# Radon and Ridgelet transforms applied to motion compensated images

Lorenzo Granai
Signal Processing Institute, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
Email: lorenzo.granai@epfl.ch

August 2002*

Keywords: *Video Coding, Displaced Frame Difference, Motion Compensated Images, Ridgelet, Wavelet, Radon Transform, Iterative Algorithms, Hybrid Coding.*

## 1 Introduction

Images are typically described via orthogonal, non-redundant transforms like wavelet or discrete cosine transform. The good performances of wavelets in one-dimensional domain are lost when they are applied to images using 2D separable basis since they are not able to efficiently code one-dimensional singularities.
The Ridgelet transform achieves very compact representation of linear singularities in images; instrumental in the implementation of the Ridgelet is the Radon transform, which is a powerful tool to find directions where images present line features. They can offer an important contribution in order to detect and represent edges, which in natural images are very important components and especially relevant from a visual point of view.
Speaking about video coding, following the classical hybrid scheme, the motion compensation procedure produces a displaced frame difference (dfd) that appears as an edge dominated image. It is intuitive that ridgelet transform can be a good tool for coding dfd, given its capacity to offer a sparse analysis of singularities.
This research report investigates the application of a discrete implementation of ridgelet and Radon transform to motion compensated images. Section 2 offers a brief review of ridgelet and Radon theory, including a description of a finite, discrete algorithm. Section 3 illustrates the application of discrete ridgelets to dfd, giving some results. Section 4 proposes two new algorithms that aim to exploit the ability of Radon transform to detect straight lines. Section 5 concludes the work presenting some possible future developments of these topics.

* This research has been developed in the Multimedia Signal Processing Laboratory, NTT DoCoMo, Japan and has been partly supported by NTT DoCoMo, Inc.

## 2 Radon and Ridgelet Theory

### 2.1 Continuous Ridgelet Transform

Let us briefly review the continuous ridgelets transform, defined by Candes [1], starting from the Radon transform, instrumental in its implementation. Given an integrable bivariate function *f*(***x***), its Radon Transform (RDN) is defined by:

$$RDN_f(\theta,t) = \int_{R^2} f(x)\delta(x_1\cos\theta + x_2\sin\theta - t)dx. \quad (1)$$

Basically the Radon operator maps the spatial domain into the projection domain ($\theta$, $t$), in which each point corresponds to a straight line in the spatial domain; conversely, each point in the spatial domain becomes a sine curve in the projection domain.

The Continuous Ridgelet Transform (CRT) is nothing else than the application of a 1D-wavelet ($\psi_{a,b}(t) = a^{-1/2}\psi((t-b)/a)$) to the slices of the Radon transform:

$$CRT(a,b,\theta) = \int_R \psi_{a,b}(t)RDN_f(\theta,t)dt = \int_{R^2}\psi_{a,b,\theta}(x)f(x)dx \quad (2)$$

where the ridgelets $\psi_{a,b,\theta}(x)$ in 2-D are defined from a wavelet-type function $\psi(x)$ as

$$\psi_{a,b,\theta}(x) = a^{-1/2}\psi((x_1\cos\theta + x_2\sin\theta - b)/a). \quad (3)$$

This shows that the ridgelet function is constant along the lines where $x_1\cos\theta + x_2\sin\theta = const$.

Comparing ridgelets with wavelets we observe that the parameters of the first are scale factor and line position (respectively $a$ and $(b,\theta)$ in (2)), while the latter uses scale factor and point position. As a consequence, wavelets are very effective in representing objects with isolated point singularities, while ridgelets are very effective in representing objects with singularities along straight lines.

It is interesting to notice that if we apply a 1D Fourier transform along $t$ instead of a wavelet we will obtain the 2D Fourier transform of $f(x)$. This result is known as projection-slice theorem and is illustrated in figure 1.

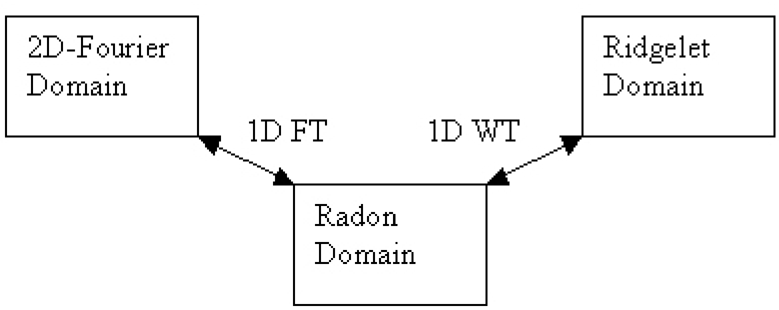



Fig.1 Projection-Slice Theorem

### 2.2 Ridgelet Optimality for Linear Singularities

Theoretically Ridgelets turn out to be optimal for representing bivariate functions with linear singularities. Being *H* the Heavyside $H(t) = 1_{t>0}$, *s* a nonnegative integer and *g* a smooth arbitrary function with compact support and finite Sobolev norm $\|g\|_{W_2^s}$, the following theorem [8,9] states that we obtain a rate of approximation as if the function were not singular, simply by thresholding the orthonormal ridgelet expansion.

<u>Theorem:</u> Let $g \in W_2^s(R^2)$ and $f(x_1, x_2) = H(x_1 \cos\theta_0 + x_2 \sin\theta_0 - t_0) \cdot g(x_1, x_2)$.

Then the sequence $\alpha_i$ of orthonormal ridgelet coefficients of *f* satisfies

$\#\{|\alpha_i| \geq 1/n\} \leq C \cdot n^{2/(s+1)} \|g\|_{W_2^s}$.

The constant *C* does not depend on *f*. As a consequence, the n-term approximation $f_n$, obtained by keeping the terms corresponding to the *n* largest coefficients in the ridgelet expansion, satisfies

$\|f - f_n\|_2 \leq C \cdot n^{-s/2} \|g\|_{W_2^s}$.

This is unlike all systems currently in use: for example the same singularity causes partial wavelet reconstructions to converge very slowly while its effect on the approximation rate of truncated ridgelet series is negligible.

If singularities along arbitrary curves are studied the previous result does not hold anymore and ridgelets loose their optimality.

### 2.3 Finite Ridgelet Transform

In order to apply ridgelets to digital images a discrete transform is needed, and this leads to the research of a discrete Radon transform. In [2], Do proposes a new procedure that results to be invertible, orthogonal and achieves perfect reconstruction: the *Finite RIdgelet Transform* (FRIT). FRIT is based on the *Finite RAdon Transform* (FRAT) [10], which is defined as summations of image pixels over a certain set of “lines”. Those lines are defined in a finite geometry in a similar way as the lines for the continuous Radon transform in the Euclidean geometry. Denote $Z_p = \{0,1,.., p-1\}$, where $p$ is a prime number. Note that $Z_p$ is a finite field with modulo *p* operations.

The FRAT of a real function *f* on the finite grid $Z_p^2$ is defined as:

$$r_k[l] = FRAT_f(k,l) = \frac{1}{\sqrt{p}} \sum_{(i,j) \in L_{k,l}} f[i,j] \quad (4).$$

Here $L_{k,l}$ denotes the set of points that make up a line on the lattice $Z_p^2$, i.e.

$$L_{k,l} = \begin{cases} \{(i,j) : j = (ki + l)(\text{mod}\, p), i \in Z_p\} \; if \; 0 \leq k < p \\ \{(l,j) : j \in Z_p\} \; if \; k = p \end{cases} . \quad (5)$$

The inverse transform is obtained through the *Finite Back-Projection operator* (FBP) defined as a sum of Radon coefficients of all the lines that go through a given point. Here *f* is supposed to be a zero-mean image:

$$FBP_r(i,j) = \frac{1}{\sqrt{p}} \sum_{(k,l)\in P_{i,j}} FRAT_f(k,l), \quad (6)$$

from (5) we can found that $P_{i,j}$ is:

$$P_{i,j} = \{(k,l) : l = (j - ki)(\text{mod } p), k \in Z_p\} \cup \{(p,i)\}. \quad (7)$$

Substituting (6) into (4) we obtain that

$$FBP_r(i,j) = \frac{1}{p} \sum_{(k,l)\in P_{i,j}} \sum_{(i',j')\in L_{k,l}} f[i',j'] = \frac{1}{p}\left( \sum_{(i',j')\in Z_p^2} f[i',j'] + p \cdot f[i,j] \right) = f[i,j] \quad (8)$$

and so the perfect reconstruction is achieved.

It is easy to compute that the number of operations required by FRAT is $p^3$ addictions and $p^2$ multiplications, so comparable with other transforms like 2D-FFT. A drawback of this discrete implementation is the wrap-around effect, already observed by Do.

The finite ridgelet can be obtained by applying a 1D-Discrete Wavelet Transform (DWT) to FRAT projections. We used the Daubechies-4 function. In all the examples presented in section 2 the FRAT is obtained using the optimal ordering for the finite Radon transform coefficients presented by Do in [2], but we must observe that using the classical ordering the results do not differ significantly; in fact this ordering technique is not designed for dfd.

### 2.4 Size Matters

FRIT needs an input image of size *p*x*p,* where *p* is a prime number, and this is an important limitation of this algorithm. Moreover wavelets usually require a dyadic length signal and this is absolutely incompatible with the FRAT output. In the test we made, we extended the length of the signal from *p* to *m*, where *m* is defined as:

$$m = \min\{n \in N : (n > p) \text{ and } (n = 2^d), d \in N\}.$$

Typically we took *p*=31 or *p*=127 and so *m*=32 or 128 respectively.

## 3 FRIT Applied to Dfd

Motion compensated images are a special kind of image, characterized by a different range respect to natural images and a high frequency behavior. Usually, as already observed, they present a lot of edges. We aim to catch these features exploiting the directional information given by the Radon transform and consequently by the FRIT. For their nature ridgelets are able to represent efficiently straight lines, but have problems with curves. As in a dfd either line or curve edges can be found, an idea is to apply this transform to small blocks where each edge can be approximated by a straight line. These blocks can be obtained from a quad-three algorithm [4,5] or localized in the maximum energy zones of the image. Even if the ridgelet has a good behavior on test images composed by one or more lines, when applying it to real dfd blocks the result is not so promising: comparing FRIT with wavelets (a 2D-DWT with Daubechies9,7 basis functions) the error is lower only at very high compression rates. Moreover when the number of FRIT coefficients is increasing a very unpleasant noise-like effect appears (see figure 4a for an example of this effect).

In table 1 we can see the mse for a block 31x31 of the Stefan sequence.

| Coefficients | Frit mse | Wavelet mse |
|---|---|---|
| 5 | 304.5 | 334.3 |
| 10 | 270.5 | 290.8 |
| 50 | 191.6 | 130.3 |
| 100 | 139.4 | 80.7 |
| 200 | 80.2 | 38.7 |

Table 1: MSE for FRIT and Wavelet coding

Examining bigger blocks results do not improve. In the majority of the cases we observed that FRIT is not able to efficiently code a dfd, even if, given its completeness, it guaranties a perfect reconstruction.
However the very first FRIT coefficients are able to detect the direction of the main lines and this can be really useful for coding zones mainly characterized by a well-defined line.

### 3.1 Hybrid Coding

A possibility to exploit the advantages of FRIT is to apply it to a block of a dfd and check if it is able to detect lines with the first coefficients. We propose a method composed by two stages: first the FRIT is used to represent lines with its biggest coefficients, afterwards the reconstructed image is subtracted from the original one and the residual can be coded with an other method. Here the second stage is realized with wavelets. Figure 2 illustrates the scheme of this algorithm.
For this approach block's dimensions can be larger than in the previous case, since we simply aim to detect the main structure of an image and then pass the residual part to the following step. It is not necessary to split the image in order to find straight lines, since only the features that are really efficiently represented by the ridgelet transform are considered; all the rest is left to the second stage. It could be possible to apply the FRIT to the whole image too. Please, note that in the following scheme delays are not indicated.

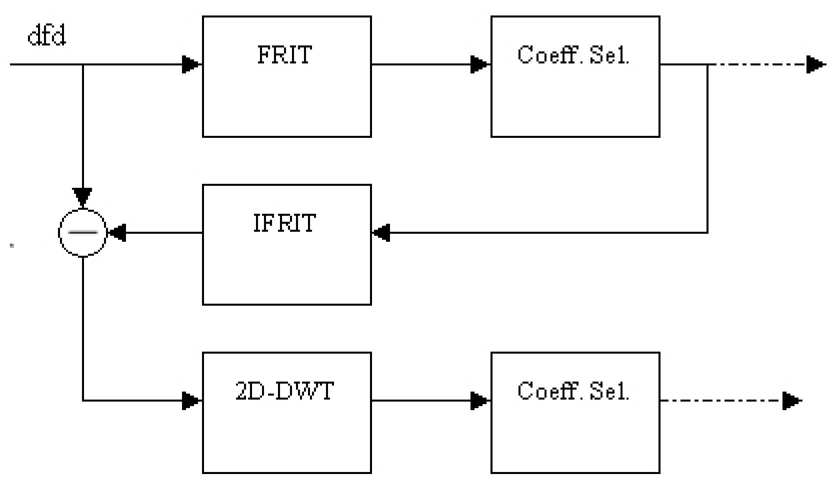



Fig.2a Hybrid FRIT-based dfd Encoder

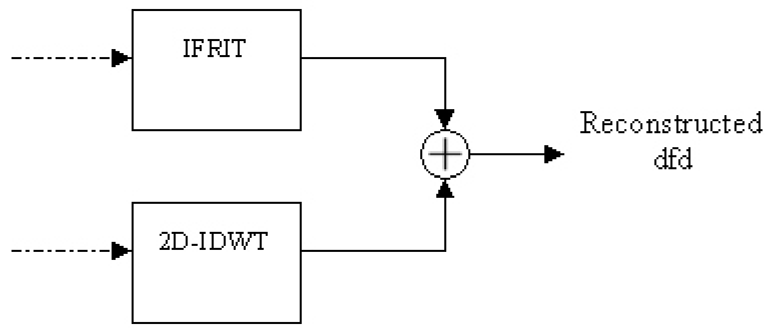



Fig.2b Hybrid FRIT-based dfd Decoder

### 3.2 Results

Let us consider the case of a block of 127x127 pixels, extracted from the sequence "Stefan" and that we can call "Lufthansa". This part of the dfd is reproduced in figure 3a but obviously the pixels values are shifted, being the original range of the frame [-255, 255]. A line that crosses the frame and is well coded by the ridgelet transform characterizes the block. Figure 3b describes the mse decay with an increasing number of FRIT and wavelet coefficients: it is evident that at the beginning the FRIT has a better behavior but then a lot of interferences appears and wavelets surpass ridgelets.

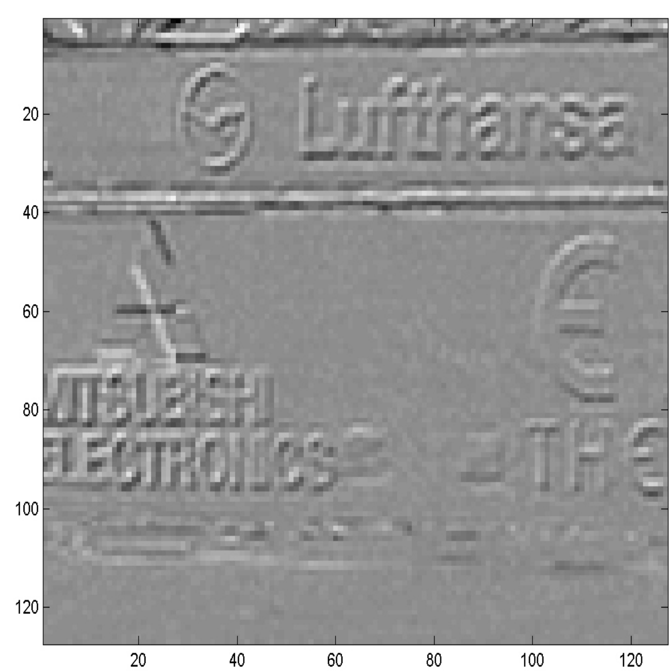



Fig. 3a: Block "Lufthansa"

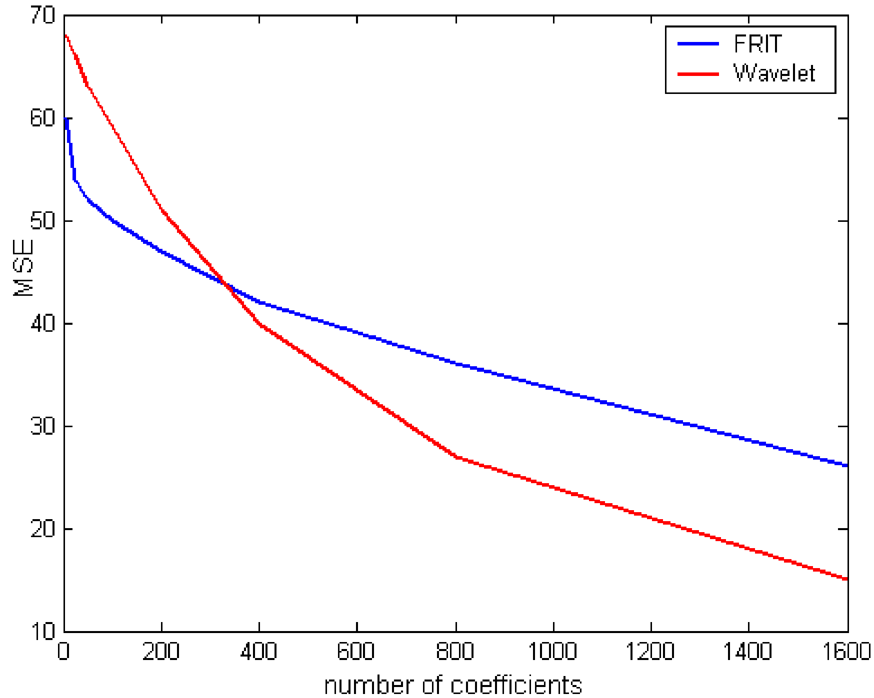



Fig.3b: Mse decay for wavelets and ridgelets (in blue)

The hybrid scheme selects the 5 biggest FRIT coefficients (in absolute value) and then applies the 2D-DWT on the residual image, taking again its 395 biggest coefficients (in absolute value).

The mse obtained with 400 coefficients in all is 40.1, 42.3, and 34.8 respectively for wavelet, ridgelet and hybrid scheme. Figure 4 shows the reconstructed dfd Lufthansa blocks. In this case 5 FRIT coefficients are enough for individuating the line and so the wavelet can code the other features of the image. Note that the compression ratio is more than 40. Of course this hybrid-coding scheme is particularly efficient when dealing with blocks that include lines. In our experiments the 2D-DWT is based on the Daubechies9,7 functions.

Obviously the second stage can be realized using other techniques than wavelet, such as DCT or Matching Pursuit [6,7].

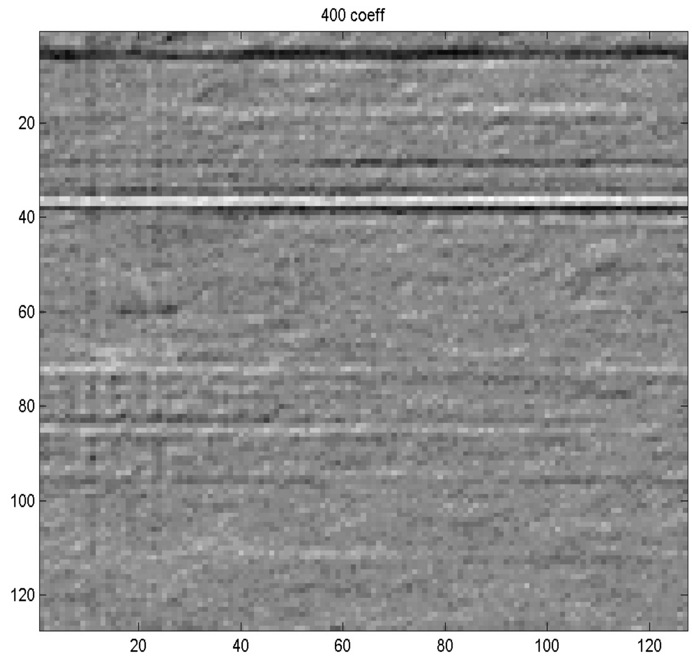

400 coeff
20
40
60
80
100
120
20
40
60
80
100
120

Fig.4a

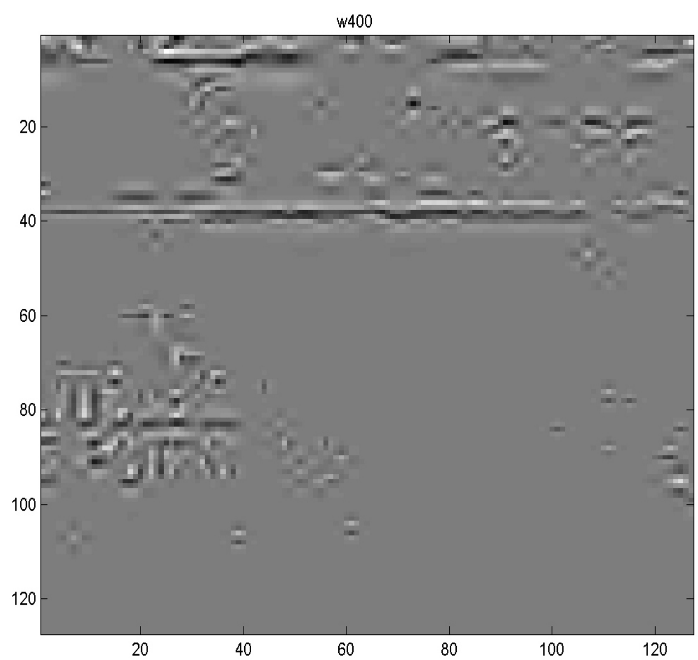

w400
20
40
60
80
100
120
20
40
60
80
100
120

Fig.4b

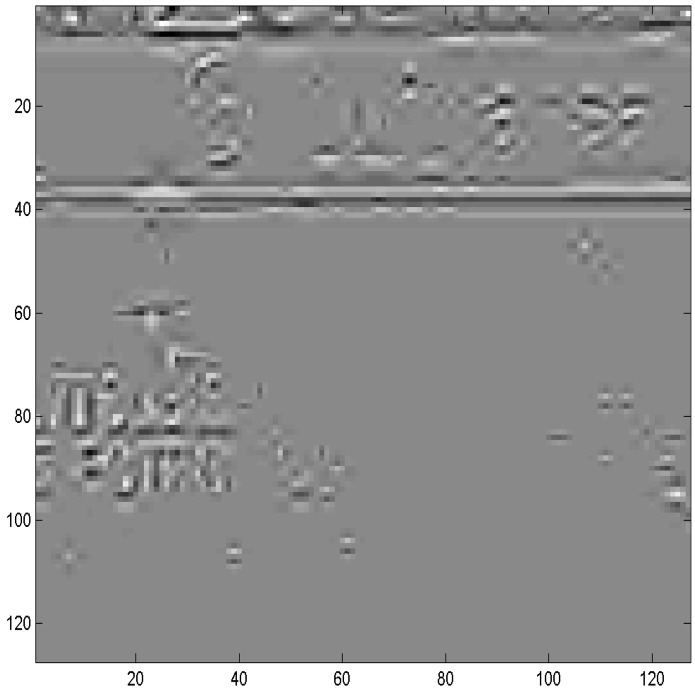



Fig.4: Block Lufthansa reconstructed with 400 coefficients:

a) Ridgelet
b) Wavelet
c) 5 Ridgelet and 395 Wavelet

## 4 Two Iterative Schemes that Exploit Radon Information

In a nutshell, the proposed scheme has good performances but it is not able to efficiently detect the real features of an image when they are not straight long lines. The problem is that a big FRIT coefficient can be generated from high energy zones of the dfd that are not necessary lines that cross all the frame. Figure 5 shows a reconstructed frame where a line is longer than the original one and another has been created where there were only high-energy pixels.
In this section we look for a method that is able to identify lines with their real length and does not create lines where they are not present or unnecessarily extend the lines throughout the whole image.

Two completely new algorithms for exploiting the directional information given by the Radon transform are proposed: they represent a sort of pre-processing for a dfd and the residual image that they produce will be an input for another coding stage. Basically the high level block scheme is shown in figure 6 and follows this points:

a. Pre-process the dfd with an iterative algorithm that find the straight lines.
b. The residual image produced by (a) is passed to a 2D encoder.
c. Reconstruction: output of (a) and (b) are separately decoded and finally added.

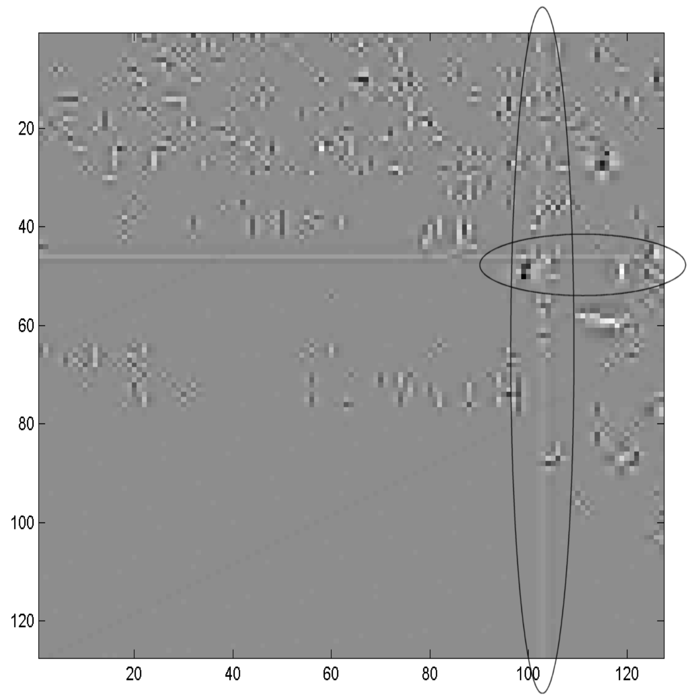


Fig.5: Problems

The iterative algorithm that is the core of the procedure can be one of the two illustrated in subsections 4.1 and 4.2.

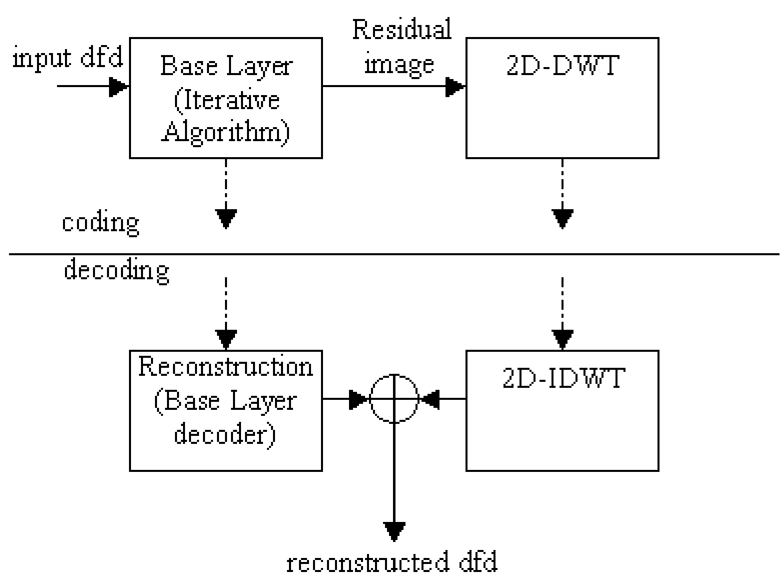


Fig.6: High level codec scheme

## 4.1 Projecting the line on the image

The main idea of this scheme is to compute the projection between the original image and the one reconstructed with one radon coefficient. Two goals are so reached: first, it will be known how well the ridgelet has been able to detect the line, then it will be possible to keep only the part of the lines given by the FRAT that matches the image. Note that in this new algorithm the Ridgelet is not used anymore, but the directional information is only given by the Radon transform. In fact, as only the very first coefficients are kept it is no more useful to apply the 1D-DWT to Radon slices.
To guarantee perfect reconstruction we propose an iterative technique (see figure 7) based on the finite Radon transform: at each step the biggest coefficient of the FRAT is used to reconstruct a line, which is then projected onto the image.

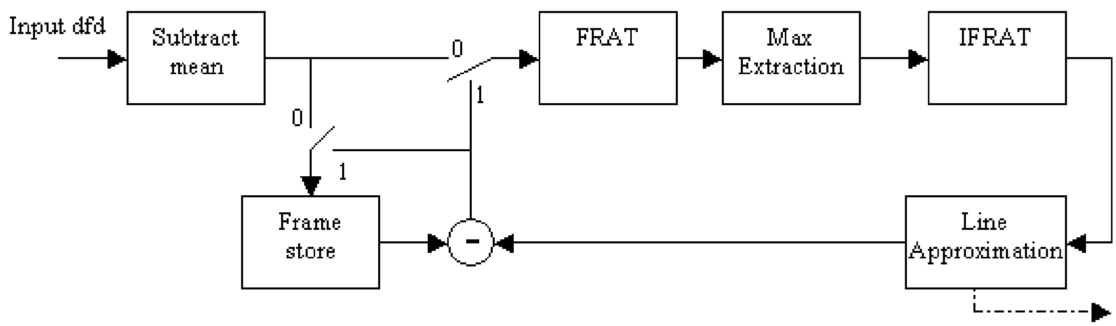



Fig.7: Iterative scheme that finds out the main line on a dfd

The result of projecting the line onto the image is compared with a threshold and only the pixels that exceed it (and so the part of the line that matches the original one) is kept. Now rejecting isolated pixels we create a continuous line and, finally, the value of each pixel is set to the average of the projection. In this way an approximation of the original line is obtained.
After this decomposition residual image can be coded with other methods (Wavelet, DCT, Matching Pursuit…) as proposed in the previous section. For the first stage each line will be coded by means of its value, length and starting point.
If the value of the projection of the found line onto the image is too low it is possible to completely reject this line. It means that the image has no particular structures that can be seen by the Radon transform. In this case the coding task is completely transferred to the second method.
As already observed in section 3.1 the block size will be as big as possible, compatibly with the tie that $p$ must be prime.

## 4.2 Maximum distortion line

Appling the discrete Radon transform to a dfd we basically sum the values of all the pixels along a line whose position and orientation are determined by the parameters $(k,l)$ see equation 4).
Roughly speaking, the biggest FRAT coefficient (in absolute value) indicates a line that presents a huge number of pixels with high values and the same sign. Keeping the notation of section 2.3 let us define $M$ as:

$$M = \max_{k,l}(FRAT(|f(i,j)|)) \qquad (9)$$

The position of the maximum $M$ indicates the line with highest distortion in the image, in the sense that is the line where the error committed by the motion estimation was the biggest.

The second procedure proposed is based on this easy way to detect the maximum distortion line. Afterwards this line is extracted from the image, coded by wavelet decomposition and subtracted from the image; this step is repeated iteratively, in order to find the main features of the frame. Finally the residual image is coded with a different technique (here, 2D-DWT) as already described. Few wavelet coefficients are enough to well represent the line.

A high coefficient of the FRAT of the absolute value of a frame can be produced by points with high values and the same sign (so basically an edge) or by points with high values but alternate sign. Coding these lines with wavelets permits to deal with both cases. The first stage of the algorithm is illustrated on figure 8.

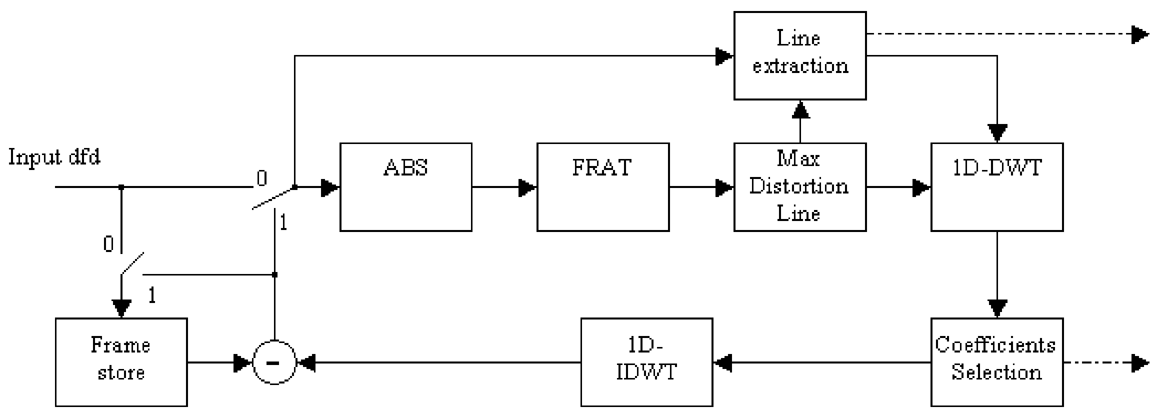



Fig.8: Iterative scheme that finds out the max energy line on a dfd and represent it through wavelets

Note that there is no IFRAT in the scheme: the FRAT is used only to find out the signal to pass to the 1D-wavelet encoder.

The parameters to transmit are line indexes $(k,l)$ and selected wavelet coefficients.

The decoder has to receive the indexes that identify the line *(k,l)* and of course the wavelet coefficients.

### 4.3 Results

For a question of clearness and simplicity let us indicate with “frat+proj+wave” the algorithm presented in section 4.1 and with “absradon+wave” the one in section 4.2.

The motion compensated image shown in figure 9 has been taken from the sport sequence “Jleague”. This figure illustrates the original dfd and the lines that are detected by the first procedure and subtracted from the input image.

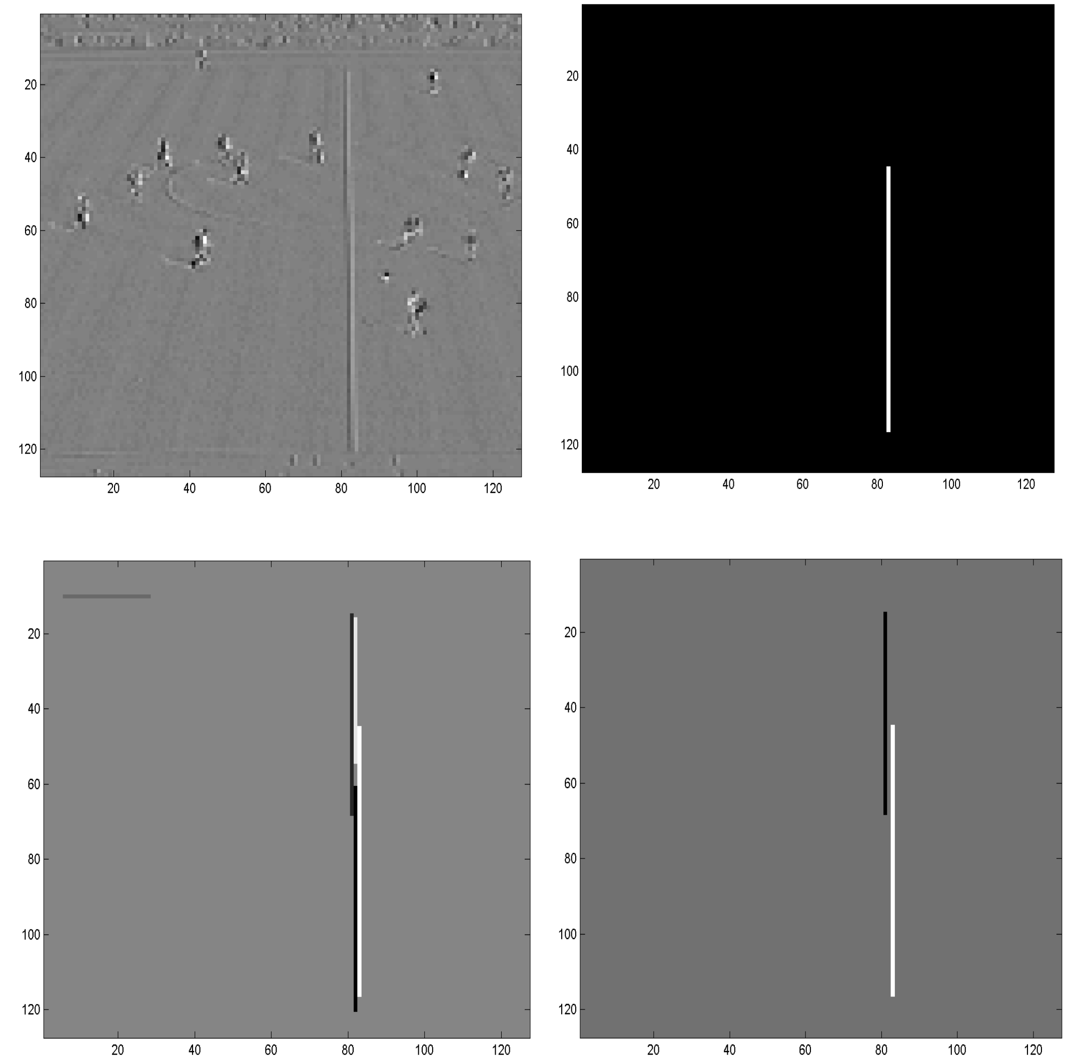
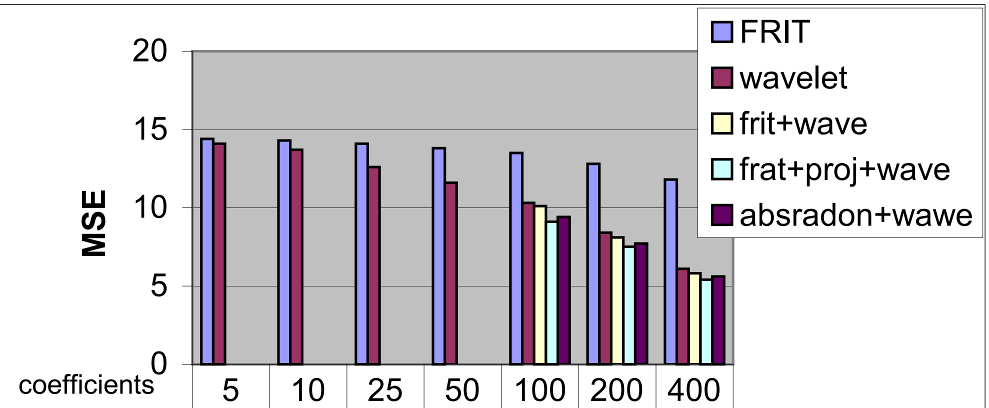


Fig.9: From high left, clockwise: original dfd, first detected line, first 2 detected lines, and first 5 detected lines

After $n$=5 iterations we performed a wavelet coding on the residual image that now is easier to compress because a lot of features have been eliminated by the previous stage.
For “absradon+wave” the coefficients are selected in this way: 3 lines, each one coded with 8 wavelet coefficients, so 24 in all. For the residual image, then 76, 176 and 376 2D-wavelet coefficients in order to have respectively 100, 200 and 400 coefficients.
Figure 10 shows the mse values obtained with all the presented techniques. In figure 11 the same block presented in figure 9 is reconstructed with 400 wavelet coefficients and 400 coefficients of the technique presented in sub-section 4.2.




| coefficients | 5 | 10 | 25 | 50 | 100 | 200 | 400 |
|---|---|---|---|---|---|---|---|
| FRIT | 14.4 | 14.3 | 14.1 | 13.8 | 13.5 | 12.8 | 11.8 |
| wavelet | 14.1 | 13.7 | 12.6 | 11.6 | 10.3 | 8.4 | 6.1 |
| frit+wave | | | | | 10.1 | 8.1 | 5.8 |
| frat+proj+wave | | | | | 9.1 | 7.5 | 5.4 |
| absradon+wawe | | | | | 9.4 | 7.7 | 5.6 |

Fig.10: MSE values for the Jleague dfd (127x127)

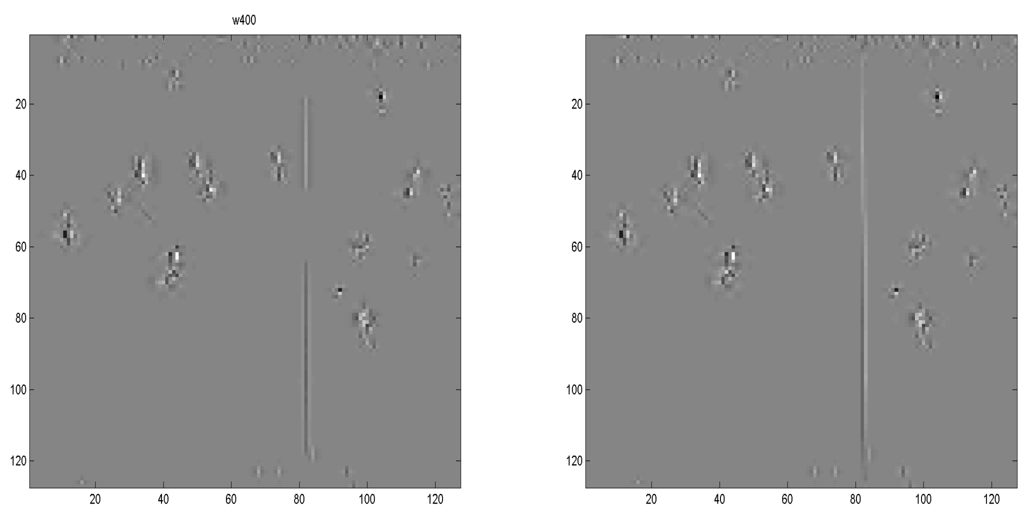



Fig.11: Block of J-league reconstructed with wavelet and "absradon+wave"

Table 2 illustrates the mse values in a difficult and particularly inconvenient case. The block is taken from the sequence "Akyio" and does not present any straight line at all. In these situation the last two procedure shows to be quite robust. Anyway here it appears evident that these algorithms have to be improved. In particular, since they are iterative, it will be interesting to investigate what could be the stopping criterion. In this way it is possible to find a method that uses the first procedure since there are lines that is convenient to code separately and then pass to the 2D-wavelet encoder.

| *Coefficients* | 2D-DWT | *Frit+wave* | *frat+proj+wave* | *absradon+wave* |
|---|---|---|---|---|
| 100 | 10.15 | 12.31 | 12.46 | 14.94 |
| 200 | 7.51 | 8.05 | 7.12 | 8.34 |
| 400 | 4.87 | 6.23 | 4.89 | 5.19 |

Table 2: MSE values for a difficult case

## 5 Conclusions

Ridgelet transform and finite ridgelet transform present very interesting characteristics especially because of their ability to catch straight lines that comes from the fact of being built on Radon transform. This property can be really useful as their first coefficients can capture the most important and visually relevant linear features, simplifying the coding process of the residual image. Following this idea the Radon transform can be seen as an important pre-processing instrument, whose effect is double: first, the most important lines are well identified and represented, second, he residual image entrusted to the second step is easier to code since it contains less information.

Here we propose two ways to exploit the good characteristics of Radon transform, focusing on the particular structure of motion compensated images. The algorithms are, of course, mainly adapted for images that present linear edges, but appear to be quite robust and can have not too bad performances even in unfavorable cases.
Procedures presented in section 4 may demonstrate their utility as a base layer for a 2D-wavelet based codec or for a Matching Pursuit one. If the codec used in the second stage is block-based the capacity of detecting lines in the whole frame can be especially useful.
A very important development of this work will be to understand the quota of the coding process that can be demanded to the pre-processing phase, quota that essentially depends on the nature of the image taken in account. This point turns out to be nothing else than finding a stopping criterion for the iterative algorithm, as explained in section 4.
Moreover, it could be interesting investigating the behavior of Curvelet transform [11] on motion compensated images.

## References

[1] E.J.Candès and D.L.Donoho, *"Ridgelets: a key to higher dimensional intermittency?"*, Phil.Trans.R.Soc.Lond.A. , pp.2495-2509, 1999

[2] M.N.Do and M.Vetterli, *"Finite Ridgelet Transform for Image Representation"*, IEEE Transactions on Image Processing

[3] M.N.Do and M.Vetterli, *"Orthonormal finite ridgelet transform for image compression"*, in Proc. IEEE Int. Conf. on Image Proc. (ICIP), 2000

[4] M.N.Do, P.L.Dragotti, R.Shukla, and M.Vetterli, "*On the compression of two dimensional piecewise smooth functions"*, in IEEE International Conference on Image Processing (ICIP), 2001

[5] R.M.Figueras i Ventura, L.Granai and P.Vandergheynst, *"R-D Analysis of Adaptive Edge Representations"*, submitted to IEEE 2002 International Workshop on Multimedia Signal Processing MMSP02

[6] O.K.Al-Shaykh, et al., *"Video Compression Using Matching Pursuit"*, IEEE Trans. Circuits Syst. Video Technol., February 1999. vol. 9(n.1): p. 123-143

[7] F.Moschetti, L.Granai, P.Vandergheynst and P.Frossard, *"New Dictionary and Fast Atom Searching Methods for Matching Pursuit Representation of Displaced Frame Difference"*, in Proceedings of IEEE ICIP, Rochester, NY, September 2002

[8] E.J.Candès (1999), *"Ridgelets and the Representation of Mutilated Sobolev Functions"*. SIAM J. Math. Anal.

[9] E.J.Candès *"Ridgelets and Their Derivatives: Representation of Images with Edges"*, *Curves and Surfaces*, L. L. Schumaker et al. (eds), 1999, Vanderbilt University Press, Nashville, TN

[10] F.Matus and J.Flusser, *"Image representation via a finite Radon transform"*, IEEE Trans. Pattern Anal. Machine Intell., vol. 15, no. 10, pp. 996–1006, Oct 1993.

[11] E.J.Candès and D.L.Donoho (1999). *"Curvelets - A Surprisingly Effective Nonadaptive Representation for Objects with Edges'*, Curves and Surfaces, L. L. Schumaker et al. (eds), Vanderbilt University Press, Nashville, TN.