# Digital watermarking of images in the fractional Fourier domain

*Adhemar Bultheel*

*Report TW 497, July 2007*




Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

# Digital watermarking of images in the fractional Fourier domain

*Adhemar Bultheel*

*Report TW 497, July 2007*

Department of Computer Science, K.U.Leuven

**Abstract**

We describe the implementation of a watermark embedding technique for images using the discrete fractional Fourier transform. The idea is that a 2D discrete fractional Fourier transform of the image is computed. In this transform domain, a normal distributed random sequence with certain characteristics is used to modify the transform coefficients in the middle range of magnitude. Then the inverse transform is applied to give the watermarked image. The detection is based on performing the same transform operation. The watermark is recognized if there is a strong correlation with the embedded watermark. That is when the correlation is above some threshold.

**Keywords :** Fractional Fourier transform, digital watermarking
**AMS(MOS) Classification :** Primary : 68U10,

# Digital watermarking of images in the fractional Fourier domain

A. Bultheel [1,2]

**Abstract**

We describe the implementation of a watermark embedding technique for images using the discrete fractional Fourier transform. The idea is that a 2D discrete fractional Fourier transform of the image is computed. In this transform domain, a normal distributed random sequence with certain characteristics is used to modify the transform coefficients in the middle range of magnitude. Then the inverse transform is applied to give the watermarked image. The detection is based on performing the same transform operation. The watermark is recognized if there is a strong correlation with the embedded watermark. That is when the correlation is above some threshold.

*Key words:* Fractional Fourier transform, digital watermarking
*2000 MSC:* 68U10 94A29

## 1 Introduction

Digital watermarking of signals and images, which can be defined as "imperceptible insertion of information into multimedia data" is a very broad domain and several textbooks appeared on the subject. It is not our intention in this note to give a field spanning introduction. We advise the reader to consult one of the available references on this subject. A wealth of information and references can be found on the site of Watermarking World [6].

[1] Department of Computer Science, K.U.Leuven, Belgium.
[2] The work of the author is partially supported by the Fund for Scientific Research (FWO), projects "RAM: Rational modelling: optimal conditioning and stable algorithms", grant #G.0423.05 and the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with the author.

Many techniques are DCT based. We intend to discuss here only one specific technique proposed by I. Djurović et al [4] which adds a watermark in the fractional Fourier transform domain. Not many papers deal with watermarking based on the fractional Fourier technique. See for example [5,10] for other methods which are more DCT-like since the basic idea is to subdivide the image in smaller (e.g., $8\times 8$ blocks) and hide some bits of the watermark in the transform of each block. We discuss here another technique which computes the fractional Fourier transform (FRFT) of the whole image, then adds a random watermark with certain statistical and deterministic properties and back-transform. The watermark can be detected when the statistical properties are recovered in the appropriate transform.

This is work based on the master thesis [14].

## 2 Preliminary results

The fractional Fourier transform (FRFT) over an angle $\alpha = \pi/2$ corresponds to the classical Fourier transform, which is denoted as $\mathcal{F}^1 = \mathcal{F}$. Setting $\alpha = a\pi/2$, then $\mathcal{F}^a$ is a FRFT over an angle $\alpha$ and the notation suggests that it can be interpreted as the $a$th power of the Fourier operator. Its meaning is that a signal whose energy is concentrated in some domain $D$ of the time-frequency plane, will after the transform $\mathcal{F}^a$ have its energy concentrated in a domain that is obtained by rotating $D$ over an angle $\alpha$ from time axis towards frequency axis. The FRFT $\mathcal{F}^a$ of order $a \in \mathbb{R}$ is a linear integral operator that maps a given function (signal) $f(x)$, $x \in \mathbb{R}$ onto $f_a(\xi)$, $\xi \in \mathbb{R}$ by

$$f_a(\xi) = (\mathcal{F}^a f)(\xi) = \int K_a(\xi, x) f(x)\,\mathrm{d}x$$

where the kernel is defined as follows. Set $\alpha = a\pi/2$ then

$$K_a(\xi, x) = C_\alpha \exp\left\{-i\pi\left(2\frac{x\xi}{\sin\alpha} - (x^2+\xi^2)\cot\alpha\right)\right\},$$

with

$$C_\alpha = \sqrt{1 - i\cot\alpha} = \frac{\exp\{-i[\pi\mathrm{sgn}(\sin\alpha)/4 - \alpha/2]\}}{\sqrt{|\sin\alpha|}}.$$

For $k \in 2\mathbb{Z}$, limiting values are taken which means that

$$K_{4n}(\xi, x) = \delta(\xi - x) \quad \text{and} \quad K_{2+4n}(\xi, x) = \delta(\xi + x), \quad n \in \mathbb{Z}.$$

The inverse of $\mathcal{F}^a$ is given by $\mathcal{F}^{-a}$: just change the sign of angles.

The discrete FRFT (DFRFT) is for the FRFT like what the DFT is for the Fourier transform. There is not a unique definition, but there is some general agreement on the definition given by Candan and co-workers. It is based on a fractional power of the DFT matrix. For details we refer to the literature [2,3,12,11,1].

For the DFRFT of an image, we consider subsequently the DFRFT of the rows to give a row-transformed matrix, and then apply the DFRFT on the columns of the resulting matrix. Note that the transformation angle for rows and columns can be different.

## 3 Watermark embedding

The general idea of watermarking in a transform domain is to first compute the transform, then modify the coefficients of the transform and transform back.

The method proposed in [4] goes along this idea. First compute the DRFFT of the image. The watermark should not be embedded in the smallest coefficients because that would make it very sensitive to e.g., noise removing or compressing operations, but it should neither be embedded in the largest coefficients because that would disturb the image too much while we want the watermark to be hidden for normal visual inspection. Let us sort the DFRFT coefficients according to their magnitude and denote the sorted array as $\{S_i = V_i + jW_j : |S_i| \leq |S_{i+1}|, i = 1, 2, \ldots\}$. Then we embed the watermark into the coefficients $S_i$, $i = L+1, L+2, \ldots, L+M$.

The watermark itself is a sequence of $M$ complex numbers. The real and imaginary parts are drawn from a normal distribution with mean zero and variance $\sigma^2/2$. Let us denote the watermark as $\{s_i = v_i + jw_i : i = L+1, \ldots, L+M\}$.

The sorted vector $S_i$ is then modified to embed the watermark by setting

$$S_i^w = S_i + v_i|V_i| + jw_i|W_i|, \quad i = L+1, \ldots, L+M.$$

After the modified array $S_i^w$ has been rearranged in the original two-dimensional array where

the original $S_i$ come from, the watermarked image is then obtained by computing the inverse DFRFT.

## 4 Detection of the watermark

The image that is submitted to see if it contains the watermark is first transformed with the same DFRFT angles and the result is put in the same order as was used for the embedding. Because the image may have undergone modifications that may have been deliberate or accidental attacks to the image, we do not get the coefficients $S_i^w$ but we get some coefficients $S_i^a$ instead.

Next the detection value $d$ is computed as

$$d = \sum_{i=L+1}^{L+M} [v_i - jw_i] S_i^a.$$

The expected value of $d$ (assuming that the watermark and the image are uncorrelated and assuming that $S_i^a = S_i^w$) is given by

$$E[d] = \frac{1}{2}\sigma^2 \sum_{i=L+1}^{L+M} (|V_i| + |W_i|).$$

In [4], it is decided that a watermark has been detected if the computed value of $d$ is larger than a threshold, which is set to $E[d]/2$. This seems reasonable since an image without a watermark has $E[d] = 0$.

However, when attacks have modified the image, then the expected value may change drastically. For example, noise may have been added which increases the total energy, or after cropping several of the watermarked DFRFT coefficients may have been set to zero, which may give a much lower value for the computed $d$ which produces the false conclusion that there is no watermark, while it may still be perfectly detectable.

Therefore the following method is proposed. After the modified image has been DFRF-transformed with the correct angles, then the value of $d$ is computed as suggested above. Next the detection value $d^t$ for a sufficiently large number of random watermarks $s^t$, $t = 1, \ldots, n$ is computed. The average (say $\mu$) and the standard deviation (say $\sigma$) of these $d^t$ give for the manipulated image

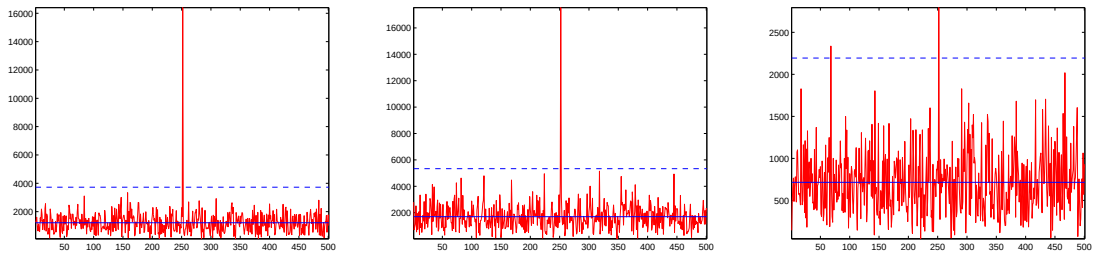



Fig. 1. Computation of the detection value $d$ for 500 random watermarks and with the correctly embedded watermark on the middle position 251. For jpeg compressed image (left), the same with added noise (middle), noisy image cropped (right)

(which may be cropped, noise corrupted,...) an estimate for the average value of $d$ when the correct watermark is not embedded. The value $d$ for the correct watermark should stand out above this average. Here "stand out" can be defined as being larger than $\tau = \mu + p\sigma$ where $p$ is a suitable number (say 4). The value of $p$ can be used to avoid wrong conclusions. If $p$ is too large, then the real watermark may not be detected and if $p$ is too small, some random watermarks may be identified as the right one, while they are different form the correct one.

It is illustrated in Figure 1 that the threshold may vary depending upon the modification that has been induced on the image.

So to do a correct detection one has to know

- The correct angles for the DFRFT
- The correct place and length of the watermark (i.e., $L$ and $M$ and the ordering of the DFRFT coefficients).
- The correct watermark, i.e., the sequence $\{s_i : i = L+1, \ldots, L+M\}$ or a way to generate it.

These data have to be stored during the embedding stage and have to be passed on to the detection algorithm.

## 5 Variations on this theme and computational aspects

In the previous sections we have described the basic idea of embedding a watermark into a digital gray-scale image. This idea can be extended or simplified in many different ways. We

give some examples.

- What has been said above for one watermark embedded in an image can be generalized when more than one of these watermarks are emdedded, each one with its own FRFT angles, its own location and its own sequence.
- Instead of working with the DFRFT, one could also use the discrete fractional cosine or sine transform (DFRCT/DFRST) [13] which are computationally a bit more efficient.
- For color images, one may embed watermarks in each of the three components whether they are YUV or RGB decomposed. In the case of YUV representation, the Y-component is most robust and can absorb the highest level of watermark information.
- What has been said about the DFR(F/C/S)T van be repeated for the corresponding random transforms [9,7,8]. Here the eigenvectors of the DFRFT matrix are placed in random order. This is an extra parameter that can be passed on to the detection algorithm and it makes it even harder to detect the watermark if one does not have all the parameters.
- On the other hand, a simpler watermark can be embedded by only modifying the real or the imaginary part of the DFRFT coefficients.

## 6 Some numerical experiments

The example image that we take is Lena, shown on the left in Figure 2. It is a 512×512 jpeg



Fig. 2. The original image (left) and the watermarked image (right).

image. The image is transformed by the DFRFT with powers 0.8 and 0.5 in the $x$ and $y$

direction respectively. Then three watermarks are embedded using the method described above with parameters $L, M$ and $\sigma^2$ given by

| watermark | 1 | 2 | 3 |
|---|---|---|---|
| $L$ | 96000 | 90000 | 100000 |
| $M$ | 800 | 800 | 400 |
| $\sigma^2$ | 50 | 100 | 30 |

The watermarked image is saved as a jpeg file (8BPP) and the result is submitted to the detection algorithm that is described above. As can be seen on Figure 2, the watermarked image can not be visually distinguished from the original.

The mean and standard deviation of the detection value is computed using 1000 [3] random watermarks with the same parameters as the correct one. The threshold is fixed at the mean plus 4 times the standard deviation. These values and the PSNR can be found on row (1) of the table in Figure 3. The values $\mu_i, \sigma_i, \tau_i, d_i$ refer to the mean, the standard deviation, the threshold and the detection value of watermark $i$. The corresponding plots are given on the first row of Figure 4. In this way, it can be seen that the three embedded watermarks are detected.

The next attack is adding noise to the watermarked image. To the watermarked and compressed image, a zero mean Gaussian noise with standard deviation 50 is added. The result is again jpeg compressed and given again to the detection algorithm. The resulting image can be seen on the left in Figure 5. The mean and standard deviation of the randomly selected watermarks is definitely different from what was found in the previous case. These are given, together with the PSNR of the image in row (2) of the table in Figure 3. The 1000 values of $d$ and the correct one on position 501 for the three embedded watermarks is what is shown in the middle plot of Figure 4. Again the three watermarks were detected.

Finally we crop the previous noisy image by taking only the first 100 rows out of the 512 rows in the image. See the right image of Figure 5. The result is again submitted to the detection algorithm and the mean and standard deviation for 1000 random watermarks is computed. The result is shown on row (3) of Figure 4. The values of the PSNR, the mean and the threshold are given on row (3) of the table in Figure 3. Now the first two watermarks are detected, but the third is not. For another type of cropping (see at the bottom of Figure 5 where 150 rows

[3] We use a 1000 random sequences in this example, but for practical computations, a 100 random sequences is certainly sufficient to estimate the mean and the standard deviation.

| | PSNR | $\mu_1$ | $\sigma_1$ | $\tau_1$ | $d_1$ | $\mu_2$ | $\sigma_2$ | $\tau_2$ | $d_2$ | $\mu_3$ | $\sigma_3$ | $\tau_3$ | $d_3$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 39.41dB | 1236 | 664 | 3892 | 17772 | 1716 | 905 | 5334 | 27999 | 633 | 342 | 2001 | 4433 |
| 2 | 28.95dB | 1781 | 915 | 5443 | 19412 | 2527 | 1328 | 7839 | 27838 | 943 | 496 | 2927 | 4590 |
| 3 | 0.94dB | 681 | 345 | 2061 | 1847 | 952 | 503 | 2966 | 4149 | 387 | 204 | 1204 | 969 |

Fig. 3. The values obtained for the compressed and watermarked image (1), when this result is contaminated by additive noise (2), and when that result is cropped (3)

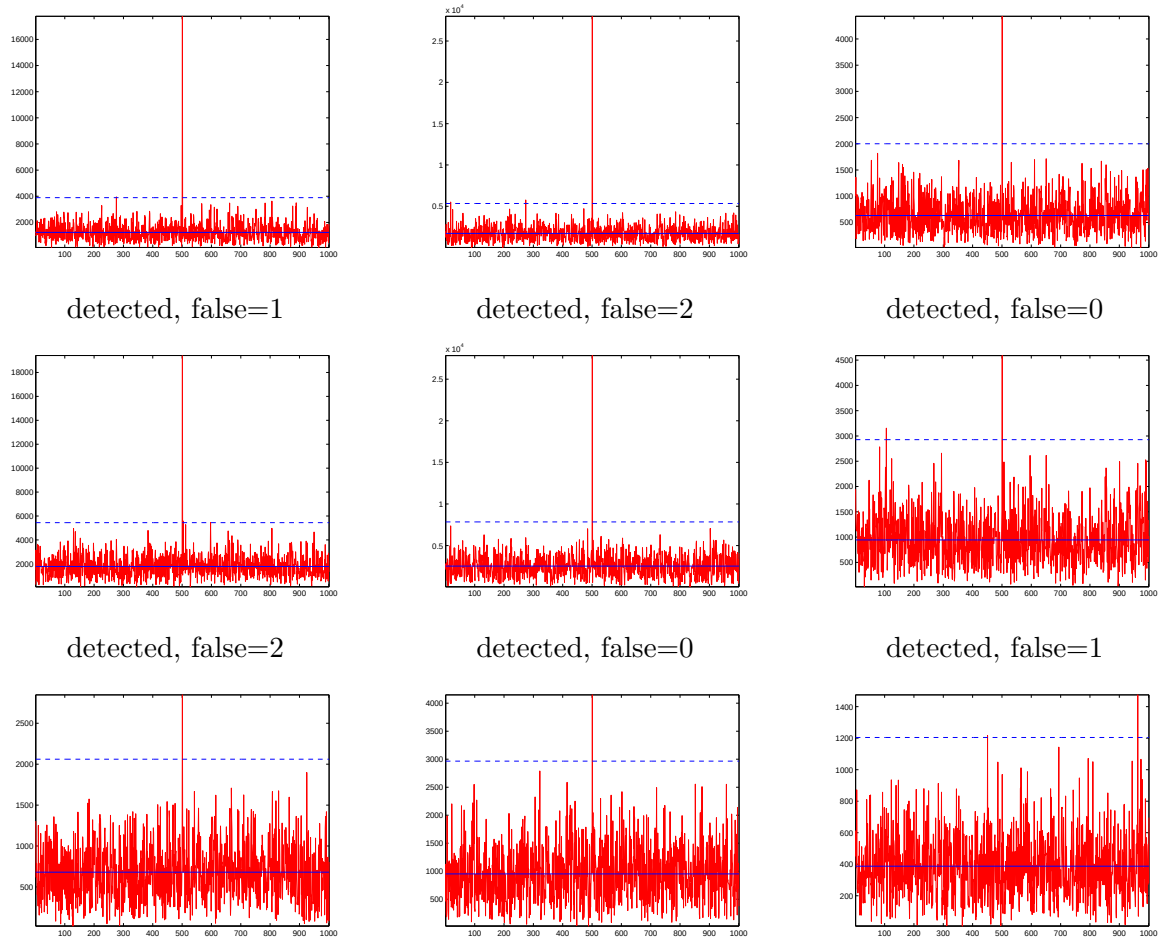



Fig. 4. A 1000 samples for three watermarks (columns) for the compressed image (top), the noisy compressed image (middle), and the noisy cropped and compressed image (bottom).



Fig. 5. The watermarked image with added noise (left) and its cropped version (right). Another cropping at the bottom.

and columns are removed on all borders), similar results are obtained. Indeed the effect of the watermark is spread out over the whole image and can be recovered from any part of the image.

Looking at the plots of Figure 4, it is also seen that there are sometimes 'false alarms'. Some of the randomly generated watermarks also give a detection value above the threshold. It is clear from these plots that a threshold $\tau = \mu + p\sigma$ with $p = 4$ is not a bad choice. If we take $p$ larger, we require the detection value of the correct watermark to be a more pronounced outlier. If we decrease $p$, we are at a higher risk that we detect a watermark that is not the correct one. It could of course accidentally happen that the falsely detected watermark is strongly correlated with the correct one, in which case of course its detection value should be in the neighborhood of the detection value of the correct watermark.

The number of false alarms for the three different watermarks on the three different corrupted situations is mentioned in Figure 4 (the "false=n" text). Obviously, in these particular situations the chance of having a false alarm when the image is not too much corrupted is very small.

To illustrate the sensitivity with respect to the powers that are used in the DFRFT, we have produced Figure 6. We have used the same image of Lena with only the first watermark em-

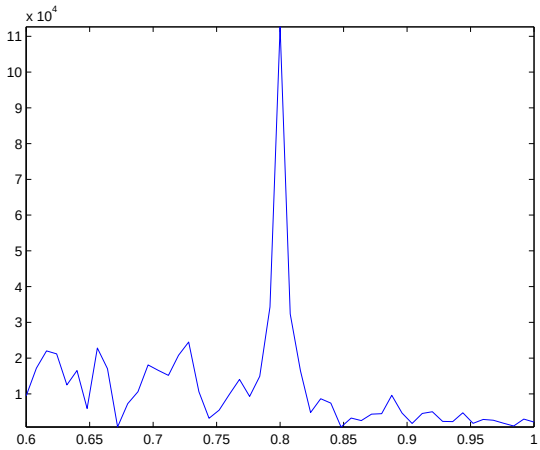



Fig. 6. The sensitivity of the DFRFT power.

bedded via the DFRFT with powers 0.8 and 0.5. We have computed the detection value of the true watermark but via a DFRFT using the powers $a_1$ and $a_2$ where $a_2$ has the exact value 0.5, but choosing 51 equidistant values between 0.6 and 1.0 for $a_1$. The figure illustrates clearly that only the correct value 0.8 gives a peak for the detection value, while missing it by less than 1%, already gives a much lower value.

## 7 Conclusion

We have discussed the implementation of a watermarking technique for digital images based on the discrete Fractional Fourier Transform.

It is shown that the variety of parameters that can be used make it more difficult to damage the watermark and for someone who does not have all the key-parameters, it will be much harder to detect it. These parameters are the location of the watermark, the watermark (random sequence) itself or its parameters to generate it, and the angles of the DFRFT used.

A proposal is made for a threshold for the detection value. It is not too large so that the correct watermark is recognized and it is not too small so that false alarms are avoided in most cases. Yet it adapts itself to the modifications that the watermarked images has suffered.

The technique is illustrated for jpeg images, but it can be applied for other formats in the same way. The transformation of the image from one format to another will however make the watermark unrecognizable even of it is transformed back to the original format unless of course the original image is perfectly recovered.

The matlab code for the embedding and corresponding detection algorithm together with a test-code, can be found on the website

`http://www.cs.kuleuven.be/cwis/research/nalag/research/software/FRFT/`

## References


[1] A. Bultheel and H. Martínez Sulbaran. Computation of the fractional Fourier transform. *Appl. Comput. Harmonic Anal.*, 16(3):182–202, 2004.

[2] Ç. Candan. *The discrete Fractional Fourier Transform.* MS Thesis, Bilkent University, Ankara, 1998.

[3] Ç. Candan, M.A. Kutay, and H.M. Ozaktas. The discrete Fractional Fourier Transform. *IEEE Trans. Sig. Proc.*, 48:1329–1337, 2000.

[4] I. Djurović, S. Stanković, and I. Pitas. Digital watermarking in the fractional Fourier transform domain. *J. of Network and Computer Applications*, 24:167–173, 2001.

[5] J. Guo, Z. Liu, and S. Liu. Watermarking based on discrete fractional random transform. *Optics Communications*, 272(2):344–348, 2007.

[6] M. Kutter. Watermarking world. www.watermarkingworld.org.

[7] Z. Liu, Q. Guo, and S. Liu. The discrete fractional random cosine and sine transforms. *Optics Communications*, 265(1):100–105, 2006.

[8] Z. Liu, Q. Guo, and S. Liu. Erratum to 'the discrete fractional random cosine and sine transforms'. *Optics Communications*, 267(2):530, 2006.

[9] Z. Liu, H. Zhao, and S. Liu. A discrete fractional random transform. *Optics Communications*, 255(4-6):357–365, 2005.

[10] X.M. Niu and S.H. Sun. Digital watermarking for still image based on discrete fractional Fourier transform. *J. Harbin Inst. Technology*, 8(3):309–311, 2001.

[11] H.M. Ozaktas, Z. Zalevsky, and M.A. Kutay. *The fractional Fourier transform.* Wiley, Chichester, 2001.

[12] S.C. Pei and M.H. Yeh. Two-dimensional discrete Fractional Fourier Transform. *Signal Processing*, 67:99–108, 1998.

[13] S.C. Pei and M.H. Yeh. The discrete fractional cosine and sine transforms. *IEEE Trans. Sig. Proc.*, 49:1198–1207, 2001.

[14] N. Vancluysen. Digitaal watermerken van beelden. Master thesis, Dept. Computer Science, K.U.Leuven, 2007. (In Dutch).