2)+j2\pi f_0(x+y)}. \tag{11}$$

The variables x and y take values from $[0, ..., T-1]$, for $T \times T$ blocks, and in this specific example, $T = 15$. Therefore, the instantaneous frequency in (11) ranges from f_0 to $\beta_0(T-1) + f_0$.

The spectrum of the 2-D chirp waveform is important in the performance of the detection and robustness. In designing the FM waveform, the initial frequency and chirp rate are selected such that, at the specific (β_0, f_0), the projection of the image spectrum is relatively low and the chirp is robust against image compression. For this sake, the high-frequency band is first excluded from consideration and then the parameters are optimized by choosing those where the image spectrum is low.

Figure 2 illustrates the quantized waveform of a 2-D chirp. The chirp transform spectrum of the first (upper-left corner) block of the original Lena picture and the 2-D chirp are shown in Fig. 3 where the PSNR is 40 dB. It is clear that the image has a wide spectrum with its peak power located at low frequencies and low chirp rates. On the other hand, the chirp spectrum can be designed to be away from the region where the image spectrum is concentrated.

C. Chirp Parameter Selection

For the binary phase modulation, the probability of erroneous detection, i.e., embedding information s and deciding in favor of $r \neq s$, is given be

$$\begin{aligned} P_e &= P(r \neq s) \\ &= P(s=-1)P(r=+1|s=-1) + \ P(s=+1)P(r=-1|s=+1). \end{aligned} \tag{12}$$

The probability is evaluated for all blocks. The chirp parameter selection, in essence, is to find (β_0, f_0) such that, given an embedding power, the above error probability, i.e., the total error bits divided by the total information bits, can be minimized. It is noted that, although we used the term probability here for convenience, the image information over different blocks is determinant and is known at the embedder. This is in fact the known-host state method that we exploit below.

Therefore, the optimum values of (β_0, f_0) can be selected by searching (β, f) such that the above error probability is minimized. However, an insight look of the decision process can deepen our understanding as well as help us in determining the waveform parameters. In the next, we consider the adaptive chirp power allocation.

D. Adaptive Chirp Power Allocation

The known-host state method allows us to embed the watermark in such a way as to push the decision metric into correct decision region. To ensure correct detection at all blocks, the following condition should be satisfied,

$$C_W(m,n,\beta_0,f_0) \underset{<}{\overset{>}{}} -C_I(m,n,\beta_0,f_0), \text{ if } s(m,n) \underset{<}{\overset{>}{}} 0. \tag{13}$$

The matched filter output of the embedded waveform at block (m,n), $C_W(m,n,\beta_0,f_0)$, takes the form of

$$C_W(m,n,\beta_0,f_0) = s(m,n)H(\beta_0,f_0,k), \tag{14}$$

where $H(\beta_0,f_0,k) = |C_W(m,n,\beta_0,f_0)|$ is the magnitude. We emphasize that H is a function of k as k constitutes an important part of the chirp waveform design.

Substituting (14) to (13) yields,

$$s(m,n)H(\beta_0, f_0, k) \underset{<}{\overset{>}{}} -C_I(m,n,\beta_0,f_0), \text{if } s(m,n) \underset{<}{\overset{>}{}} 0. \tag{15}$$

or, equivalently,

$$H(\beta_0, f_0, k) > -s(m,n)C_I(m,n,\beta_0,f_0). \tag{16}$$

Because both $s(m,n)$ and $C_I(m,n,\beta_0,f_0)$ are known to the embedder, we can choose different values of k at different blocks. That is, at each block (m,n), $k(m,n)$ is chosen to be minimal to maintain error free detection. In particular, when $s(m,n)$ and $C_I(m,n,\beta_0,f_0)$ have the same sign, requirement (16) is always satisfied irrespective of the value of k. Therefore, $k=0$ can be chosen. For this value of k no chirp is actually added to the image block.

It is noted that, however, if the watermark has to be detected blindly, the total watermark energy has to be maintained such that the detection and parameter estimation can be carried out successfully.

E. Multi-Chirp Watermark Waveform Consideration

A generalization of the approach presented in Section IV-B is to use multiple 2-D chirps to form the watermark waveform. In this case, the watermark basis function expression becomes

$$W(x,y) = \sum_{l=1}^{L} W(x,y,\beta_{0,l},f_{0,l}) = \sum_{l=1}^{L} e^{j\pi\beta_{0,l}(x^2+y^2)+j2\pi f_{0,l}(x+y)}, \tag{17}$$

where L is the total number of chirp components. The chirp components have low mutual correlations, and should satisfy the same conditions imposed on the design of a single-chirp watermark.

In detecting the multi-chirp watermark, the 2-D chirp transform of the watermarked 2-D picture $G(m,n,x,y)$ is evaluated at each of the components. The output of the kth component is expressed as

$$C_l(m,n) = \sum_{x=0}^{T-1}\sum_{y=0}^{T-1} G(m,n,x,y)W^*(x,y,\beta_{0,l},f_{0,l}). \tag{18}$$

Several combining methods can be used to incorporate the above outputs corresponding to L chirp components. Depending on different weighting criteria to be used, the combining scheme can be expressed in terms of the following general expression

$$C(m,n) = \sum_{l=1}^{L} |C_l(m,n)|^{\alpha} C_l(m,n), \tag{19}$$

where α is a constant scalar determining the combining weighting.

While the combing methods used in wireless communications for fading mitigation are well known, we emphasize the difference between the underlying watermark detection problem and the wireless communications. In wireless communications, the mean noise power is constant and the strength of the received signal is likely to reflect that of the desired signal. Therefore, stronger components are enhanced because they likely represent the less faded signals. Typically, α takes the value of unity. For the proposed application, however, the watermark power is constant for different components, and the high value of chirp transform is likely to imply a high picture (which acts as noise in the watermark detection) strength. Moreover, the picture and watermark waveforms are similarly distorted when the image is attacked by, for example, compression. Therefore, it is preferred to select α to be zero or a small negative value.

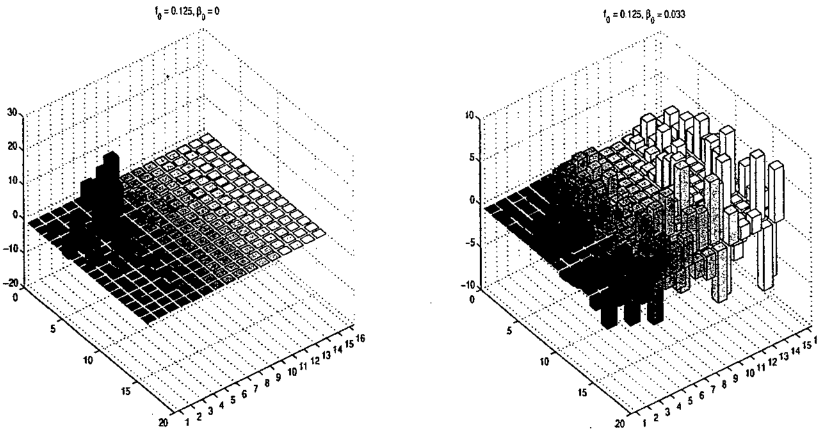
V. Chirp Parameters Optimization for JPEG Compressed Images

A common signal processing operation on most images is compression based on JPEG. The question to be answered here is the extent to which BER is affected by varying levels of compression, and more importantly, the choice of $\{\beta_0, f_0\}$ to make the watermark robust to compression. Rewrite (1) for the underlying chirp signal case as

$$G(m,n,x,y) \;\; = I(m,n,x,y) + Q\{k\Re[s(m,n)W(x,y,\beta_0,f_0)]\}. \tag{20}$$

Baseline JPEG consists of 4 sequential steps: (a) block DCT; (b) quantization; (c) zigzag scan; and (d) entropy coding. The goal here is to spectrally shape the chirp to make it most robust to JPEG for a given quality factor Q.

Fig. 4 shows how chirp energy distribution can be changed to counter JPEG quantization matrix. When $\beta_0 = 0$, the watermark is sinusoidal, and the energy is localized in the DCT domain. On the other hand, when $\beta = 0.033$ which is relatively large, the energy is distributed in the DCT domain, particularly in the high frequency regions.



(a) 16 × 16 DCT transform (β_0=0, f_0=0.125)

(b) 16 × 16 DCT transform (β_0=0.033, f_0=0.125)

Fig. 4. Block DCT of a chirp for different $\{\beta_0, f_0\}$.

These figures illustrate how chirp's spectrum is modified by different choices of $\{\beta_0, f_0\}$. The distribution of DCT coefficients should be closely matched to the quantization matrix to produce the lowest BER. Since quantization matrix tends to compress higher frequency bands more aggressively, compression affects chirps with higher frequency contents more. The advantage of using a chirp versus sinusoid is clearly demonstrated here. Sinusoid's energy is concentrated at specific frequency bands and can be easily removed by selective filtering or compression. In addition, there are virtually no degrees of freedom to spread the spectrum and optimize detection for varying JPEG Q factors.

The question of chirp survival after JPEG cannot be discerned solely by observing chirp DCT since the quantizer operates on the DCT of the image plus chirp and not the DCT coefficients individually. Denote $G(m', n')$ as the (m', n')th block of the watermark where each block is of size 8 × 8 to match the JPEG compression standard, and let $\mathcal{G}(m', n') =$ dct$[G(m', n')]$. We also define $\mathcal{I}(m', n') =$ dct$[I(m', n')]$ and $\mathcal{W}(m', n', \beta_0, f_0) =$ dct$[\tilde{W}(m', n', \beta_0, f_0)]$ in a similar way, where $\tilde{W}(m', n', \beta_0, f_0)$ is the watermark defined at the (m', n')th block with chirp parameters (β_0, f_0). Then, at the (m', n')th block, the quantized DCT coefficients are given by

$$\left\lceil \frac{\mathcal{G}(m',n')}{Q} \right\rceil = \left\lceil \frac{\mathcal{I}(m',n')}{Q} + \frac{\mathcal{W}(m',n',\beta_0,f_0)}{Q} \right\rceil \tag{21}$$

where $Q = [q_{i,j}]$ is JPEG quantization matrix, $i,j = 0,...,7$. $\lceil . \rfloor$ is rounding towards nearest integer. Note that division in (21) is an element-by-element matrix division. The decoder then performs an inverse quantization on (21) followed by inverse DCT to obtain $\hat{G}(m',n')$,

$$\hat{G}(m,n) = \mathrm{dct}^{-1}\left(Q \times \left\lceil \frac{\mathcal{I}(m',n')}{Q} + \frac{\mathcal{W}(m',n',\beta_0,f_0))}{Q} \right\rceil\right). \tag{22}$$

It is clear that the watermark will be eliminated unless the chirp has enough strength to survive compression and decompression cycle. For the watermark to survive in the presence of images,

$$\frac{\mathrm{dct}(ksW_{\beta f})}{q} \geq 0.5 - \mathrm{mod}\left(\frac{\mathrm{dct}(I)}{q}\right) \tag{23a}$$

or

$$\frac{\mathrm{dct}(ksW_{\beta f})}{q} < -0.5 - \mathrm{mod}\left(\frac{\mathrm{dct}(I)}{q}\right). \tag{23b}$$

where $\mathrm{mod}(a) = a - \lfloor a \rfloor$ denotes the fraction part of a.

If above conditions do not hold at any of the 64 locations, the quantization matrix removes the chirp entirely. In such cases, the detector is effectively presented with an unmarked block. The probability of this event is governed by the statistics of $\alpha_q = \mathrm{mod}\left(\frac{\mathrm{dct}(I)}{q}\right)$. Using (23), the probability of losing the watermark bit for a given block, or miss probability, is

$$P_M = P\left(-0.5 - \frac{dct(ksW_{\beta f})}{q} \leq \alpha_q < 0.5 - \frac{dct(ksW_{\beta f})}{q}\right). \tag{24}$$

It should be noted, however, that the detector may still decide, by chance, in favor of +1 or −1 even for an unmarked block or a block that has lost the embedded chirp due to compression. The statistics of α_q is image dependent but follows some general trends. α_q is bounded to $[0,1)$ and follows exponential-like distributions. The distribution of α_q for Lena averaged over all frequencies is shown in Fig. 5. For $k = 0$, i.e, no watermark is embedded, $P_M = 1$. Again, in spite of 100% miss probability, some watermark bits will be correctly decoded by pure chance. For large k values hence stronger watermark,

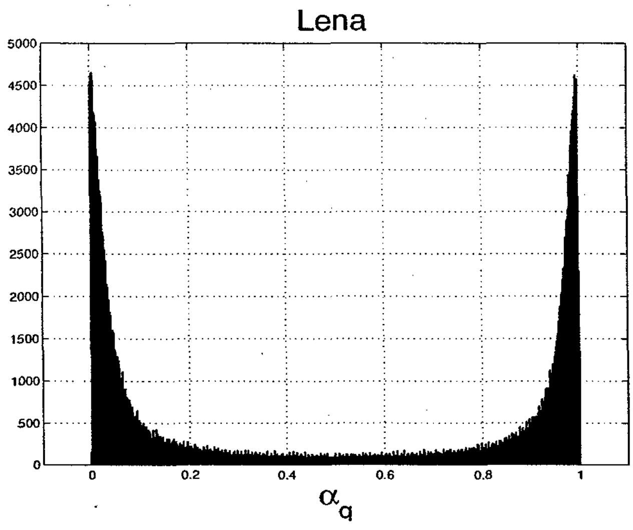


Fig. 5. Distribution of α_q for Lena averaged over all frequencies across the image.

$P_M \to 0$. In both cases the results are consistent with watermark survival as a function of power.

Successful selection of chirp parameters in a compressive environment must exclude cases where watermark is removed. To test for watermark survival in an $N \times N$ image block, we define the following metric,

$$e = \frac{1}{N^2} \sum_{m'=0}^{N-1} \sum_{n'=0}^{N-1} \left| \hat{G}(m',n') - \hat{I}(m',n') \right|, \tag{25}$$

where $\hat{I}$ is the unmarked image block that is compressed by the same Q factor as $\hat{G}$. Zero e is a sign that the watermark has been completely removed by compression. Plots of watermark survival metric, e, for fixed Q and PSNR and varying $\{\beta_0, f_0\}$ are shown in Fig. 6. Different image blocks generate different error profiles. This observation leads to the conclusion that different blocks may benefit from custom designed chirps with specific $\{\beta_0, f_0\}$.

The higher e values indicate stronger presence of the watermark following JPEG encoding and decoding. Those combinations of $\{\beta_0, f_0\}$ that result in zero e at the given Q factor must be avoided.

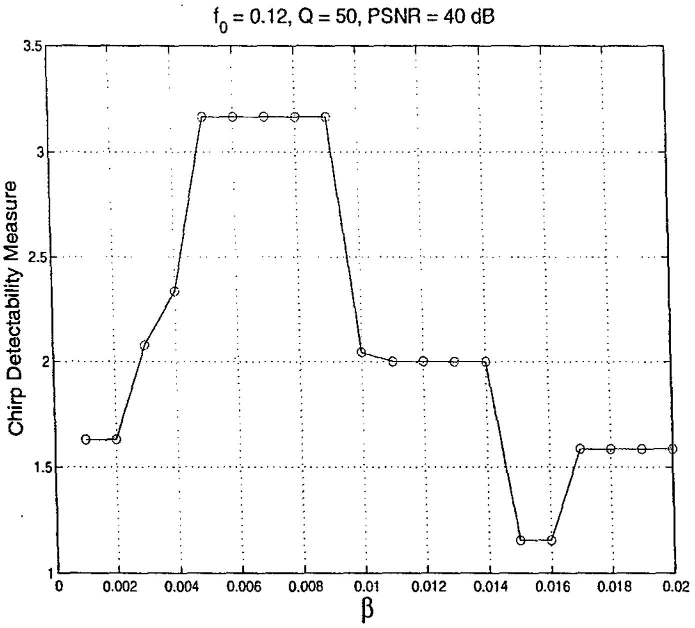


Fig. 6. Watermark survival metric vs. chirp rate, β.

VI. Simulation Results

A. Chirp Parameter Selection

We first consider the application of a single-component 2-D chirp watermark. To understand the effect of different chirp parameters to the watermark detection performance, the BER performance is shown in Fig. 7 versus the chirp rate β_0, where the initial frequency is fixed to $f_0 = 0.2333$. It is clear that, when JPEG compression is not applied, a high value of β_0 tends to provide low BER performance, because the picture does not have much power at the high-frequency band. With moderate JPEG compression, however, there exists an optimum range of β_0, because the high-frequency band will be suppressed in the process of image compression.

The next example shows the sensitivity of $|C_I(m, n, \beta_0, f_0)|$, i.e., the magnitude of the matched filter output of the original image, to the values of (β_0, f_0). Two different sets of chirp parameters, that is, $(\beta_0, f_0) = (0.008, 0.08)$ and $(\beta_0, f_0) = (0.011, 0.11)$, are considered and compared. Fig. 8 shows their histogram over the 1024 blocks, whereas the corresponding cumulative distribution plots are shown in Fig. 9. The energy of the matched

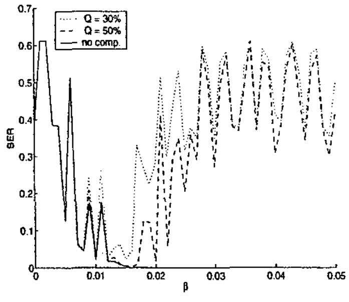
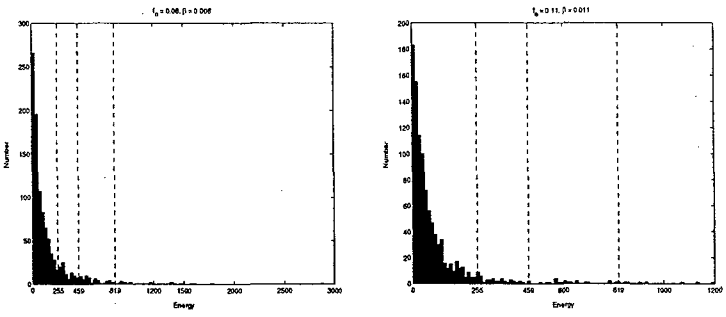


Fig. 7. BER versus chirp rate (PSNR = 35dB).

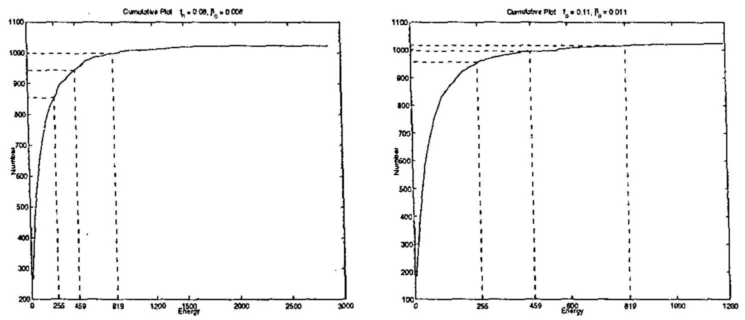
filter output of the watermark, corresponding to PSNR = 35, 40, and 45dB, are depicted in the figures. For the watermark chirp (0.008, 0.08), there are 25, 81, and 175 image blocks with a projection magnitude value more than the energy of the watermark at PSNR = 35, 40, and 45dB, respectively, whereas using the other watermark chirp, there are 8, 30, and 76 blocks having projection magnitude value more than the watermark's energy at the corresponding PSNR. Clearly, at the specific value of PSNR, $(\beta_0, f_0) = (0.011, 0.11)$ is preferable in this case, since the chance is lower for the image to influence the detection of the sign of s.



(a) $\beta_0 = 0.008, f_0 = 0.08$.

(b) $\beta_0 = 0.011, f_0 = 0.11$.

Fig. 8. Histogram of the matched filter output of the original image



(a) $\beta_0 = 0.008, f_0 = 0.08$.

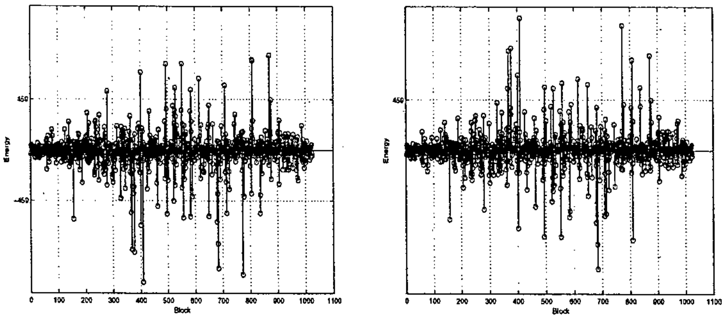
(b) $\beta_0 = 0.011, f_0 = 0.11$.

Fig. 9. Cumulative histogram of the matched filter output of the original image

B. Adaptive Watermark Power Allocation

It is evident from the previous example that, the required watermark level to assure correction detection is different for different blocks. To find the minimum watermark energy for each block, we plot in Fig. 10(a) the matched filter output of the original image in a sequential order. The dashed lines show the levels corresponding to the watermark at PSNR=40dB. At those blocks where the magnitude of the image contribution exceeds the watermark output, there is a possibility that the watermark information bit is wrongly decided. However, whether it occurs or not depends on the sign of the embedded information.

To incorporate the watermark information, therefore, we plot in Fig. 10(b) the result of $-s(m,n)C_I(m,n,\beta_0,f_0)$. A watermark decision error will occur in each of the blocks where this value is higher than the watermark output magnitude. In other words, we can choose the watermark energy at each block such that it merely exceeds $-s(m,n)C_I(m,n,\beta_0,f_0)$. As such, the watermark energy is minimized whereas low error-ràte-watermark embedding is assured. The PSNR required to achieve BER = 0 is 51dB.



(a) $C_I(m, n, \beta_0, f_0)$ (b) $-s(m,n)C_I(m, n, \beta_0, f_0)$

Fig. 10. Matched filter output of the original image. Dashed line corresponds to watermark output at (PSNR = 40 dB)

C. Watermark Detection Capability

Detection capability is another issue which should be considered in the watermarking. We refer to detection capability as the ability to answer to this question, Does a particular image have watermark hidden in it? In our approach of embedding which is a block-wise embedding, we claim that because of the low power of the watermark we cannot tell for each block whether or not we added watermark for sure. But having the whole image we can certainly tell if a particular image has embedded watermark. Figs. 11, and 12 show the sum of chirp transform spectra of all blocks of the image for both cases of with and without watermark. In the figure with embedded watermark, PSNR = 40 dB, the watermark chirp parameters are $\beta = 0.011$, and $f_0 = 0.11$. It is clear from this figures that in the spectrum of watermarked image we can mark the position of chirp watermark which was designed away from the content of image.

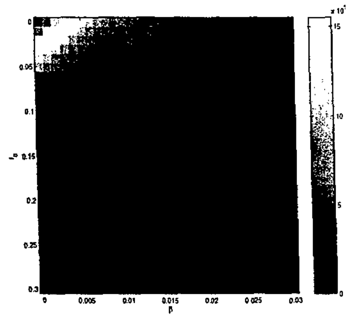


Fig. 11. Sum of chirp transform spectra of all blocks of Lena picture.

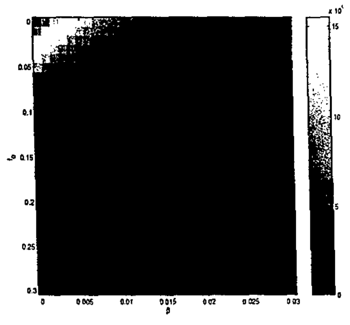
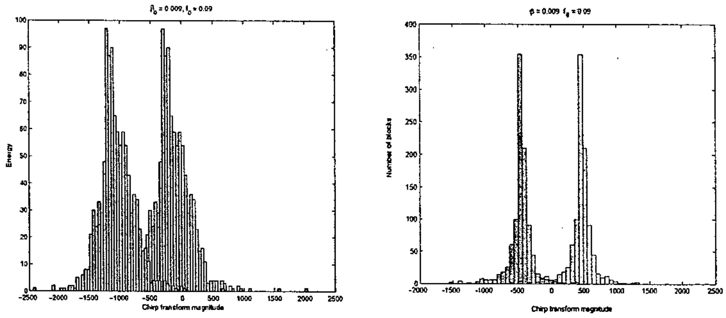


Fig. 12. Sum of chirp transform spectra of all blocks of watermarked Lena picture.

D. Effect of DC Components

The previous simulation results have assumed that the DC components of the watermarked image and the watermark waveform are removed, as we discussed in Section III. The next example shows that, if such DC component removal is not properly performed, there may be a significant bias which, in turn, will affect the watermark detection.

Figure 13(a) shows the histograms of $C(m, n, \beta_0, f_0)$, the matched filter output of the watermarked image. The chirp parameters are $(\beta_0, f_0) = (0.009, 0.09)$ and the PSNR is 40dB. The histograms corresponding to watermark information $+1$ and -1 are overlapped in this plot. It is obvious that the high bias observed in this figure will make the watermark decision very difficult. When the DC components are removed, however, as shown in Fig. 13(b), no bias is observed.



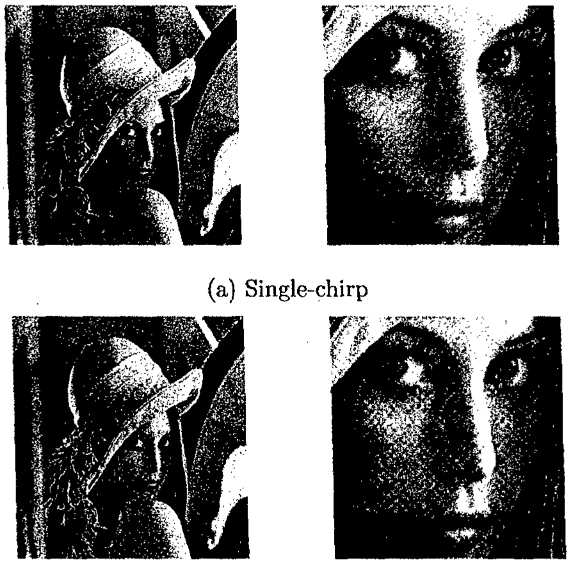
(a) DC components not removed.

(b) DC components removed.

Fig. 13. Histograms of $C(m, n, \beta_0, f_0)$ with $s(m, n) = +1$ and -1 embedded ($\beta_0 = 0.009, f_0 = 0.09$)

E. Multi-Chirp Watermark

In Figure 14, the watermarked Lena picture is shown where the JPEG quality factor $Q = 30\%$, PSNR $= 30$dB, and $\beta = 0.015$, for both cases of single- and multi-component chirp watermarks. $\alpha = 0$ is used for multi-chirp combining. The lower perceptibility of the four-component 2-D chirp scheme is evident, comparing these two pictures, though it may not be that clear in print.

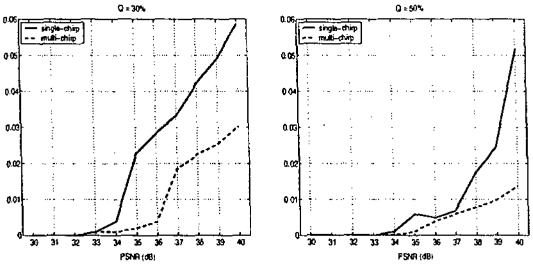


(a) Single-chirp

(b) Multi-chirp

Fig. 14. Comparison of watermarked images (PSNR = 35dB, $\beta = 0.015$, Q= 30%). Left: full size picture; Right: enlarged picture of the face part.

In Figure 15, we show the BER versus the PSNR where the respective optimum value of β_0 is used in each scenario. Again, $f_0 = 0.2333$ is used. The performance curves for no compression case are not shown since the respective BER = 0 throughout the range of PSNR. Comparing the two figures, it is clear that the multi-chirp watermarking outperforms its single-component counterpart.

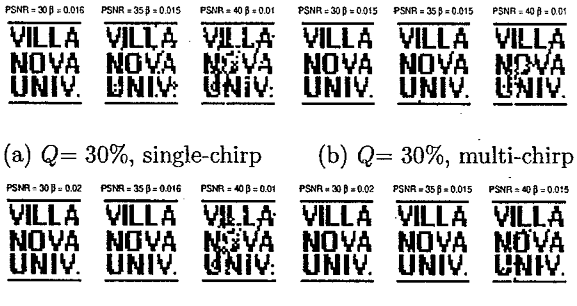


(a) Q = 30%

(b) Q = 50 %

Fig. 15. BER performance versus PSNR.

Figure 16 depicts the blindly decoded seal results. The results clearly show the robustness of the watermarking and detection, even when the PSNR is high and the picture is highly attached by image compression. The advantage of using four-component chirp watermarking is also evident.



(a) Q= 30%, single-chirp (b) Q= 30%, multi-chirp

(c) Q= 50%, single-chirp (d) Q= 50%, multi-chirp

Fig. 16. Seal detection results.

F. JPEG Compressed Imagery

Because JPEG operates on 8 × 8 blocks, a single watermark waveform is spectrally divided among multiple image blocks when watermarked image blocks are larger than the JPEG block. In our example, a 16 × 16 block is divided into 4 image blocks as shown in Fig. 17.

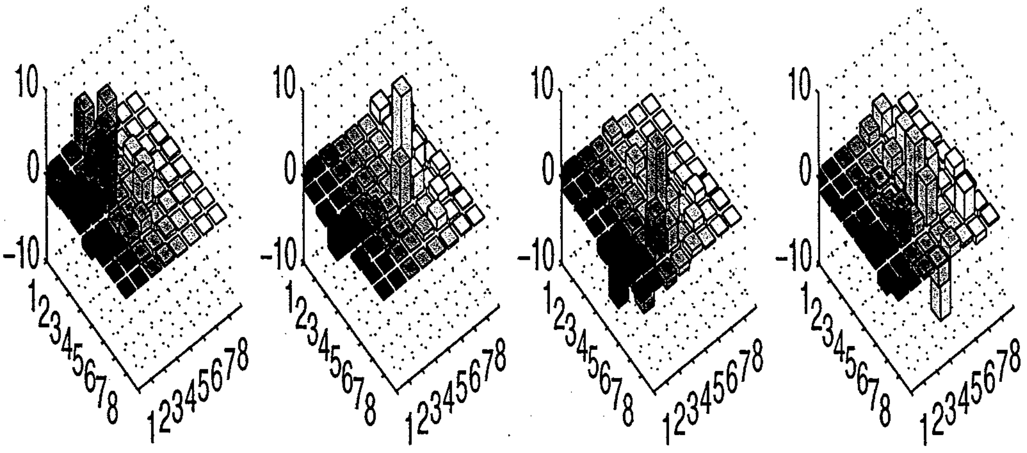
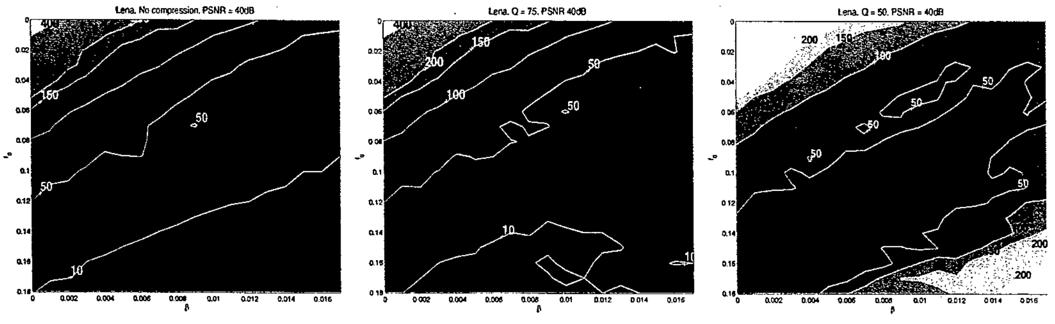


Fig. 17. Nonoverlapping 8×8 DCT blocks of a 16×16 chirp.

It is now possible to observe the BER versus (β_0, f_0) for various compression levels for a given PSNR. The objective is to tune the chirp parameters for optimum BER performance across a wide range of Q factors. The 512×512 image is divided into 16×16 blocks, each carrying one watermark bit. BER values are then found by simulations for Q factors $=$ 50 and 75 as well as the case with no compression. Fig. 18 shows contours and patches of constant BERs for Lena, where the values denote the numbers of incorrectly decided bits out of the 1024 total watermarked bits. For comparison, we also show the same results for the Elaine picture in Fig. 19.

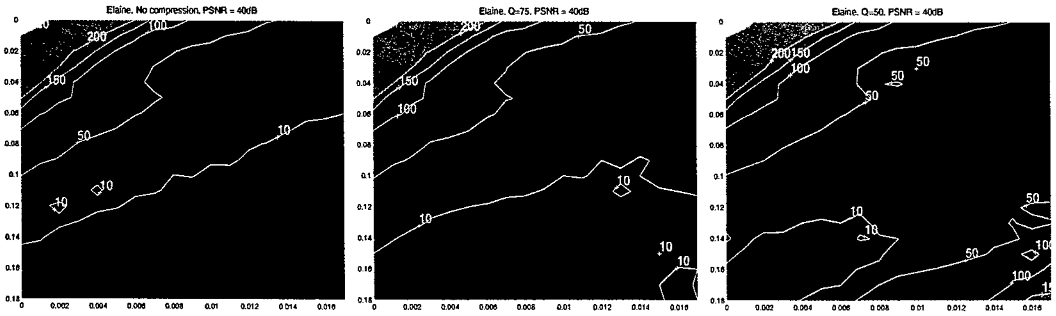
It is observed from these figures that, for higher compression factors, the low BERs appear in lower (β_0, f_0) regions. This behavior is consistent with JPEG compression as more high frequency components are suppressed, along with the watermark. Similar simulations have been carried out for Elaine. Trends are similar but variations of BER vs. (β_0, f_0) are clearly image-dependent. It is often desired to select (β_0, f_0) pairs that survive compression across a range of Q factors. Fig. 18 can be used to identify overlapping portions of (β, f) to achieve certain BERs. It is interesting to note that for Elaine, it is possible to select chirps that meet BER < 0.01 across all $Q \geq 50$. The same cannot be said for Lena.

An inspection of BER contours reveals another important property of the chirp. As shown in Fig. 19(a), to achieve BERs below 0.01 using sinusoids allows the use of only a limited number of frequencies below $f_0 = 0.143$. Using a chirp instead greatly expands the possible (β_0, f_0) pairs that achieve the required BER. The expanded choice is important in tuning the watermark in order to counter compression effects. For example, in Lena image there are no sinusoids that achieve BER < 0.01 for $Q = 75$, whereas there are plenty of chirps that would achieve this specified BER.



(a)No compression. (b) With compression ($Q = 75$). (c) With compression ($Q = 50$).

Fig. 18. BER contour for Lena.



(a)No compression. (b) With compression ($Q = 75$). (c) With compression ($Q = 50$).

Fig. 19. BER contour for Elaine.

G. A Comparison with Spread-Spectrum Watermarking

Here we consider the same system for watermark embedding and detection but substituting 2-D chirp watermark with a 2-D M-sequence watermark.

In figures 20, 21, and 22 a comparison of bit error vs. PSNR is shown for both cases of chirp and M-sequence watermark. The chirp parameters that we chose are $\beta = 0.011$, and $f_0 = 0.11$.

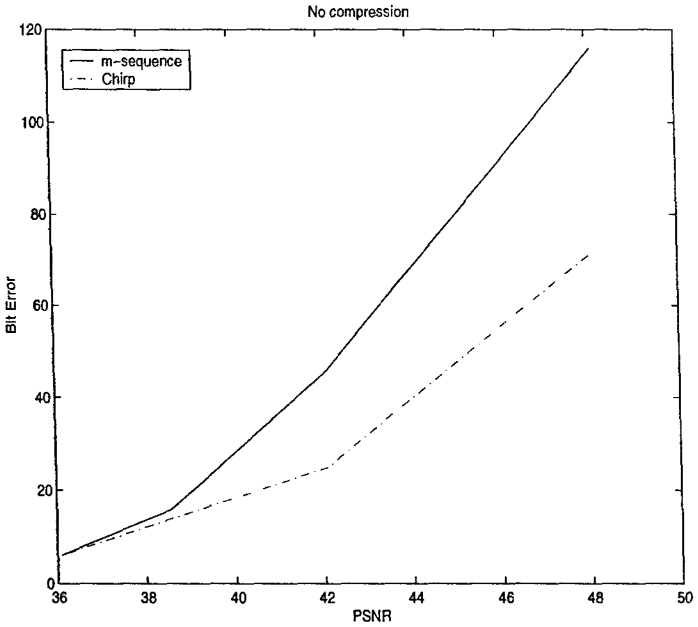


Fig. 20. Bit error vs. PSNR.

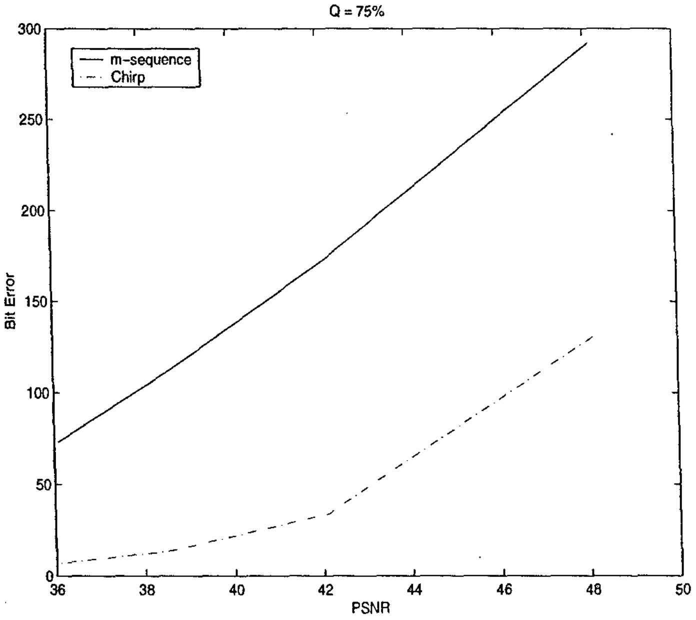


Fig. 21. Bit error vs. PSNR.

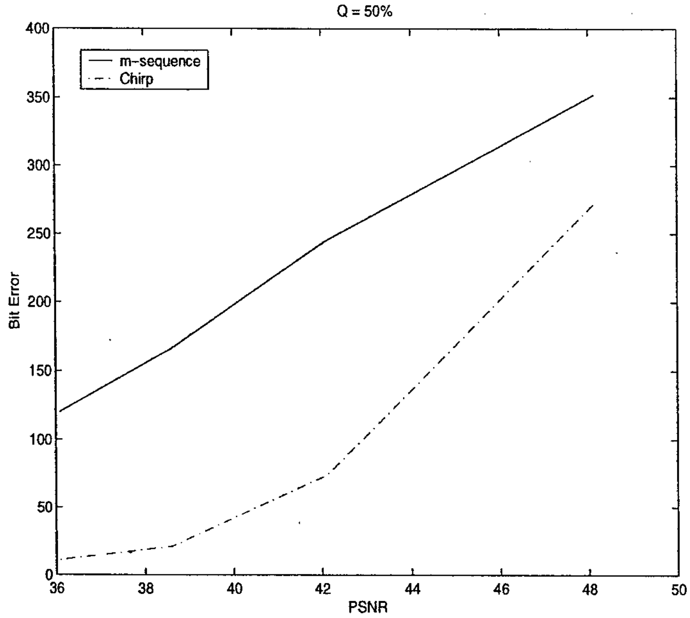


Fig. 22. Bit error vs. PSNR.

These figures obviously show the better performance of chirp watermark vs. M-sequence watermark as far as bit error is concerned, specially for the cases with JPEG compression.

The following table shows a comparison between the two methods when JPEG compression is applied. The PSNR = 39dB. This table shows the bit error for different compression rates. As can be seen from the table, for M-sequence which has a spread spectrum the degradation from JPEG compression is so high. The chirp with the same chirp parameters as before survives the JPEG compression perfectly.

Bit Error	no compression	$Q = 75\%$	$Q = 50\%$
chirp	14	14	21
M-sequence	16	112	166

Figures 23, and 24 show a comparison for the perceptibility of the watermarked image. It is clear that the M-sequence watermark has lower perceptibility because of its pseudo-noise nature.

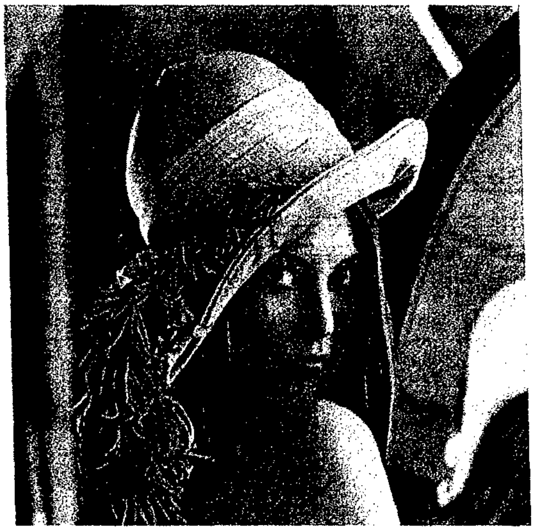


Fig. 23. Watermarked image using 2-D chirp.

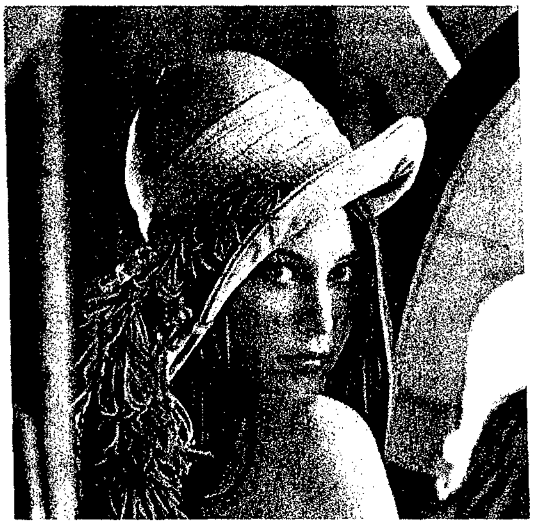


Fig. 24. Watermarked image using M-sequence.

H. Attacks on Watermark

The idea of an attack is any processing of watermarked image that might damage the watermark. Attacks could be accidental, like JPEG compression of a watermarked image, or hostile. like an effort by a multimedia pirate to remove or destroy watermark. Some examples of attacks include compression, linear filtering, and geometric transformations like rotation, shearing, scaling, and cropping. In the following, we provide simulation results for scaling and median filtering.

Scaling

We scale the watermarked image to half of its size and to detect the watermark we rescale image to its original size which has apparently lost a lot of fine details. The half sized and rescaled watermarked image are shown in Fig. 25 for PSNR = 40 dB, $\beta = 0.011$, and $f_0 = 0.11$. The number of bit errors in detected seal in this case is 42. For the case of scaling to 99% and 80% and rescaling back to the original size, the number of errors are 34 and 40, respectively. For the same parameters if we do not apply any scaling and rescaling the number of errors is 14.

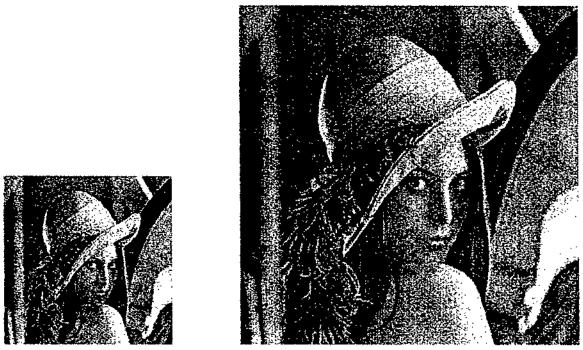


Fig. 25. Scaled (left) and rescaled (right) watermarked image

Median Filter

Here we consider median filtering as an attack to watermarked image. The median filtered version of watermarked image with the median filter of size 3 × 3 is shown in Figure 26. The number of errors is 34. All other parameters are same as parameters in scaling attack part. For median filters of sizes 4 × 4, 5 × 5, and 6 × 6, numbers of errors are 189, 311, and 459, respectively. As you can judge by the number of errors except for the median filter of size 3 × 3, other median filters affect the watermark in an adverse manner that the seal would not be detectable.

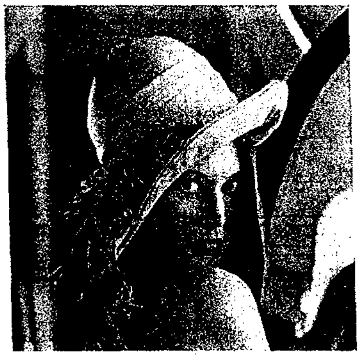


Fig. 26. Median filtered version of watermarked image

VII. Conclusions

In this chapter, a novel method has been proposed for digital watermark embedding and detection. The watermarking is based on 2-D FM signals which is robust to various attacks. 2-D chirp signals are used as examples for extensive investigation. By properly choosing the chirp parameters, a watermark chirp signals can be designed to maximally distinguish itself from the original image, resulting in optimum watermark coding. An adaptive chirp power technique improved the performance and also imperceptibility of the watermarked image. The watermarking scheme can embed multiple chirp components for more secure and flexible data coding, and the results have shown superiority over the case when a single chirp is used.

References

[1] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multemedia," *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[2] J. R. Hernandez, M. Amando and F. Perez-Gonzalez, "DCT-domain watermarking techniques for still images: detector performance analysis and a new structure," *IEEE Trans. Image Processing*, vol. 9, pp. 55–68, Jan. 2000.

[3] F. Hartung and B. Girod, "Digital watermarking of uncompressed and compressed video," *Signal Process.*, vol. 66, pp. 283–301, May 1998.

[4] M. Kutter and S. Winkler, "A vision-based masking model for spread-spectrum image watermarking," *IEEE Trans. Image Processing*, vol. 11, pp. 16–25, Jan. 2002.

[5] C. Honsinger and M. Rabbani, "Data embedding using phase dispersion," *Eastman Kodak*, 2000.

[6] S. Stanković, I. Djurović, and I. Pitas, "Watermarking in the space/spatial-frequency domain using two-deimentional radon-wigner distribution," *IEEE Trans. Image Processing*, vol. 10, no. 4, pp. 650–658, Apr. 2001.

[7] F. Pérez-González, F. Balado, and J. R. H. Martin, "Performance analysis of existing and new methods for data hiding with known-host information in additive channels," *IEEE Trans. Image Processing*, vol. 51, pp. 960–980, April 2003.

[8] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," *Proc. IEEE*, vol. 87, no. 7, pp. 1127–1141, July 1999.

[9] M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Inform. Theory*, vol. IT-29, no. 3, pp. 439–441, May 1983.

[10] L. Cohen, *Time-Frequency Analysis.* Upper Saddle River, NJ: Prentice Hall, 1995.

[11] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking digital image and video data," *IEEE Signal Processing Magazine*, Sep. 2000.

[12] C. I. Podilchuk, and E. J. Delp, "Digital watermarking: algorithms and applications," *IEEE Signal Processing Magazine*, July 2001.

[13] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform: with Applications in Optics and Signal Processing.* New York, NY: John Wiley, 2001.

[14] V. Katkovnik, "Discrete-time local polynomial approximation of the instantaneous frequency," *IEEE Transactions on Signal Processing*, vol. 46, no. 10 , pp. 2626–2637, Oct. 1998.

[15] X. -G. Xia, "Discrete chirp-Fourirer transform and its aplication to chirp rate estimation," *IEEE Trans. Signal Processing*, vol. 48, pp. 3122–3133, Nov. 2000.

List of Publications

1. Yimin Zhang, Moeness G. Amin, Behzad Mohammadi Dogahe, and Gordon J. Frazer, "Time-frequency analysis for maneuvering target detection in over-the-horizon radars," International Symposium on Signal Processing and Its Applications, Paris, France, July 2003.

2. Yimin Zhang, Behzad Mohammadi Dogahe, Bijan Mobasseri, and Moeness G. Amin, "Digital watermarking with two-dimensional FM waveforms", SPIE Annual Conference, Denver, CO, Aug. 2004.

3. Yimin Zhang, Behzad Mohammadi Dogahe, Bijan Mobasseri, and Moeness G. Amin, "Digital watermarking with two-dimensional FM waveforms", IEEE Transactions on Image Processing (to be submitted).

DIGITAL WATERMARKS

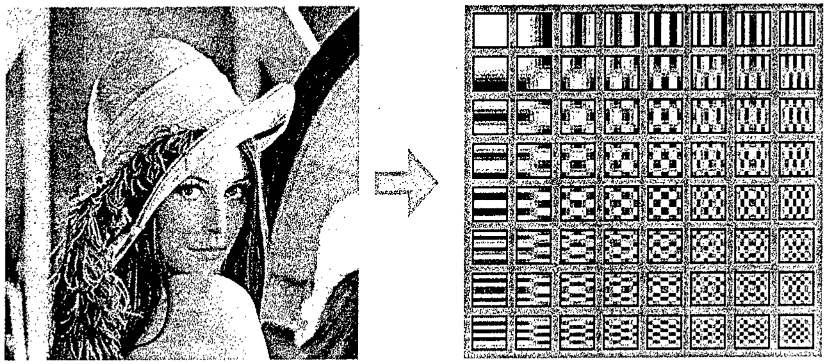
- Authentication and Metadata Embedding

....10001010101010....

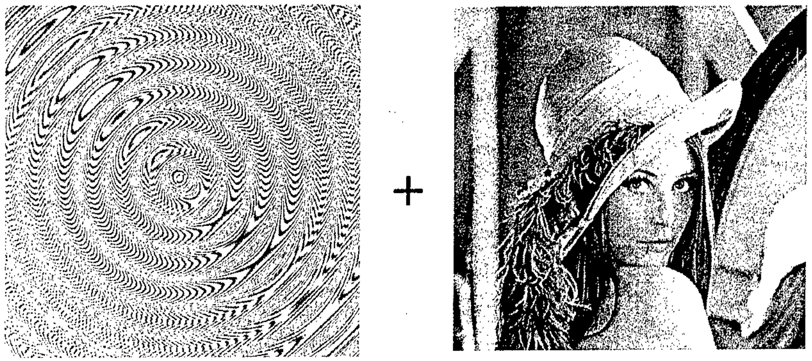
Military Unit or Location

WATERMARK TYPES

- Bitstream ....1001010100...
- Spectral



- Spatial



PERFORMANCE METRIC

- Security
- Robustness: Compression, Geometric Attacks: Rotation
- Perceptual Transparency

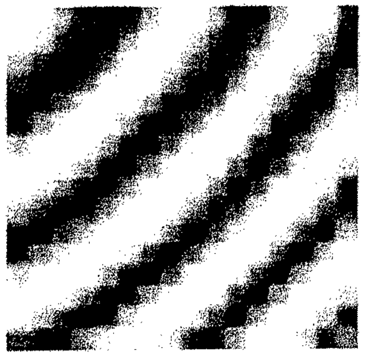
Watermarked (40dB)

Original

THE CHIRP

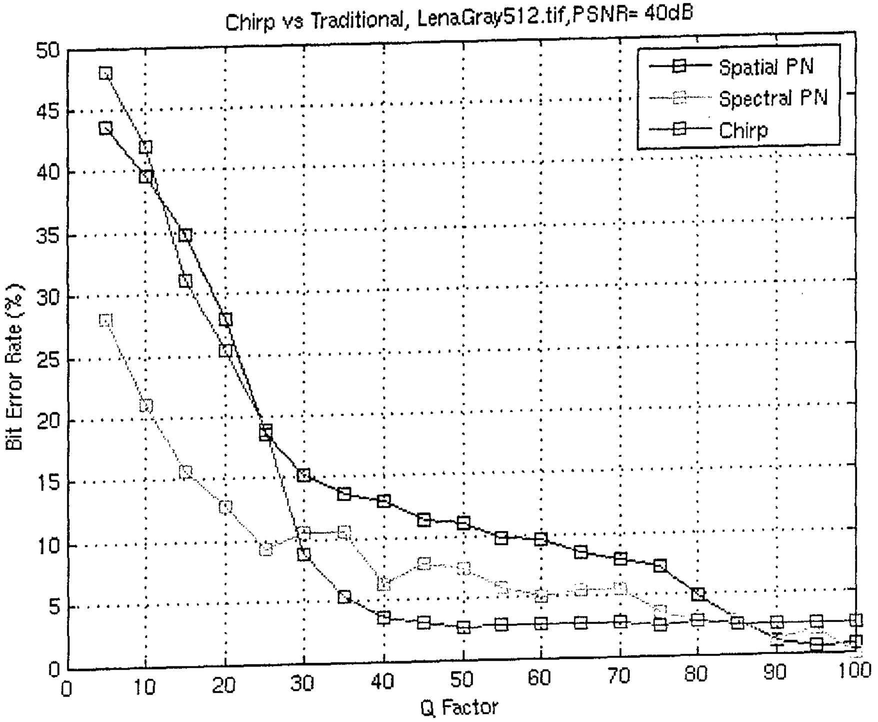
$$W\left(x,y\right)=e^{j\pi\beta\left(x^{2}+y^{2}\right)+j2\pi f\left(x+y\right)}$$

- Spatial Embedding
- Avoid Compression
- Tunable Variable
  - Frequency Rate= β
  - Natural Frequency= f



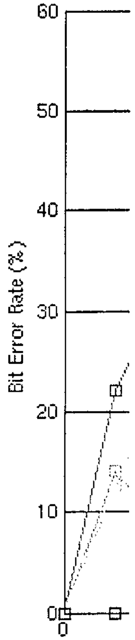
16x16 Block Chirp

COMPRESSION BENEFITS

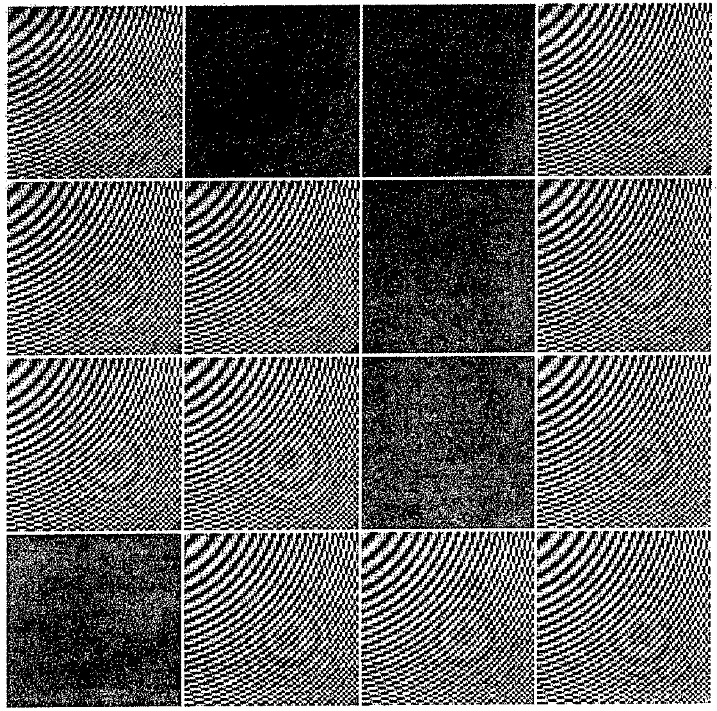


- Traditional Algorithms
  - Spectral PN Sequence
  - Spatial PN Sequence

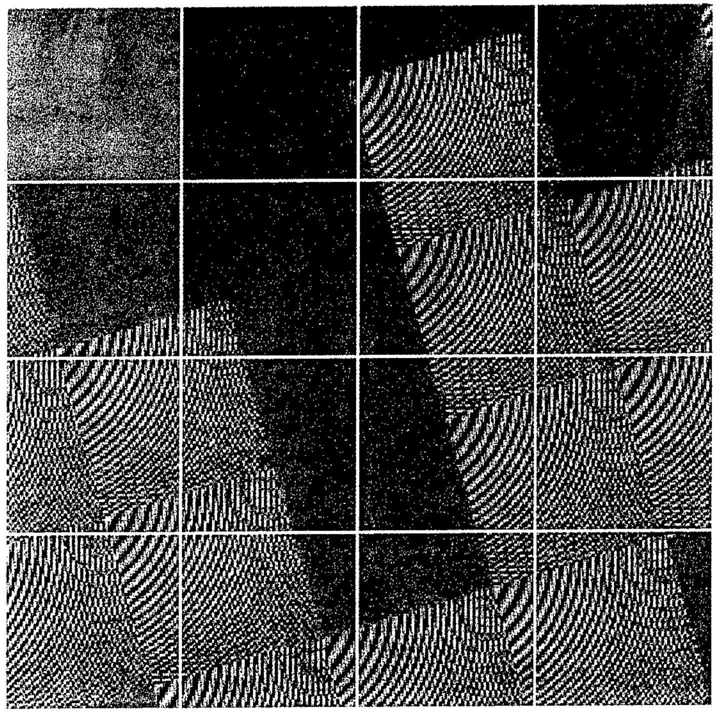
CHIRP DISADVANTAGE



No Rotation



Rotated 16⁰



1 degree(s)

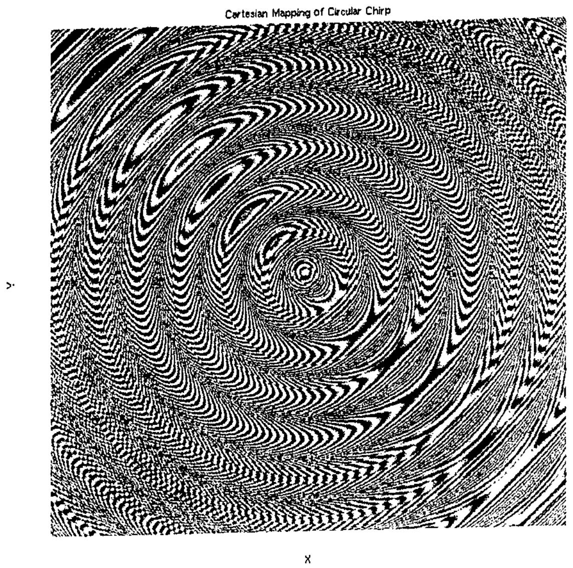
46.939

THE RING CHIRP

$$W\left(x,y\right)=e^{j\pi\beta\left(x^2+y^2\right)+j2\pi f\left(x+y\right)}$$

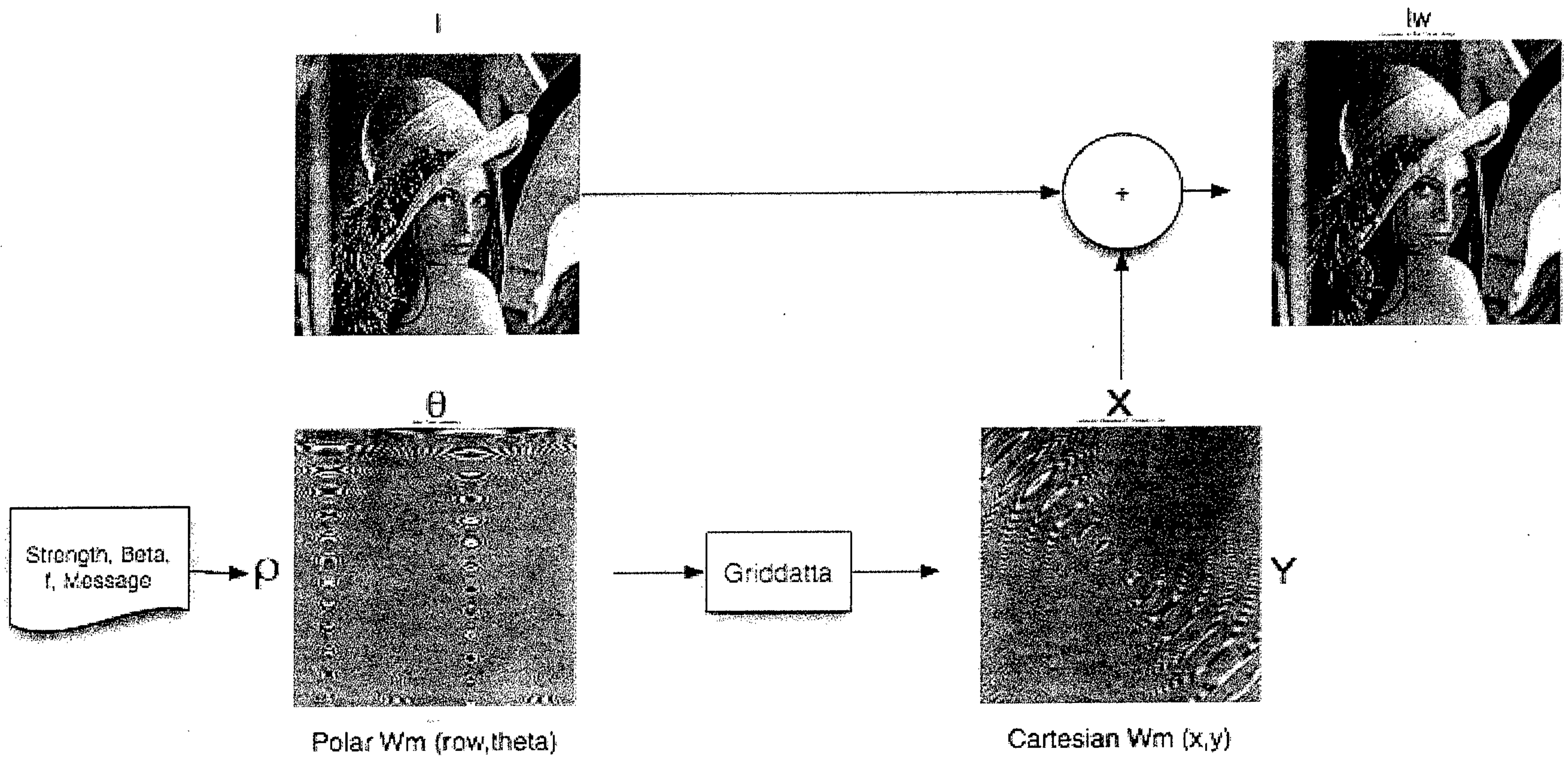
$$x=\rho\cos(\theta)$$
$$y=\rho\sin(\theta)$$

$$W\left(\rho,\theta\right)=e^{j\pi\beta\left(2\rho^2\right)+j2\pi f\rho\left(\cos(\theta)+\sin(\theta)\right)}$$



- Circular Implementation
- Tunable Spreading Function
- Spatially Embedded

EMBEDDING



I

Iw

θ

X

Strength, Beta, f, Message

ρ

Griddatta

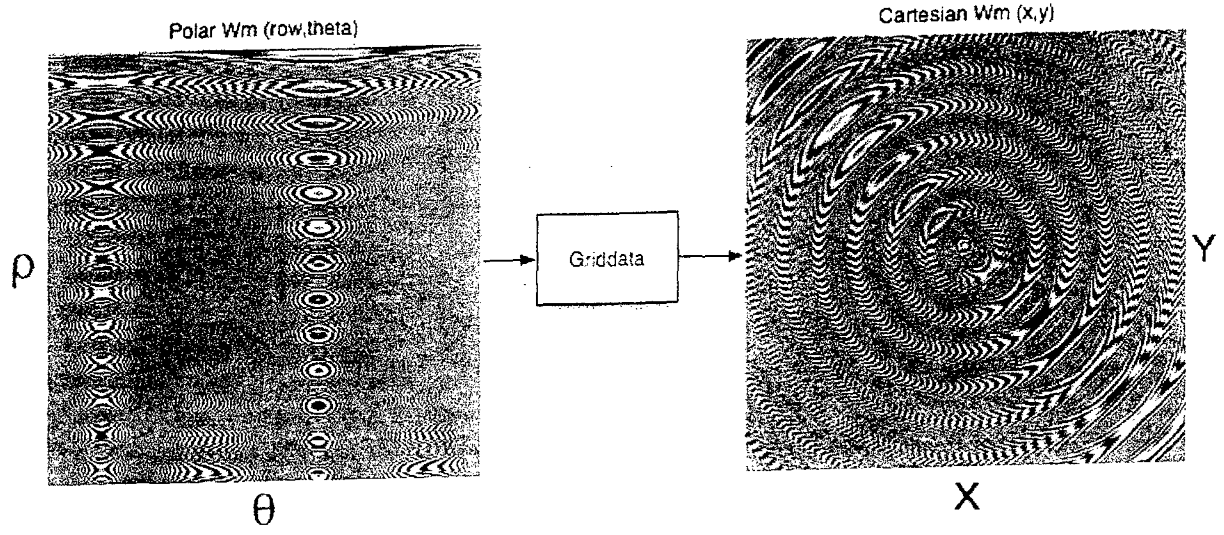
Y

+

Polar Wm (row,theta)

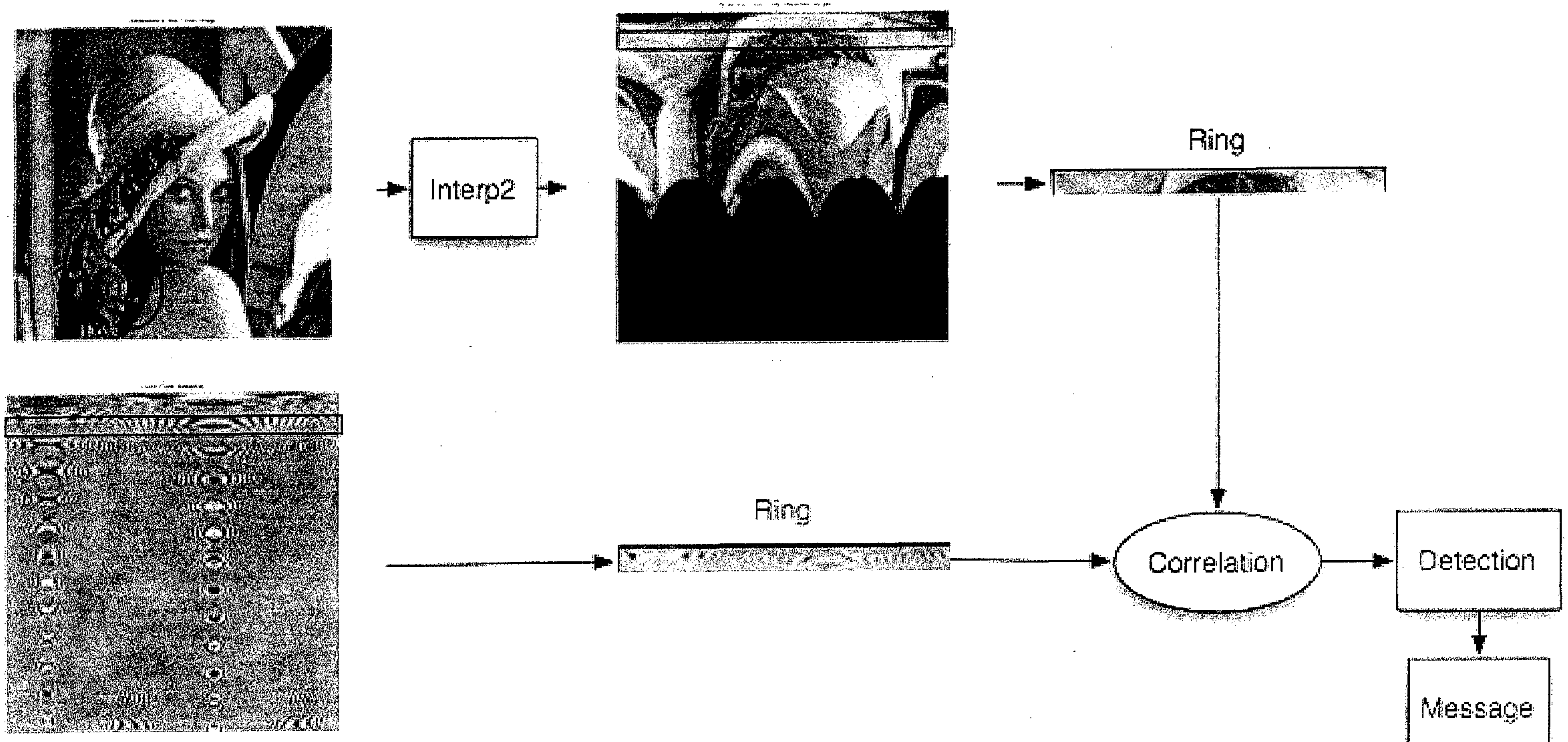
Cartesian Wm (x,y)

POLAR TO CARTESIAN



- Image Pixels are in Cartesian Coordinates
- Interpolation Necessary
- Authentication Marker

DETECTION



Cartesian Iw (x,y)

Interp2

Polar Iw (row, theta)

Ring

Recreated Polar Wm (row,theta)

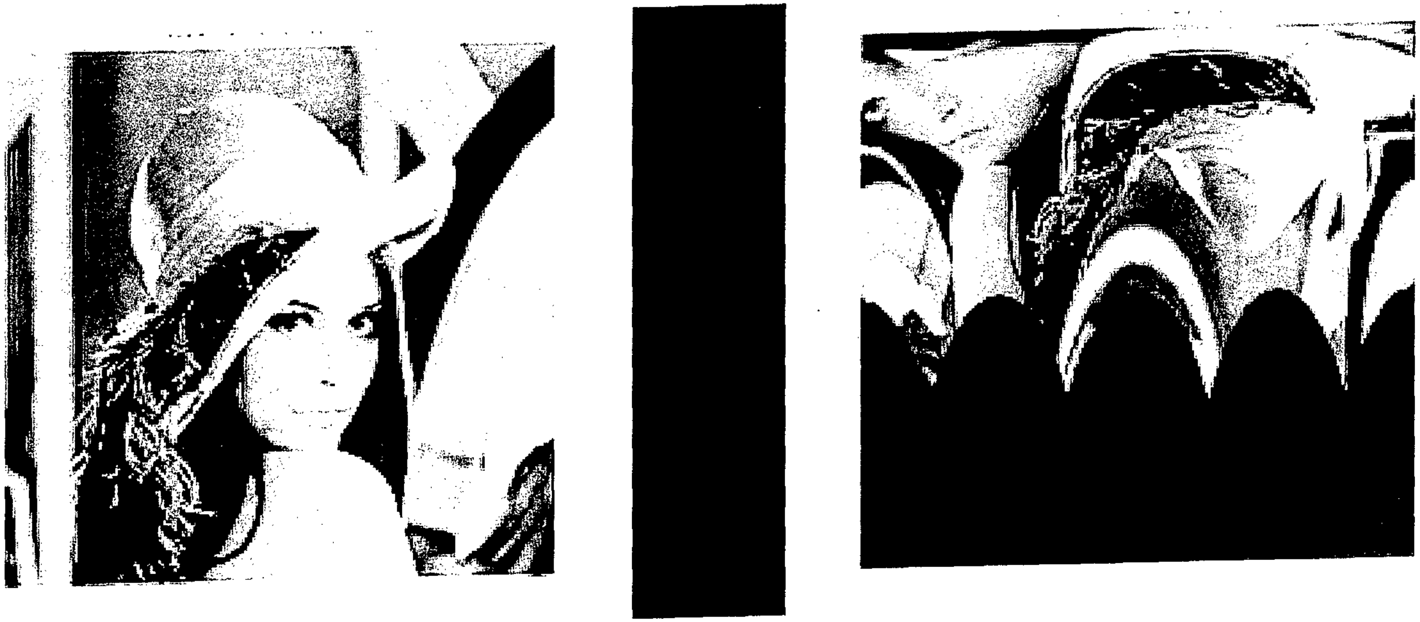
Ring

Correlation

Detection

Message

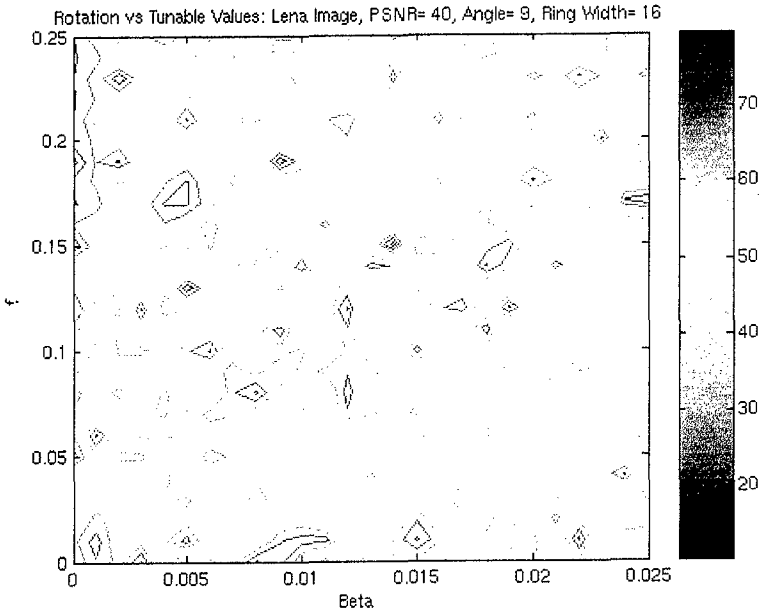
ROTATION TO TRANSLATION



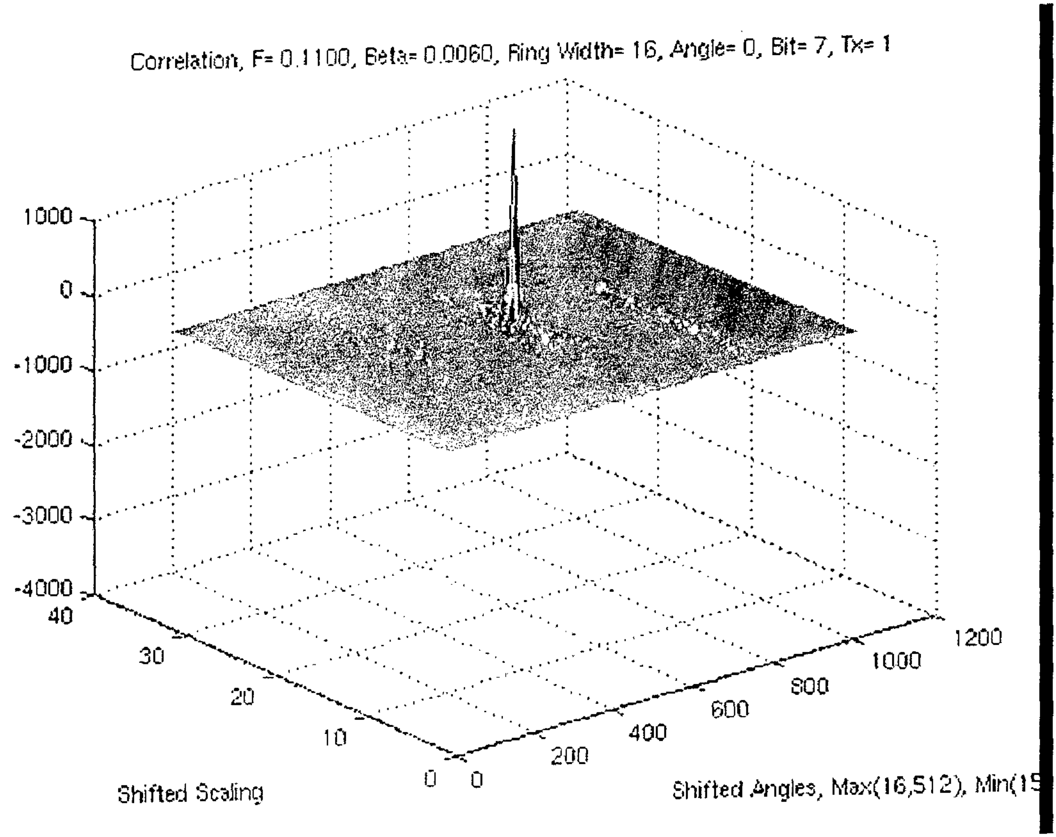
- CCW Rotation is Converted to a Left Shift

DETECTION OPTIMIZATION

- Cross Correlation
- Corner-less Detection
- Tunable Variable Security



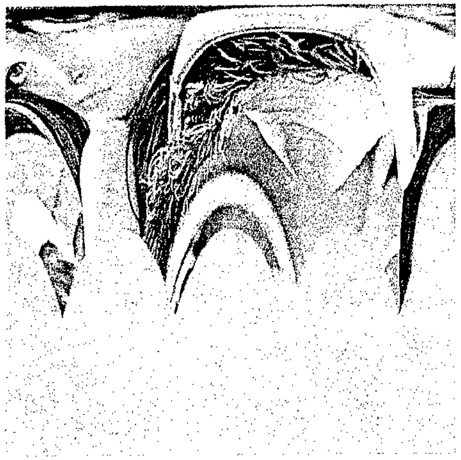
CROSS CORRELATION



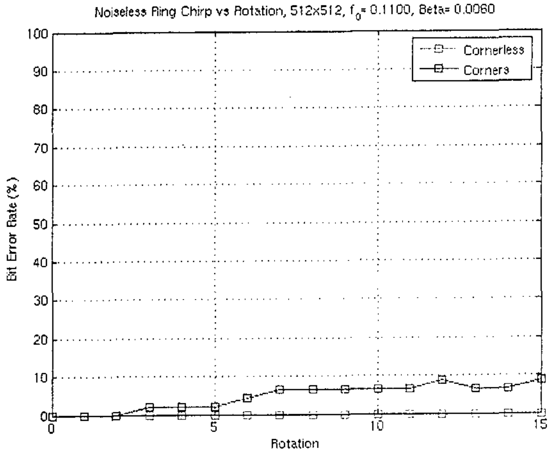
- Shifting Correlation
- Rotation Estimation

CORNER-LESS DETECTION

Cartesian



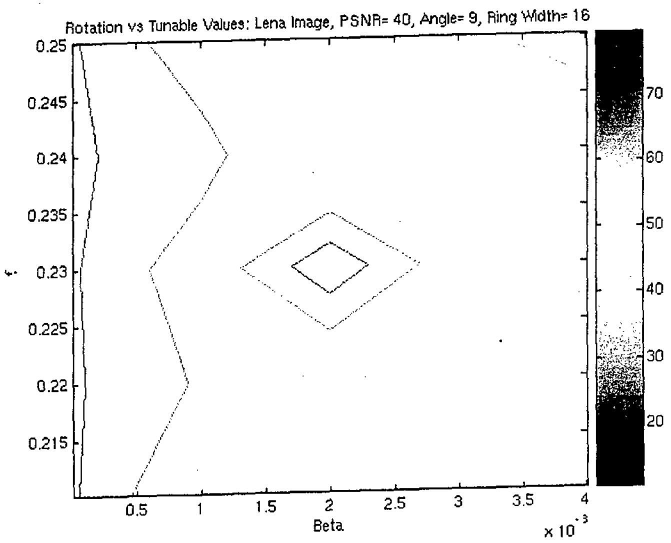
Polar

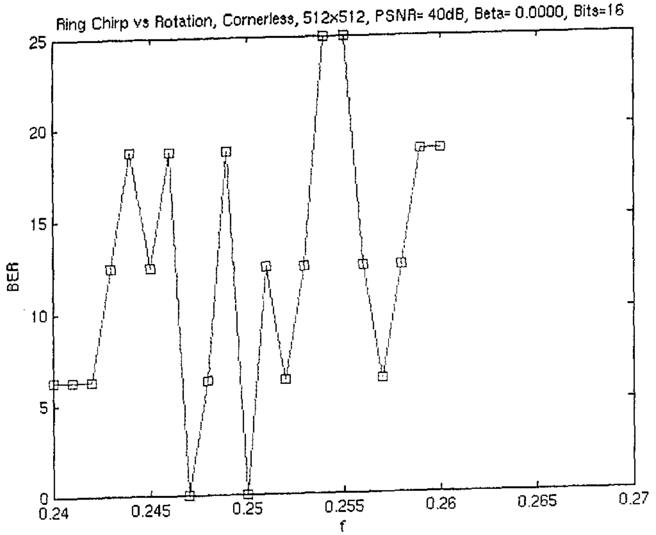


Performance

- Highlighted Region is Avoided
- Decreased Performance

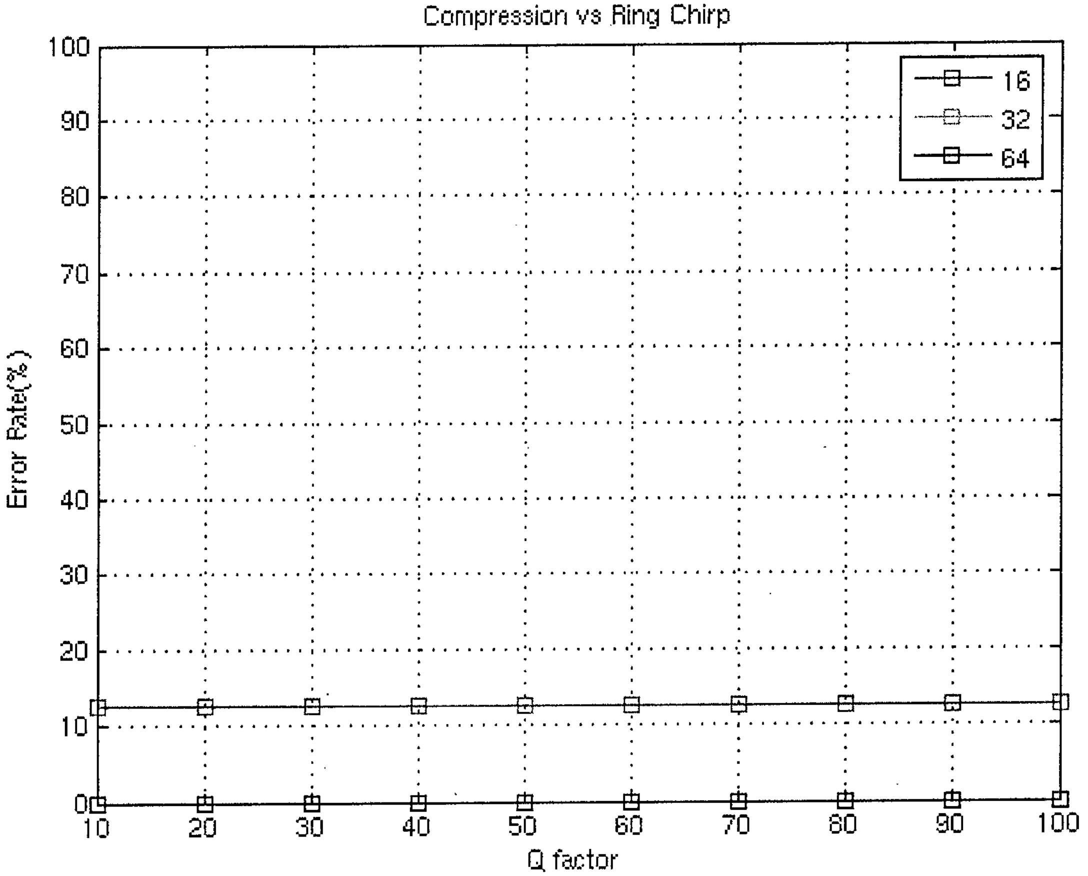
SECURITY





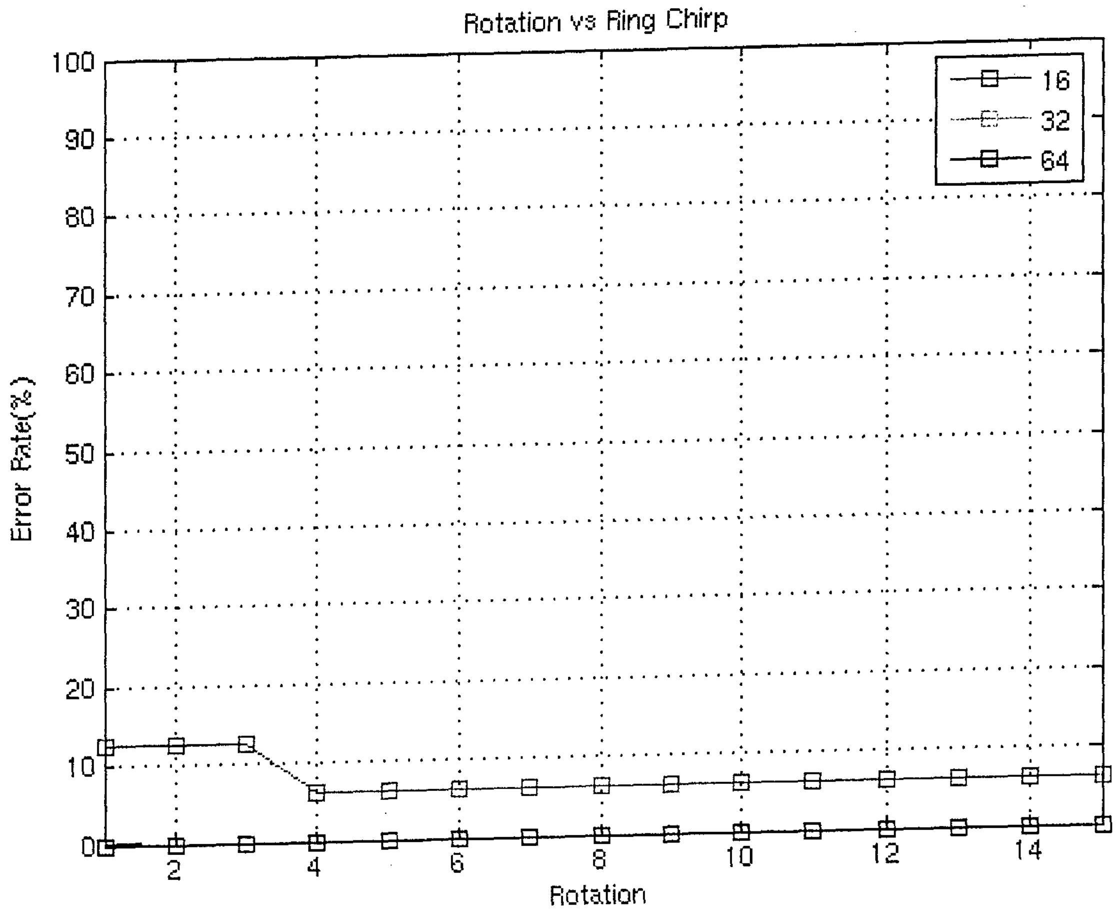
- Ring Width
- Authentication Markers
- Beta and F
- Discrete Range

COMPRESSION RESULTS



- Optimal Beta and F

ROTATION RESULTS



- Optimal Beta and F

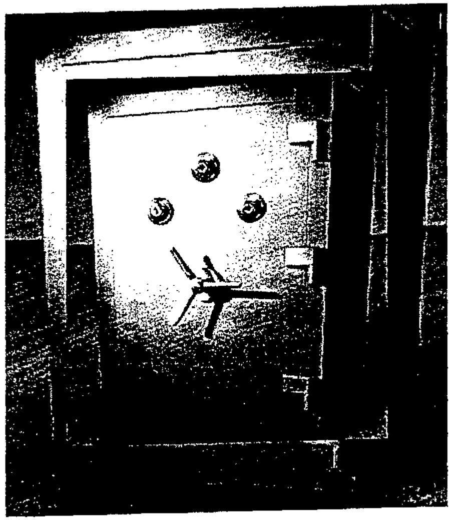
EXAMPLES

Original

Watermarked

CONCLUSIONS

- Robust to Rotation
- Resilient to Compression
- Secure



FUTURE WORK

- Rotation Estimation
- Sectoring
- Center Point Error
- Tunable Variable Precision

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 31-10-2005	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* 01-06-2004->31-08-2005

4. TITLE AND SUBTITLE

Digital Watermarking of Autonomous Vehicles Imagery and Video Communications

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014-04-1-0630

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Bijan G. Mobasseri, PI

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Center for Advanced Communications
Villanova University
800 Lancaster Ave.
Villanova University
Villanova, PA 19085

8. PERFORMING ORGANIZATION REPORT NUMBER

5-27729

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
875N. Randolph Street
12th Floor
One Liberty Center
Code 02
Arlington, VA 22217-5660

10. SPONSOR/MONITOR'S ACRONYM(S)

ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release; Distribution is Unlimited

13. SUPPLEMENTARY NOTES
N/A

14. ABSTRACT

We have developed, implemented and tested a known-host-state methodology for designing image watermarks that are particularly robust to compression. The proposed approach outperforms traditional spread spectrum watermarking across all JPEG quality factors. The fundamental approach is based on using 2D chirps as spreading functions, followed by chirp transform to recover the watermark. The reason for enhanced performance is the ability to spectrally shape the chirp to match image content and JPEG quantization. The energy localization of chirp is exploited to embed low power watermark per image blocks while maintaining reliable detection performance. In the course of this study we discovered that a chirp defined over a square grid is very susceptible to rotations of even small amounts. To address this difficulty we defined a *ring chirp* defined over a polar coordinate system. Embedding capacity is reduced but substantial robustness to image rotation is achieved.

15. SUBJECT TERMS
Digital watermarking, time-frequency, chirp, polynomial phase

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Bijan G. Mobasseri
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU		**19b. TELEPHONE NUMBER** **610-519-4958**