LET IT WAVE

8/16 rue Paul Vaillant-Couturier – 92240 Malakoff – FRANCE

e.mail : info@letitwave.fr – **Web :** www.letitwave.com

Super-Resolution Bandlet Upconversion for HDTV

Stéphane Mallat

1 Upconversion is Highly Needed but Extremely Difficult

Let's face it, it is extremely difficult to *upconvert* Standard Definition (SD) television images into good quality High Definition (HD) images. It involves deep image processing, mathematical and real time processing problems. Yet, upconversion is now a central issue for broadcast and television industries. High Definition broadcast channel need to convert existing Standard Definition archives, films, advertising in an HD format to broadcast them over their HD channel, while maintaining high quality images. The proportion of SD upconverted programs is often over 80% during the first years of an HD broadcast channel. Moreover, HD flat LCD or Plasma screens are now invading the consumer market and these screens must display progressive HD images whatever the format of the incoming source, which is most often interlaced standard television images. The television set must therefore perform the upconversion from SD to HD before the image display. Given the important cost of these large screen televisions, consumers expect high quality images which they do not get from existing upconversion technologies embedded in televisions.

The tremendous challenge of upconversion is to compute missing pixels of HD images from pixels of input SD images, in order to produce HD video sequences that are sharp and detailed. This is called a "super-resolution" process because the resolution of images are apparently increased with an appropriate recombination of the information available. Another source of difficulty is the presence of distortions in the input SD images. These images may be contaminated by the camera noise. This noise is sometime attenuated by the camera electronics with a local averaging, which then introduces a blur. However, the worst distortions are often introduced by MPEG video compression. Because of bandwidth limitations, the quality of SD compressed images delivered in the homes is often barely acceptable. The last challenge of HD upconversion is to find a computational architecture that is fast enough to output 50 or 60 HD images per second. For an 1080p HD image format, this means outputting over 200 mega bytes of good quality HD color images per second from degraded SD input images. This is probably one of the most difficult image processing problem nowadays.

From a mathematical point of view, SD to HD upconversion is an *ill-defined inverse problem* of the worst kind. You must recover good quality missing pixels from low resolution input pixels that are degraded by complex distortions such as multiplicative camera noise, compression distortions and unknown blurring. Creative engineers and image processing researchers have come out with elegant ideas providing simple solutions that resulted in products available for the television industry. Yet, these solutions do not provide the image quality that consumers are expecting from HD images.

Existing technologies do not take advantage of the recent advances of applied mathematics to image processing. Efficient solutions of such a complex problem can come from a close interaction of high level mathematics, image processing and parallel hardware com-

putational architectures. Let It Wave's upconversion solution was developed in this spirit, and is implemented in a medium size Altera FPGA chip. This white paper describes the evolution of important ideas and techniques for upconversion, which gives the background to explain the principles of Let It Wave's algorithms.

Section 2 begins by describing in further details the different video formats and the resulting difficulties of upconversion. The evolution of solutions was first driven by the increase of computational power. Section 3 explains the resulting *motion adaptive* technologies currently used by the industry. Most engineers and researchers view *motion compensation* techniques as the future of upconversion. I will explain why I disagree, despite the considerable research and development efforts devoted to this approach. Avoiding unstable motion measurements, Section 3.3 describes a super-resolution procedure that computes missing pixels by minimizing a total variation norm. The resulting images are sharp with nearly no oscillation in space (jaggies) and in time (flickers).

Upconversion must also be robust to the distortions of the input SD source. Section 4 reviews traditional linear and adaptive filtering to remove noise and distortions. Wavelet thresholding algorithms provide efficient solutions to adaptive estimation which also restore sharp image structures. Section 4.3 explains how Let It Wave's bandlet technology improves wavelet denoising by taking advantage of spatial and time geometrical structures in videos.

2 Multiple Format Conversion

The number of video formats is increasing beyond reason, with various flavors corresponding to different image sizes (number of rows and columns), different time sampling rate (number of images per second) and different space-time sampling pattern (interlacing or not). Next section begins by explaining these different formats and their possible combinations, to understand the requirements of an upconversion process.

2.1 Upconversion of Interlaced Videos

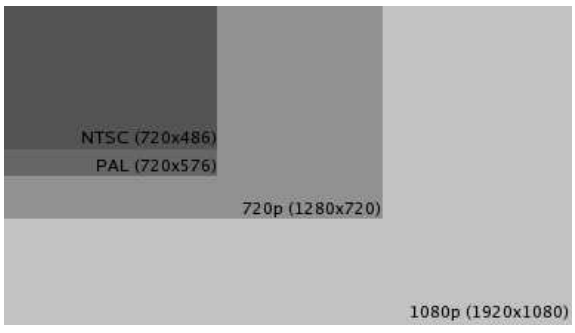

FIG. 1 – Sizes in rows x columns of Standard Definition and High Definition television formats

Let me begin with interlacing which is a major source of complexity. The European PAL and the American and Japanese NTSC standard television formats are interlaced. An interlaced video is a succession of 50 or 60 *fields* per second, where each field carries only half of the total image rows. One field at a time t includes only the even rows and the next field at $t + 1/50$ or $t + 1/60$ includes only the odd rows and so on. This is an elementary form of compression that takes advantage of the quick time response of the

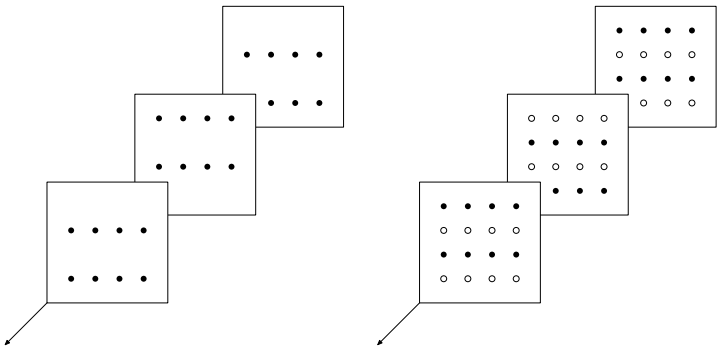

FIG. 2 – *Left :* Interlaced video with 3 fields including respectively even and odd pixel rows. *Right :* Deinterlaced sequence with calculated missing pixels shown as white circles.

CRT screen technology and the persistence of human vision. NTSC interlaced videos have 60 fields per second with 486 rows total (even and odd) and 720 columns. PAL/SECAM interlaced videos have 50 fields per second with 576 rows total (even and odd) and 720 columns. These images are displayed on a screen with a 4/3 ratio, with a sampling interval that is not equal along rows and columns.

Flat LCD and plasma screens mostly display progressive images. A progressive image, also called *frame*, includes all rows, the even and odd rows. High definition television progressive video sequence have 50 or 60 frames per second where as a progressive film format has 24 frames per second.

Deinterlacing is a process that computes the missing even or odd rows of each field of an input interlaced video to output a progressive video sequence, as illustrated in Figure 2. Deinterlacing is necessarily included in all flat screen televisions. Getting high quality images requires a super-resolution process that computes missing pixels from available ones, with the best possible resolution.

Besides deinterlacing, a *scaling* is necessary to increase the image size from SD to HD formats. HD images have a 16/9 ratio, with 720 rows and 1280 columns for the 720p format and 1080 rows and 1920 columns for the 1080p format, as shown by Figure 1. If the scaling adjusts the number of rows, since the 4/3 ratio of SD images is smaller than the 16/9 ratio of HD images, the scaling does not produce enough columns. Missing columns are then shown as black vertical bands (pillar box). One can also adjust the scaling factor to match the number of columns in which case there are too many rows. Top and bottom image rows then do not appear in the 16/9 format.

2.2 Video/Film Cadence

Interlaced videos often have a more complex time structure than the succession of even and odd fields previously described. This happens when films are converted into interlaced videos or when computer graphics elements such as crawling text or logos are inserted in images. This creates complex *cadences* that must be taken into account by the deinterlacing process.

A digital film is a progressive video with 24 frames per second that may be converted in an NTSC format with 60 fields per second, for television broadcast. Each frame is first divided in two fields that respectively carry the even and odd rows, which results in 48 fields per second. To obtain 60 fields per second, an extra field is added every 4 fields. Two successive frames (A : B) in the original film are thus first converted in 4 fields (A-odd, A-even : B-odd, B-even) and a new field is added to output (A-even, A-odd, A-even : B-

odd, B-even) which is called a $3:2$ cadence because it includes 3 fields of 1 frame followed by 2 fields of the next one. The same pattern then repeats. In this $3:2$ cadence, the first 3 fields correspond to rows of the same original frame (same time) and the next two also correspond to the same frame, whereas for an interlaced video, each field gives the even or odd rows of a different frame at a different time. The deinterlacing must to take into account these properties when computing the missing rows of each field.

Other cadences than $3:2$ may be created by other post-processing. To accelerate a film and fit it with a time lot, one technique is to drop 1 every 12 fields, which is barely noticed by a viewer. The resulting cadence is $3:2:3:2:2$ which means that like in a $3:2$ cadence there are 3 fields of the first frame then 2 of the next then 3 of the following one then again 2 and afterward we drop one field and hence instead of 3 there are 2 fields. This pattern is repeated afterward. With the diversification of video sources such as camcorders of animation fields, the variety of cadences gets more and more complex. Moreover bad editing of videos may modify periodic cadences.

Another source of complexity comes from post-processing that can mix video materials corresponding to different cadences. A crawling text or a portion of film can be inserted in the fields of an interlaced video. As a result, some pixels correspond to an interlaced video and some other to a converted film that may have a $3:2$ or a more complex cadence. Taking these edits into account, requires a *per pixel cadence detection.* which computes a potentially different cadence for each pixel. Finding stable estimates of per pixel cadence parameters is yet another challenge of video conversion.

2.3 Real-Time Parallel Computations

A fundamental specificity of video over fixed image processing is the real-time computational constraint. It often drives the elaboration of video processing algorithms. One may think that it is just a matter of hardware implementation once the algorithms are well optimized. It is not the case because computational complexity issues are not the same for a von Neumann architecture of a sequential computer and a parallel implementation in a hardware chip such as an FPGA or an ASIC.

For computer scientists, the complexity of an algorithm such as a Fast Fourier Transform or a Fast Wavelet Transform is typically the number of additions and multiplications and the memory size for the storage of intermediate calculations. For real-time video processing, the data rate is huge, over 200 mega bytes/second for 1080p HDTV. As a result, memory bandwidth is often more crucial than the number of arithmetic operations.

To accelerate standard sequential algorithms, one can use an interconnected array of processors that operate in parallel. A sequential algorithm is divided it into multiple parallel sequential computations that communicate their intermediate results. This strategy simplifies the algorithmic effort to accelerate a sequential algorithm but the hardware is not used efficiently. As a result, advanced upconversion algorithms that are implemented on programmable parallel array architectures require large size chips having an important power consumption.

Hardware processing can be viewed as a flow of particles that move, communicate in parallel, aggregate the results and continue their multiple paths. For complex problems such as real-time video upconversion, it is important to take advantage of the full flexibility offered by parallel calculations. Our experience at Let It Wave is that exploring the flexibility of parallel computations has transformed the nature of our algorithms. This is how we were able to implement sophisticated non-linear real-time HD video processing on a single medium size FPGA such as an Altera Cyclone II-70, with a low power consumption.

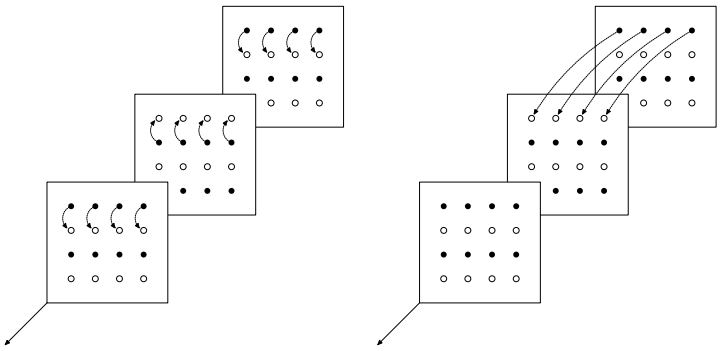

FIG. 3 – *Left :* A line doubling computes missing rows by reproducing (doubling) the even and odd rows. *Right :* A time weaving gets each missing from the corresponding row of the previous field.

3 Deinterlacing and Scaling

3.1 Motion Adaptive Deinterlacing

Although deinterlacing and scaling can be viewed as a single super-resolution process that compute missing pixels, they are often calculated in two separate stages for flexibility and computational efficiency reasons. The output image quality depends mostly upon the efficiency of deinterlacing, which is the most difficult process. I will thus concentrate on deinterlacing before considering scaling. Problems related to film and complex cadences will be described afterward.

Deinterlacing was already studied in the 1970's and there is a large body of elegant ideas and solutions with their limitations [2]. In the 1980's and 1990's, the first challenge of deinterlacing was real-time digital video processing at a minimum cost. Minimizing the number of operations and memory requirements was thus necessary, which first lead to two simple methods : spatial line doubling or time weaving.

Spatial line doubling simply copies each even or odd row of a field in respectively the next odd or previous even row, to recover missing rows, as shown by Figure 3 *Left*. This gives a good result wherever the image intensity varies smoothly, but for sharp transitions it reduces the vertical resolution and produces artifacts such as *jaggies* along diagonal edges. These jaggies appear in the examples of deinterlaced images shown in the *Upper Left* of Figures 6, 7, 8, 9.

Time weaving is another simple deinterlacing technique which copies the even rows of an image field into the missing even rows of the next field and the odd rows of a field in the missing odd rows of the next field. This is illustrated by Figure 3 *Right*. Time weaving gives a perfect result if nothing moves in the video. In presence of motion, a time weaving mixes odd and even rows that are shifted, which produces "comb" or "mouth teeth" artifacts. This appears in the examples of deinterlaced images shown in the *Upper Middle* of Figures 6, 7, 8, 9.

With more processing available these basic spatial line doubling and time weaving techniques have been replaced by linear spatial interpolations and linear time interpolations. The copy of a spatial line doubling is then replaced by an average between the top and bottom pixel as illustrated by Figure 4 *Left*. The jaggy artifacts in the *Lower Left* of Figures 6, 7, 8, 9, are reduced but remain very strong. This 2-tap linear interpolation can be replaced by an interpolation using more spatial neighbors, but it does not improve the result. Since one row out of two is missing, the images are spatially undersampled with

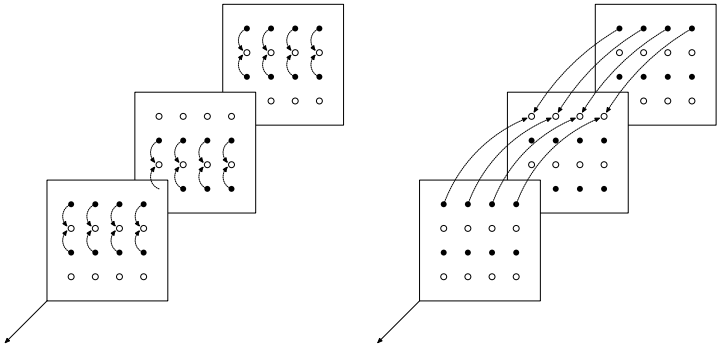

FIG. 4 – *Left :* Missing pixel are computed by averaging top and bottom pixels. *Right :* Missing pixel are computed by averaging same position pixels before and after.

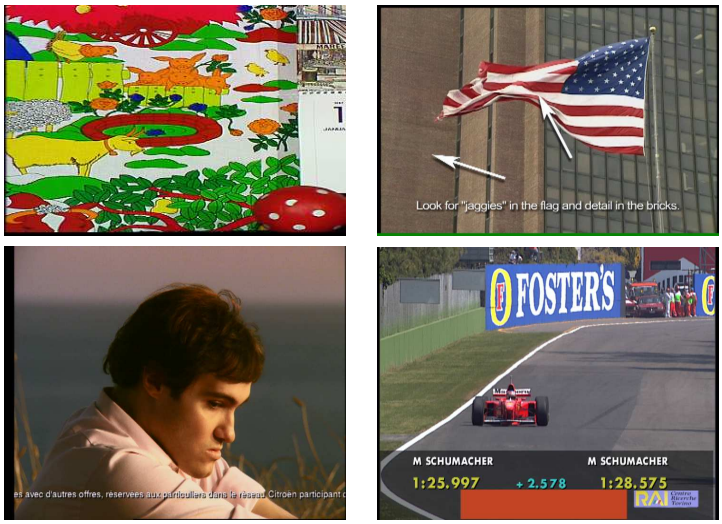

FIG. 5 – Fields of interlaced SD videos.

respect to their frequency content. Interpolating such signals produces all types of *aliasing* artifacts including jaggies and *Moiré effects* on periodic patterns.

Similarly, the time copy of a time weaving can be replaced by a time linear interpolation that computes each missing pixel as an average of the next and previous pixel at the same location. This is illustrated by Figure 4 *Right*. It improves the quality of time-weaving deinterlacing but comb artefacts remains as shown in the *Lower Middle* of Figures 6, 7, 8, 9. This is also due to an aliasing phenomenon in time. For fast motions, the time sampling rate is smaller than the maximum time frequency content of the video. As a consequence, any time linear interpolation produces aliasing artifacts.

Time interpolations gives good results when there is no or little motion even along sharp spatial transitions whereas spatial interpolations give good results in smooth spatial regions even when there are fast motions. A natural improvement of these techniques is to mix them to get the best of both, which corresponds to *motion adaptive* algorithms. A motion detector finds if there is a strong movement by calculating the energy of the differences of odd rows in time and of even rows in time. If the movement is "strong" then a missing pixel is calculated with a spatial interpolation otherwise it is calculated with a time interpolation. For quick movements of sharp structures such as edges or textures, this technique uses a spatial interpolation which reduces the spatial resolution and produces

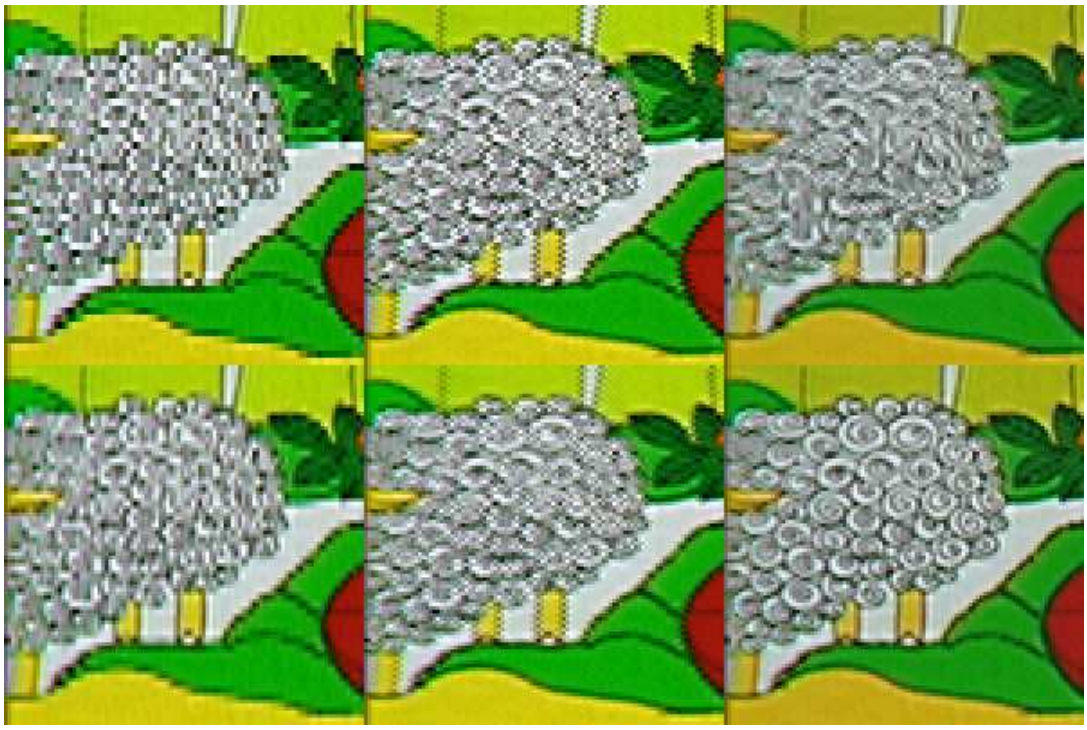

FIG. 6 – Deinterlaced zoom on upper left of Figure 5. *Upper left :* line doubling. *Lower left :* spatial interpolation. *Upper middle :* time weaving. *Lower middle :* time interpolation. *Upper right :* motion adaptive. *Lower right :* Let It Wave's deinterlacing.

aliasing artifacts such as jaggies. The improvement and typical remaining artifacts are shown in the *Upper Right* of Figures 6, 7, 8, 9.

Up to this point, innovations came essentially from hardware architecture allowing more computational power rather than creative new algorithmic ideas. To reduce artifacts of spatial interpolations an important algorithmic innovation came from *edge adaptive interpolations*, among which the Directional Correlation De-interlacer (DCDi) algorithm [6] that locally adapts spatial interpolations to the directions of local image structures. If there exists a spatial direction along which the signal is locally regular then a precise estimate of the missing pixel value is obtained with an interpolation along this direction. In the neighborhood of an edge, the interpolation should be performed parallel to the edge and not across the edge. The main difficulty is then to estimate a direction in which the signal has smooth variations in the neighborhood of a missing pixel. Many possible criteria may be used among which correlation measurements. Such adaptive directional interpolators reduce the artifacts introduced by fixed spatial interpolators but not completely because not enough data is available in a single field to perform a precise directional interpolation of missing pixels. As a result, deinterlaced video can have a *time flicker* (oscillatory artifacts), when the directional interpolations performed on even and odd fields do not give coherent information. However, this improvement is at the core of nearly all *motion adaptive* deinterlacing procedures used by the television industry.

Image artifacts have two negative impacts : they produce visible errors that degrade the image quality and they increase the cost of compressing these images with MPEG. Indeed, artifacts are irregular structures that require many bits to be coded. For broadcast channels that upconvert their SD content in HD before compressing the video, the bit rate

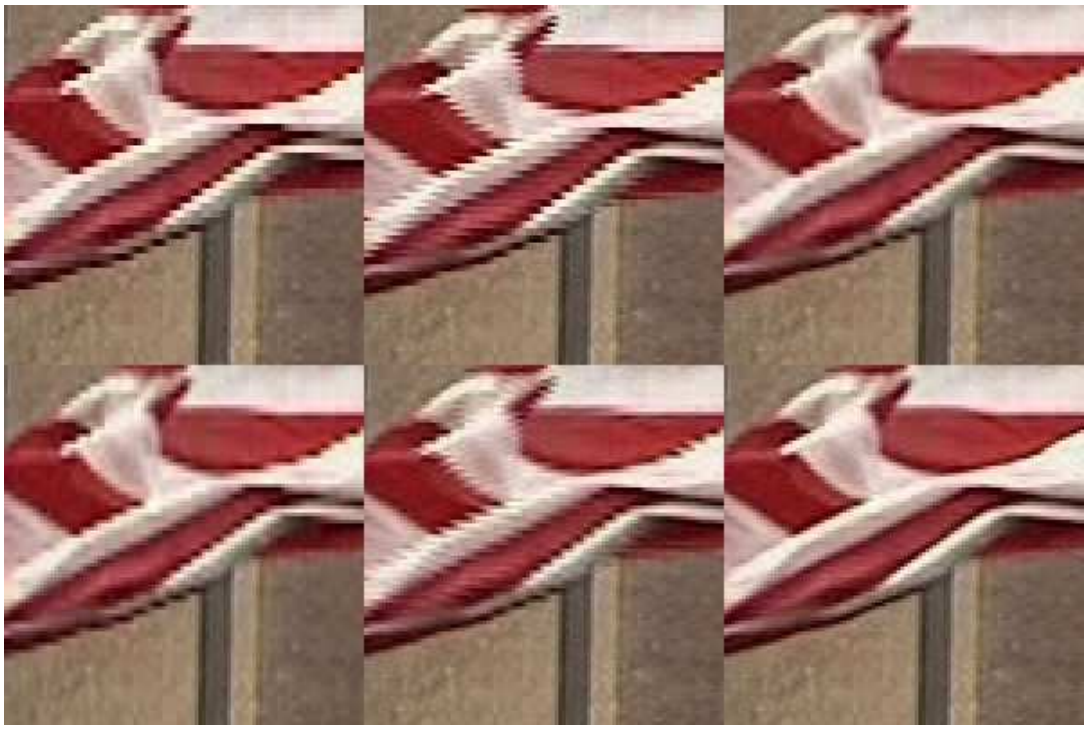

FIG. 7 – Deinterlaced zoom on upper right of Figure 5. *Upper left :* line doubling. *Lower left :* spatial interpolation. *Upper middle :* time weaving. *Lower middle :* time interpolation. *Upper right :* motion adaptive. *Lower right :* Let It Wave's deinterlacing.

FIG. 8 – Deinterlaced zoom on lower left of Figure 5. *Upper left :* line doubling. *Lower left :* spatial interpolation. *Upper middle :* time weaving. *Lower middle :* time interpolation. *Upper right :* motion adaptive. *Lower right :* Let It Wave's deinterlacing.

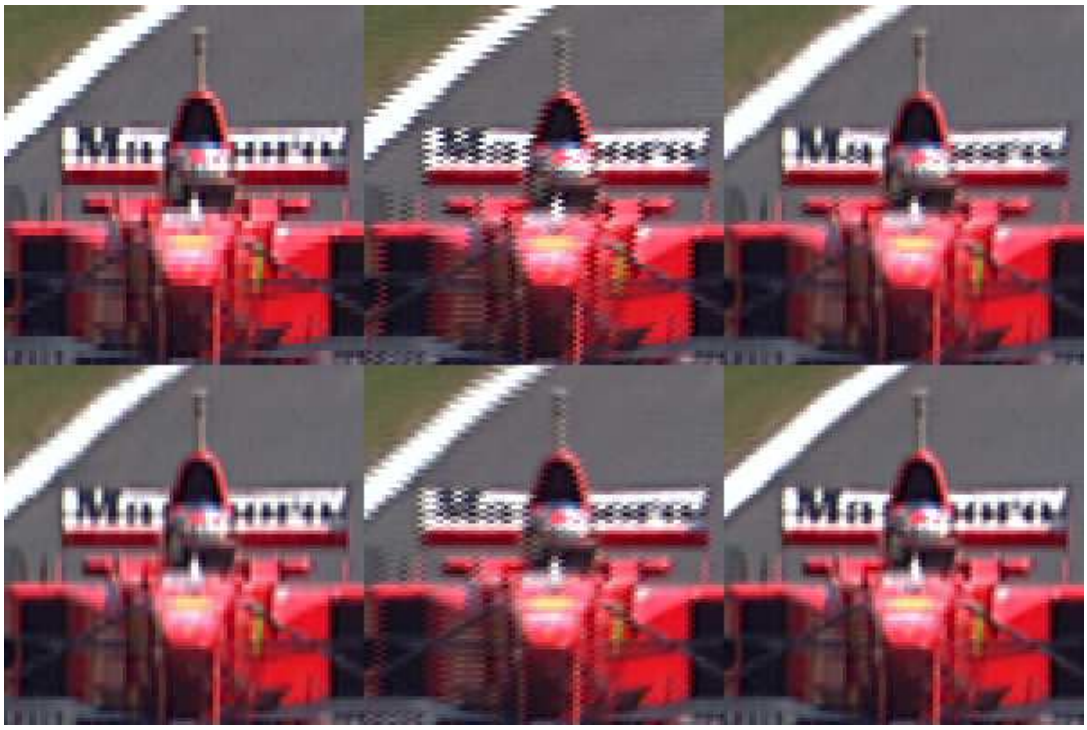

FIG. 9 – Deinterlaced zoom on lower right of Figure 5. *Upper left :* line doubling. *Lower left :* spatial interpolation. *Upper middle :* time weaving. *Lower middle :* time interpolation. *Upper right :* motion adaptive. *Lower right :* Let It Wave's deinterlacing.

is fixed and deinterlacing artifacts are therefore amplified by MPEG image compression. To reduce flickering, upconverters often include an averaging in time, which blurs images and reduces their resolution.

Scaling After deinterlacing a sequence of fields is converted into a sequence of full frames that include both even and odd rows. For HD upconversion, Section 2.1 explains that a scaling is needed to adjust the number of rows and columns. Upconversion and downconversion correspond respectively to factors that are larger and smaller than 1. After an appropriate deinterlacing, the sampling density is not quite sufficient to reproduce the highest image frequencies, but optimized linear filters produce relatively little aliasing compared to the result obtained when applying these filters for deinterlacing. Linear filters are typically implemented with separable convolutions along the deinterlaced frame rows and columns. For downconversion, a linear scaling is perfectly appropriate. For upconverting, a linear scaling produces a blurred image when the scaling factors are above 1.5.

3.2 Motion Compensated Deinterlacing

Motion adaptive techniques are not sophisticated enough when sharp image structures move. The industry and image processing research community often views *motion compensation* as the next generation technology [2]. The idea is indeed simple and appealing. First we compute the motion of each pixel and then for missing pixels we perform the interpolation in time in a direction that follows the time displacements. The interpolation is thus performed in a time direction that compensates for the motion. Developing a robust "mo-

tion compensation" deinterlacing technology was the Graal search of the video conversion industry during the last 10 years. Nearly all companies had or have engineers developing motion compensation deinterlacing algorithms. This was encouraged by thousands of research papers with various flavors of motion compensation algorithms. An Internet search on "motion compensation conversion" brings over 3 million hits ! Spectacular demonstrations are available with sophisticated techniques. Yet after all this time and effort, no motion compensation algorithm product is available with the expected quality improvement with respect to motion adaptive techniques. Motion compensation techniques do not seem to be sufficiently robust, and when they fail they produce artifacts that are more visible than motion adaptive artifacts. At a deeper level, I believe that motion is the wrong concept to understand complex image transformations in time for deinterlacing and upconversion in general.

Motion compensation is based on the assumptions that a single motion can be associated to each image pixel, and that this motion can be computed reliably. Since the 1980's, computing the motion of image pixels, also called *optical flow*, is a major image processing research topic, with tens of thousands of papers published. Optical flow research has shown that unicity and robustness assumptions are wrong. First, one cannot always associate a single motion to a pixel. For example, at an occlusion boundary between an object that moves over a background, pixels have two motions : the object motion and the background motion. In presence of transparencies, several motions are also associated to a single pixel. They correspond to the motions of the superposed structures that are being visualized. It may be a smoke over a background or a scene viewed across a window. Second, optical flow estimation is intrinsically unreliable and is known as an "ill posed problem" [13]. It requires making regularity assumption on the flow, for example that it is locally regular over a spatial neighborhood. Even the human visual system "regularizes" motion estimation, which explains the "mistakes" that are revealed by optical illusions. It does not mean that motion can never be measured accurately but that it cannot be always measured accurately.

For video compression, one may then wonder why motion compensation techniques are highly efficient and used by MPEG standards. When the motion is accurate it reduces the bit rate and when it is wrong it increases the bit rate. On average the balance is positive and motion compensation brings an important improvement to video compression. For deinterlacing, if a wrong motion is used, a wrong pixel value is calculated by the interpolation, which degrades the image quality. Pixels of different colors may be introduced in uniform color regions that are moving, which is highly visible. As a consequence, motion compensation algorithms measure the reliability of their motion calculation and when not sufficiently reliable a spatial interpolation is performed. To obtain robust results requires to be very conservative and hence perform many spatial interpolations. It thus often leads to the same type of artifacts as motion adaptive algorithms, with much more operations. Given the creativity of engineers and researchers working in this area, I am sure that motion compensation will end up being more efficient than motion adaptive algorithms, but at a considerable computational cost. I thus believe that it is not the best approach.

3.3 Let It Wave's Super-Resolution with Total Variation

Motion adaptive algorithms have a limited performance because a missing pixel is computed from a small set of available pixels, either in the same field or at the same position in the the next or previous fields. Motion compensated algorithms have the advantage of using information in a full three-dimensional space-time neighborhood but are less robust because of motion estimation errors. Let It Wave's procedure replaces motion calculations

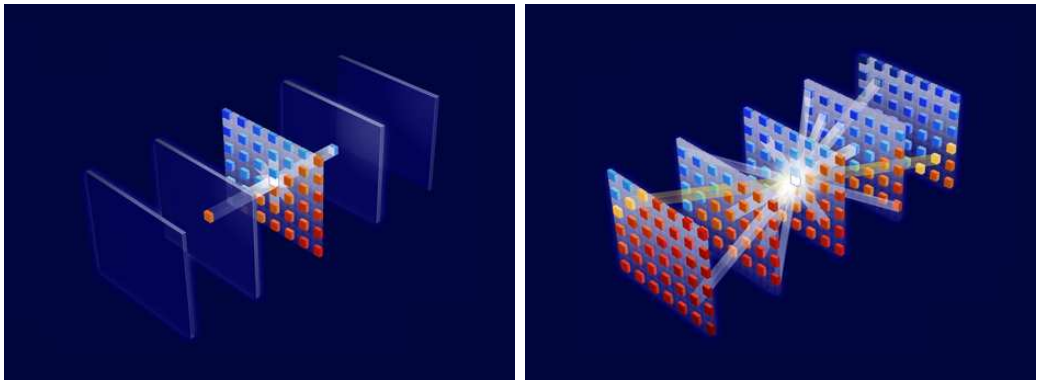

FIG. 10 – *Left :* Motion adaptive algorithms computes a missing pixel with a spatial interpolation or a time interpolation from the previous and next pixels of same location. *Right :* Let It Wave's super-resolution searches over a full space-time neighborhood for a directional interpolation that minimizes the resulting total variation.

by a minimization of a total variation norm while taking advantage of available pixel values in a whole space-time neighborhood.

Missing image pixels can be computed by minimizing a global regularity criteria on the resulting image. Classical Tykonov regularizations compute an image that is as smooth as possible, by minimizing a Sobolev norm, which measures the energy of the image gradient. The reconstructed image is maximally regular and hence blurred. In the early 1990's, it was observed that Sobolev norm are not appropriate to model the regularity of sharp images, because they typically have discontinuous edges. As a consequence their derivatives do not have a finite energy. A total variation norm is more appropriate because it measures the amplitude of oscillations and variations of the image, without penalizing discontinuities [12]. Formally, the total variation is the integral of the absolute value of the modulus of the gradient of an image, as opposed to the squared modulus which gives a Sobolev norm. One can prove that the total variation of an image is essentially proportional to the total length of edges, whether these edges are discontinuous or smooth transitions. Computing images of minimum total variation under constraints can be obtained with an iterative non-linear diffusion [8]. For two-dimensional images, impressive super-resolution interpolations results have since then been obtained by minimizing this total variation or a similar criteria. For video super-resolution, motion compensation algorithms with a two-dimensional total variation minimization have been proposed and can produce very good results [5], but they suffer from motion computation instabilities previously discussed.

Jaggies and flickers are spatio-temporal oscillatory artifacts. To reconstruct sharp images without such artifacts, one may think of minimizing a three-dimensional total variation norm. However, total variation minimizations are iterative algorithms that are computationally expensive, because they produces long range interactions between image pixels. For fast parallel calculations, missing pixels must therefore be computed with a localized total variation minimization.

Suppose that the value of a missing pixel is estimated with several directional interpolations from available pixels in a three-dimensional space-time neighborhood. The best interpolation may be defined as the one that minimizes the resulting total variation norm. This process can also be interpreted as a maximum likelihood estimation. Each directional interpolation estimates the missing pixel value from the available data and the best esti-

mation is defined as the one that minimizes a negative log likelihood defined from a local total variation norm.

A directional interpolation computes a missing pixel by interpolating one or more available pixels located close to a line going through the missing pixel with a particular spatio-temporal direction, as illustrated by Figure 10 *Right*. Any directional interpolation may be used a priori. An order 0 interpolation is a simple copy of one pixel located before or after the missing pixel along the spatio-temporal direction. Line doubling and time weaving algorithms are such copies respectively along the vertical spatial direction and along the time direction. A first order interpolation may be computed between two pixels along the specified spatio-temporal direction, or a higher order polynomial interpolation may use more than two pixels along this direction. Different interpolations produce different estimations. The set of all possible estimations is therefore the result of a choice of a family of spatio-temporal directions and a set of different interpolations along these directions.

The best directional interpolation is computed by minimizing a total variation norm that measures the oscillations introduced by the interpolated missing pixel in its three-dimensional neighborhood. To incorporate the interaction between neighborhood missing pixel values, the minimization is not performed on a single missing pixel but over a neighborhood of missing pixels. The total variation minimization is thus regularized locally.

When the search for a best directional interpolation is limited to a spatio-temporal neighborhood that is too small, the best estimator may not be ideal. A spatio-temporal bandlet thresholding regularizes the estimation with a spatio-temporal geometric flow that is derived from the directions of the computed best interpolators, as explained in Section 4.3.

Scaling Upconverting SD to HD includes a deinterlacing and scaling. Both can be performed together since it amounts to finding a grid of missing pixels, but it is computationally more efficient to perform the deinterlacing first and then compute a scaling. After deinterlacing, a linear scaling with interpolation filters produces few artifacts but sharper images can be obtained with a non-linear scaling. As previously explained, the deinterlacing optimizes for each missing pixel the spatio-temporal interpolation. In its neighborhood, the scaling can be computed with an appropriate modification of this spatio-temporal interpolator. This directional interpolation avoids blurring across sharp transitions and producing oscillatory artifacts.

Per Pixel Cadence Section 2.2 explains that standard television videos may be a mix of interlaced videos and films converted to videos by dividing its frames in even and odd fields. Instead of trying to identify complex cadences, that may depend upon the pixel when there is an insertion of films or computer graphics, we search directly for the appropriate deinterlacing solution. A film frame is divided into two or three fields corresponding to even or odd rows. The deinterlacing should replace each field by the corresponding full frame. Let us consider a field where locally the pixels correspond to the even rows of a film frame. The odd rows are either in the field just before or just after. The missing pixels (odd rows) are thus obtained with a copy from the previous or next frame. This copy is a particular spatio-temporal interpolation of order 0, along time. The deinterlacing of films in video thus requires identifying an appropriate space-time interpolation. From our point of view, this is not different from deinterlacing any other interlaced video content. It is thus performed by minimizing the same type of local total variation measure, which finds the appropriate spatio-temporal interpolation.

4 Camera Noise, Compression Distortions and Blur

Deinterlacing and scaling for upconversion requires taking into account the degradations introduced by the camera noises, compression distortions and blurs. Camera noises can be modeled as a mix of additive and multiplicative Gaussian and Poisson noises. Some camera reduce this noise with a spatio-temporal averaging that introduces a slight blur. More blur may be introduced by out-of-focus or poor optics. However, the worst degradations often come from MPEG 2 and MPEG 4 compression. Upconverting such videos while removing the distortions, without adding worst artifacts, are difficult problems. Next section briefly reviews the state of the art linear and adaptive filtering techniques. The following sections explains the improvements obtained by wavelet and Let It Wave's bandelet thresholding algorithms.

4.1 Linear and Adaptive Filtering

Noise removal is a huge research area in statistical signal processing. Up to now, HDTV upconverters have mostly taken advantage of standard linear and adaptive filtering techniques. This section briefly reviews the state of the art.

Linear Wiener filtering is a basic but powerful signal processing tool to remove noise. The noise produces random gray level fluctuations that are typically more irregular than the original image content. It is thus attenuated by averaging the noisy image with a predefined linear filter, which is optimized depending upon global noise and image statistics. For video image sequences, the averaging can be performed in space and time to take advantage of spatial and time redundancy. The noise is attenuated but the averaging blurs sharp images structures such as edges.

Figure 11 shows an example of white noise removal on a fixed image. The image in Figure 11(c) was obtained with a linear spatial filter. The trade-off resulting from the filter optimization leaves some lower frequency noise in the regular regions such as Lena's shoulder and the removal of high frequencies produces a slight blur. Other simple non-linear filters such as median filters preserve better the edges but also blur irregular textures and do not remove as well the noise in regular regions.

Adaptive filtering techniques have been introduced to locally adjust the averaging in the neighborhood of each pixel. The idea is simple : the averaging should be extensive only where the original image has smooth variations. This averaging removes the noise fluctuations without degrading the image information since it varies smoothly. However, the averaging should be reduced and even removed at pixels near edges or in irregular texture regions. Despite important research in statistics and signal processing over this approach, the resulting algorithms are often ad-hoc with instabilities or heavy computational requirements. Indeed, finding if the original image is locally smooth is a difficult estimation problem in presence of noise. Adapting the filtering to this estimation is yet another difficult issue.

In time, linear filtering procedures are often based on autoregressive filters of order 1 or 2. These recursive filters have the advantage of being computationally efficient and causal, which introduces no computational delay. Adaptive recursive filterings modify the autoregressive parameters according to the difference of values between successive video frames. Despite some good results, this adaptivity is often ad-hoc and it is often necessary to limit the adaptivity to few frames. This means that important distortions are not removed. Moreover, there is no good mathematical framework to adjust simultaneously the spatial and time adaptivity.

4.2 Wavelet Thresholding

An important improvement over adaptive linear filtering came from mathematical statistics and harmonic analysis, through wavelet thresholding. The starting point of wavelet techniques is completely different from adaptive filtering. First you try to find a basis in which the image is represented mostly by nearly zero coefficients and few large amplitude coefficients that concentrate the image energy. If the image is contaminated by a noise, this noise typically adds a small amplitude component which is distributed over many coefficients. The noise can thus be suppressed with a thresholding that sets to zero all coefficients below a threshold value. This threshold can be set to be the expected maximum amplitude of the noise coefficients. All coefficients above this threshold are kept as is.

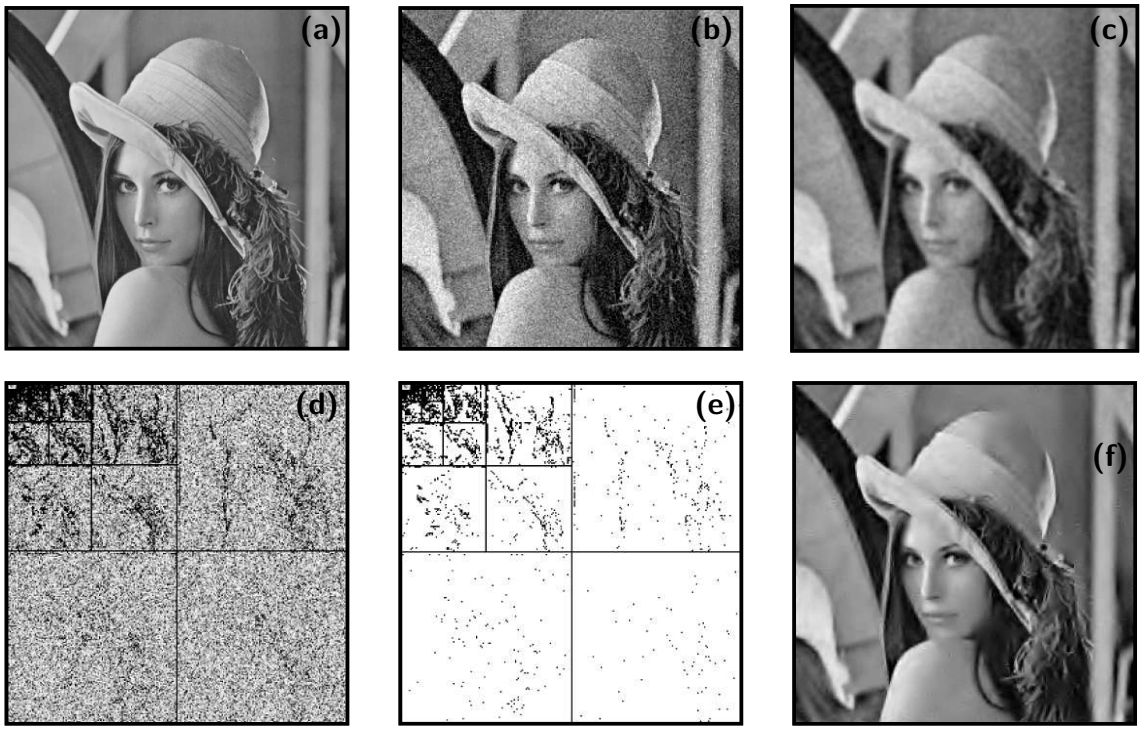

FIG. 11 – *(a) : Original image. (b) : Image contaminated by a white noise. (c) : Noise removal with a linear filter. (d) : Wavelet coefficient of the noisy image. (e) : Black points correspond to wavelet coefficients above a threshold. (f) : Image reconstructed from thresholded wavelet coefficients.*

Wavelets are bases introduced in harmonic analysis [11], which have close connections with filter bank algorithms [10]. Wavelet coefficients measure local image variations at different scales and locations and are computed with a fast algorithm that requires less operations than a fast Fourier transform [9]. Wavelet coefficients are nearly zero where the image is regular and have a large amplitude near edges and in irregular texture regions. Figure 11(d) shows the wavelet coefficients of a noisy image. Setting to zero wavelet coefficients with a thresholding removes images fluctuations at multiple scales and is thus equivalent to averaging the image at a scale that is locally adapted to the image content. It is proved that wavelet thresholding algorithms have optimal adaptivity properties for large classes of images [4, 10] including edges. Wavelet coefficients above threshold in Figure 11(e) correspond to sharp textures and edges. In regular regions, wavelet coefficients

are mostly dominated by the noise and are set to zero. The image reconstructed in Figure 11(f) from thresholded wavelet coefficients is sharp while the noise is nearly completely removed. Setting wavelet coefficients to zero in regular zones makes an extensive averaging, and sharp structures corresponding to large wavelet coefficients are well preserved.

When the noise has a large amplitude, setting to zero some coefficients close to an edge can create small oscillations such as Gibbs phenomena. More recently, number of researchers realized that many artifacts introduced by wavelets come from their inability to adapt to the directions of geometrical image structures. Wavelets have square support of various sizes which does not capture the directional regularity of edges and textures. One can improve wavelet representations with basis functions that are elongated in the direction of edges. This research lead to new constructions among which curvlets [1] and other geometrical representations [3].

4.3 Let It Wave's Bandlet Noise Removal and Quality Enhancement

Bandlets bases are geometrical wavelet bases that are adapted to capture and restore the geometrical image regularity where it exists. They are the result of a research carried in the applied mathematics department at Ecole Polytechnique, Paris [7]. Let It Wave further developed and patented the resulting procedures, and industrializes image processing products based on this technology.

Spatial Bandlets Bandlet coefficients are constructed over wavelet coefficients from a geometric flow that indicates the local direction of image structures [7], as illustrated in Figure 12. This geometrical flow is the spatial equivalent of an optical flow in time. A geometrical flow points in the direction in which the gray level image values "moves" in space. Along an edge, the geometrical flow is typically parallel to the edge. Bandlet coefficients are calculated with orthogonal transformations of wavelet coefficients along the flow. Bandlets are multiscale elongated functions that oscillates along a band that is parallel to the geometric flow. The energy of wavelet coefficients is concentrated over fewer bandlet coefficients.

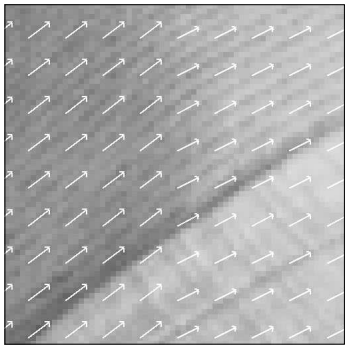

FIG. 12 – *Example of geometric flow computed on a textured region*

For noisy images, thresholding bandlet coefficients performs a multiscale adaptive averaging along the geometric flow. It regularizes image values along edges but not across edges, which restores their sharpness and geometric regularity. When the image is noisy, the flow can be obtained through a penalized estimation procedure that optimizes the resulting bandelet basis for image denoising, as explained in [7]. Figure 13 compares results obtained with a wavelet thresholding and a bandelet thresholding. Bandlets reproduce an

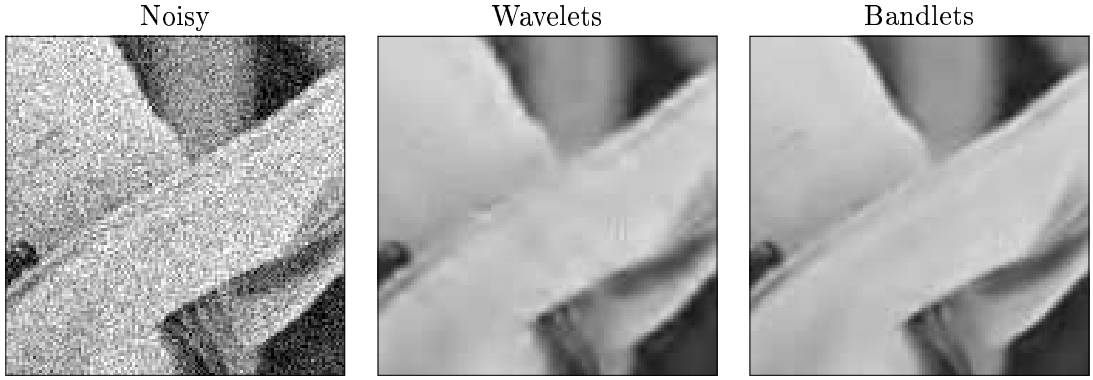

FIG. 13 – *Left :* zoom on Lena's hat with high noise. *Middle :* denoising by wavelet thresholding. *Right :* denoising by bandelet thresholding.

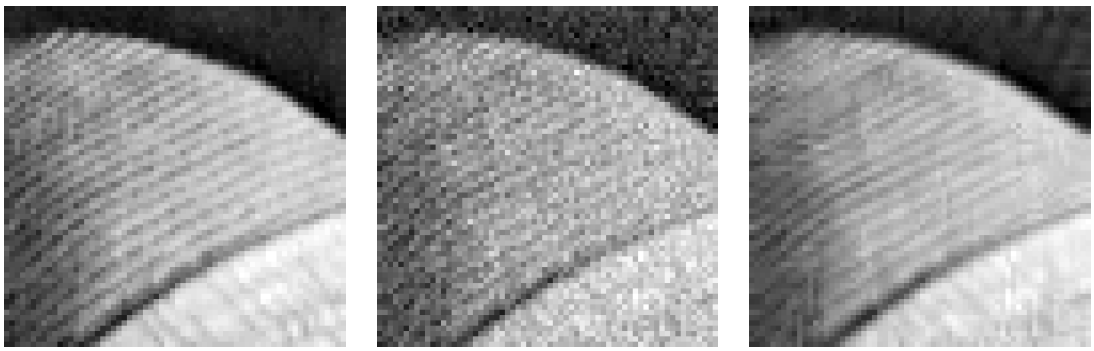

FIG. 14 – *Left : zoom on Lena's hat texture. Middle : noisy image. Right : denoising by bandelet thresholding.*

image whose geometry is better restored and most Gibbs type oscillations have been removed. Figure 14 gives another example of denoising with bandelets, which is particularly difficult because the texture to be restored has high frequency oscillations. Such textures are typically destroyed by any averaging that does not adapt to the texture direction. A bandelet thresholding reproduces such fine textures because it does not perform any averaging in the oscillatory direction. These spatial bandelet filtering techniques are used by Let It Wave to improve the quality of satellite and seismic imaging.

Spatio-Temporal Bandlets To remove noises and distortions from videos we can also use their time geometry corresponding to movements. The geometric flow then becomes a space-time flow that indicates the regularity directions in time and space. For computational efficiency, to regularize videos that have been deinterlaced, the geometric flow can be derived from the spatio-temporal directions of the interpolators that have been optimized to compute missing pixels.

Three dimensional bandlets look like elongated snakes in time, which follow the computed spatio-temporal directions. Thresholding such bandlet coefficients performs a simultaneous regularization in space and time whenever the video has no sharp transitions along the space-time flow. The threshold value is a parameter that is typically proportional to the estimated standard deviation of the noise. This is a powerful technique to suppress camera noise. Figure 15 *Left* shows an upconverted noisy frame computed from a noisy interlaced video. Figure 15 *Right* shows the resulting noise removal with an upconversion

FIG. 15 – *Left :* Upconverted frame computed from an interlaced video corrupted by a camera noise. *Right :* Upconversion including a bandlet thresholding for noise removal.

including a bandelet thresholding.

When the telecommunication bandwidth is not sufficient, MPEG 2 and MPEG 4 compression standard introduce "blocking artifacts" and "mosquito noise" that are highly visible. The square blocks that appear in compressed images result from the block calculations of the discrete cosine transform used by MPEG. Mosquito noise are random fluctuations around edges, resulting from the Gibbs oscillations introduced by the quantization of discrete cosine coefficients. These errors are highly non stationary, with varying spatial and temporal correlations. Thresholding space-time bandelet coefficients attenuate considerably the blocking and mosquito noise artifacts. Figure 16 shows an example on a film degraded by an MPEG-2 compression. The bandelet thresholding removes most of the compression distortions while keeping sharp image structures such as the shirt stripes.

Thresholding in a bandlet basis can be interpreted as "motion compensated noise removal". Spatio-temporal geometric flow cannot always be measured reliably, specially with noise and distortions. When this flow is not sufficiently precise, a bandelet thresholding does not take advantage of geometric correlations and the results are similar to a wavelet thresholding. Like in motion compensated compression, this can degrade the estimation precision but it does not introduce visible artifacts as in motion compensated deinterlacing. This is why bandelet thresholding is robust to geometric flow errors.

Details Enhancement An image blur reduces the amplitude of wavelet and bandlet coefficients, which measure image variations in space and time. Suppressing a blur requires amplifying the signal high frequencies, which often amplifies the noise as well. One can reduce the blur and the noise at the same time by thresholding to zero the smallest bandlet coefficients which are mostly dominated by the noise, and by amplifying the largest coefficients. If the amplification is too strong, it may produce Gibbs type oscillations near sharp transitions such as edges. To avoid such oscillations, the amplification can be controlled by a total variation measurement that limits the amplification factor when oscillations begin to appear. The resulting bandelet detail enhancement simultaneously removes the noise and sharpens the image without introducing oscillatory artifacts.

FIG. 16 – *Left :* Decompressed film frame corrupted by mosquito noise and blocking artifacts. *Right :* Distortion removal with bandlet thresholding.

5 Conclusion

SD to HD upconversion is extremely difficult because it requires computing missing pixels from noisy and distorted input images, at a considerable data rate. Developing more efficient procedures requires to innovate along the whole range from mathematics to image processing and fast parallel algorithms, back and forth. Let It Wave's upconverter is based on several key mathematical and algorithmic tools. The reduction of artifacts is obtained through the minimization of a total variation norm that measures oscillations in time and space. This process is further stabilized and distortions are removed with an adaptive spatio-temporal bandlet thresholding process. Despite the complexity of full spatio-temporal processing, these computations are implemented in a mid-size FPGA with parallel algorithms designed with a data flow approach.

Références

[1] E. Candès and D. Donoho. *Curvelets : A surprisingly effective nonadaptive representation of objects with edges.* Vanderbilt University Press, 1999.

[2] T. De Haan and E. Bellers. Deinterlacing - an overview. *Proc. IEEE*, 86(9) :1839–1857, 1998.

[3] M. N. Do and M. Vetterli. The contourlet transform : an efficient directional multiresolution image representation. *IEEE Transactions Image on Processing, To appear*, 2005.

[4] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81 :425–455, Dec 1994.

[5] S. Farsiu D. Robinson M. Elad and P. Milanfar. Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, 14(2) :47–57, 2004.

[6] Y. Faroudja. Method and apparatus for producing from a standard-bandwith television signal a signal which when reproduced provides a high-definition-like video image relatively free of artifacts. *US Patent*, 5,428,398, 1995.

[7] E. Le Pennec and S. Mallat. Sparse geometrical image representation with bandelets. *IEEE Transactions on Image Processing*, 2004.

[8] F. Malgouyres and F. Guichard. Edge direction preserving image zooming. *SIAM*, 39(1) :1–37, 2001.

[9] S. Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 11(7) :674–693, July 1989.

[10] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.

[11] Y. Meyer. *Ondelettes et Opérateurs*, volume 1. Hermann, Paris, 1990.

[12] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. In *Proceedings of the eleventh annual international conference of the Center for Nonlinear Studies on Experimental mathematics : computational issues in nonlinear science*, pages 259–268. Elsevier North-Holland, Inc., 1992.

[13] A. Verri and T. Poggio. Against quantitative optical flow. In *Proceedings of the International Conference on Computer Vision*, pages 171–180. IEEE Computer Society, 1987.