A note about the choice of the parameter for the Easy Path Wavelet Transform algorithm.

Stefano Ferrari and Licia Lenarduzzi

March 2, 2010

Abstract

We have implemented the rigorous form of the EPWT algorithm by Gerlind Plonka in Matlab and we test it on grey images. We show an application to a RGB image too. An index of local variation of the grey tones is used to quantify the presence of shades, and the compression rate, relevant to the wavelet coefficients, is tuned such that the shades are well rendered.

AMS subject classification: 65T60, 42C40, 68U10, 94A08

Keywords: wavelet transform along pathways, data compression, adaptivity, shade rendering, RGB image.

1 Introduction.

The nonlinear locally adaptive easy path wavelet transform (EPWT) was explored in [3] for sparse image representation.

The main idea of this transform is as follows.

In a first step, a path vector is determined through all indices of a given two dimensional set, in such a way that there is a strong correlation between the image values that correspond to adjacent pixels in the path vector.

Then, one level of a one-dimensional wavelet transform is applied to the image values that correspond to adjacent pixels in the path vector.

In the following levels, one needs to find path vectors through index sets of a low-pass image and applies again the wavelet transform.

A shrinkage of the wavelet coefficients is performed before calculating the inverse one-dimensional wavelet transform on the one-dimensional paths runned one after the other, in the order inverse to that of their determination.
The number of wavelet coefficients kept after the hard thresholding is the parameter associated to the EPWT algorithm and we denote it $M$; $M$ must be specified by the user. The numerical results of the algorithm are very accurate, even for $M$ not large. It must be noticed that theoretical results were given in [4]; there it is proven that, for a fixed image, the application of the EPWT leads to a $M$-term approximation of the form:

$$\| f - f_M \|_2^2 \leq CM^{-\alpha}$$

for piecewise Hölder continuous functions of order $\alpha$ (with $0 < \alpha \leq 1$) with allowing discontinuities along curves of finite length.

We have implemented the algorithm in Matlab and here we refer about our numerical experience. We have noticed two drawbacks: for a value of $M$ too small there can be blunders in the reconstructed image; on the other hand, with a value of $M$ large that assures accuracy, as confirmed by a good value of the PSNR, it can happen that the smooth parts of the image are not rendered smoothly enough.

In §2 we recall the spirit of the EPWT algorithm and we refer to [3] for a clear and detailed description. In §3 we address the choice of the parameter $M$ to fix for the above mentioned drawbacks. In §4 we show numerical examples.

## 2 A hint about the EPWT algorithm.

A digital function is evaluated on the integers

$I = \{(i,j), \ i = 0, \ldots, N_1-1, \ j = 0, \ldots, N_2-1\}$ with $N_1 \cdot N_2 = 2^L s$.

The grey tone $f(i,j)$ is associated to the $(i,j)$-th pixel.

For the generic pixel $(i,j)$, we consider the neighbourhood $N(i,j)$:

$$N(i,j) = \{(h,k) \in I \setminus \{(i,j)\} : |i-h| \leq 1, \ |j-k| \leq 1\}.$$

The path through $(i,j)$ includes the pixel of $N(i,j)$ not visited yet and for which the difference

$$| f(i,j) - f(h,k) |$$

is minimal. The pathway is interrupted when all the pixels of $N(i,j)$ have been visited already. However it is possible to run through the whole set $I$ as union of disjoint paths and to form $p^L$ permutation of $\{1, \ldots, 2^L s\}$. The core idea is: when moving along the path, the grey
tone changes less than in other directions for the most part of the path itself.

A one dimensional wavelet transform (DWT) is applied along the path vector.

When applying one level of the DWT, a version of the image with step of two pixels altogether is obtained (the unit is made of two pixels contiguous usually: they have an edge in common or a vertex in common).

A new search of the paths according to the minimal change of the grey tone between contiguous units is performed before applying an other level of the wavelet transform that leads to consider unities of four pixels altogether.

After performing $L$ iterations, a hard thresholding shrinkage procedure is applied to the wavelet coefficients, in order to keep the $M$ most important coefficients.

Now the one-dimensional inverse wavelet transform is performed by running the paths in the order inverse to that of their creation; we denote $\{s(i,j)\}$ the values obtained on $I$.

The wavelet representation turns out to be sparse.

About the storage of a path, the path usually connects neighbouring indices and one only needs to store the next direction of the path instead of the whole next index.

The wavelet transform considered in the current note is the classical periodic based on the $D - 4$ orthogonal filters.

3 Choice of the parameter $M$.

It can happen that undue blunders appear in the compressed image, when operating with a value of $M$ too small, for example few dark pixels against a clear sky.

It is because of few wavelet coefficients of the coarse levels of decomposition; by chance, the associated unit is made of more pixels not all contiguous (because the contiguous pixels had been already visited) and that are not spatially correlated (for example a unit is made of pixels of the clear sky together with pixels of a dark area).

This can be fixed by increasing the value of $M$ just till a prescribed reduction of the blunders is reached.
3.1 Choice of the parameter $M$: rendering of the shades.

Sometimes the shades in the smooth parts of the image are not well rendered and there is presence of blocky effects or stripe effects.

This is inherent to the technique that operates on unidimensional paths: two pathways can move contiguously but the technique operates independently on each of them.

We must observe that it is not easy to provide:

- a shrinkage with variable threshold (less thresholding for the edges and more thresholding for the shades) and very localized in its effects
- a smoothing of the shades by projecting on a $V_j$ space while maintaining the edges.

In the following we provide an index of local variation of the grey tones, by which we quantify the presence of shades. By comparing the value of the index at a location in the original image and the value of the index as calculated on the compressed image at the same location, we quantify whether the shade is well rendered or whether a larger value of $M$ must be tried in order to improve the fit of the compressed image.

In detail:

For each pixel $(i, j)$ of the original image, we calculate the local index of variation of the grey tone:

$$
\delta(i, j) := \max_{(\nu, \eta) \in N(i, j)} | f(i, j) - f(\nu, \eta) |
$$

We individuate the shades as those pixel locations $\{(i, j)\}$ at which the following equality is satisfied:

$$
\delta(i, j) < 2.5 \cdot \text{median}_{(h, k) \in \delta_{\text{index}}} (\delta(h, k)).
$$

We denote $I_{\text{shade}}$ the indices of the pixels of shade and $S_{\text{shade}}$ the set of the functional values (in double precision) associated to $I_{\text{shade}}$ by the EPWT algorithm.

We calculate the local index of variation by:

$$
\delta_s(i, j) := \max_{(\nu, \eta) \in N(i, j)} | s(i, j) - s(\nu, \eta) | .
$$

Within the set $I_{\text{shade}}$ we individuate those locations $\{(i, j)\}$ where the values $\{s(i, j), (i, j) \in I\}$ render the shade well enough, as those for which the following inequality is satisfied:

$$
\delta_s(i, j) < 2.5 \cdot \text{median}_{(h, k) \in \delta_{\text{index}}} (\delta(h, k)).
$$

and we denote $I_{s,\text{shade}} \subset I_{\text{shade}}$ their indices.
In case that the size of $I_{s,\text{shade}}$ is definitely smaller than the size of $I_{\text{shade}}$, we cannot be satisfied with the current set $S_{\text{shade}}$ and we must operate the EPWT with a larger value of $M$ in order that the size of $I_{s,\text{shade}}$ comes close to the size of $I_{\text{shade}}$.

4 Numerical Examples.

First of all we provide the results of a straightforward application of the EPWT algorithm.

**Example 1.**

The first example refers to the monitoring of a glacier. We show a dyadic subset of a set of Landsat TM data, red, near-infrared and short-wave-infrared, (courtesy of CNR-IREA Milano), covering a high mountain area in the Alps (Mt. Rosa). You see the data subset of the Lys glacier in Fig. 1.

The clean glacier (at top) is in blue; it continues in a tongue of glacier covered with debris, colored in a tone of magenta (it is pointed by the arrows in the figure). Few pixels in blue at the bottom of the tongue indicate paddles of water. From the tongue a water course originates. Away from the tongue, the ground is colored in tones of brown or green, depending on the lack of presence of vegetation and on the wetness.

We have applied the EPWT algorithm with factor of compression $K = 1/64$ separately to each of the three bands and we show the compressed image in Fig. 2.

We notice that the colors of the different local areas are maintained as in the original image; this is important because it is the particular color of the tongue of glacier covered with debris, that allows to classify the tongue correctly as part of the glacier.

• Now we show examples relevant to images in grey tones when we want to render well the shades according to what said in §3.1.

We stay with dyadic values for $M$, starting with the value of $M$ for which the factor of compression $K$ takes value 1/64.

To quantify the result we provide the PSNR value; to quantify how well the shades are well rendered we give the value

$$\gamma = \frac{\#(I_{\text{shade}}) - \#(I_{s,\text{shade}})}{\#(I_{\text{shade}})}.$$ 

**Example 2.**

For the image of the parrot ($128 \times 128$), see Fig. 7 at bottom left, we operate with $M = 256$ ($K = 1/64$) at first.
It results $PSNR = 35.5$ and $\gamma = 0.085$.

The result is not satisfying, as you see in the Figures 3 and 4, where on the left there is the detail of the original image, while on the right there is the detail as rendered by the EPWT: for example on the right of Fig. 3 the tuft on the head is not neat and it presents blurs both in the horizontal and vertical directions; the spot on the cheek is not well rounded. In the complementary way, one observes that the shades of the background are not rendered well.

With the choice $M = 512$ ($K = 1/32$), it results $PSNR = 38.09$ and $\gamma = 0.056$. The better rendering can be appreciated in the details on the right in the Figures 5 and 6.

In Fig. 7 in the bottom row, you see the full original image on the left and on the right the full reconstruction by EPWT with $M = 512$.

In the upper row, on the left you see the pixels of $I_{\text{shade}}$ in white (in black the remaining pixels), while on the right you see the pixels of $I_{s,\text{shade}}$ in white.

**Example 3.**

We refer to the image of the peppers $128 \times 128$. When operating with $M = 256$ ($K = 1/64$) it is $PSNR = 29.71$ and $\gamma = 0.05$.

In Fig. 8 in the bottom row, you see the full original image and on the right the full reconstruction by EPWT with $M = 256$.

In the upper row, on the left you see the pixels of $I_{\text{shade}}$ in white (in black the remaining pixels), while on the right you see the pixels of $I_{s,\text{shade}}$ in white.

**Example 4.**

We refer to an image $64 \times 64$ where the grey tones have a sygmoideal behavior. When operating with $M = 64$ ($K = 1/64$) it is $PSNR = 38.12$ and $\gamma = 0.17$.

In Fig. 9 at the bottom row, you see the full original image and on the right the reconstruction by EPWT with $M = 64$.

At the upper row, on the left you see the pixels of $I_{\text{shade}}$ in white (in black the remaining pixels), while on the right you see the pixels of $I_{s,\text{shade}}$ in white.

With $M = 256$, $K = 1/16$, it is $PSNR = 46.52$ and $\gamma = 0.086$.

With $M = 512$, $K = 1/8$, it is $PSNR = 53.33$ and $\gamma = 0.040$.

In Fig. 10 in the bottom row, you see the full original image and on the right the reconstruction by EPWT with $M = 512$.

In the upper row, on the left you see the pixels of $I_{\text{shade}}$ in white (in black the remaining pixels), while on the right you see the pixels of $I_{s,\text{shade}}$ in white.

**Example 5**

The image is relevant to a part $64 \times 64$ of the Tonga Trench located in the Pacific Ocean; this bottom of the sea is an example of surface
with rapidly varying data, see [2].

The original image is shown in Fig. 11. With \( M = 64, K = 1/64 \), it is \( PSNR = 36.5 \) and \( \gamma = 0.083 \); you see the result in Fig. 12.

With \( M = 256 \) it is \( PSNR = 40.9 \) and \( \gamma = 0.035 \).

With \( M = 512, K = 1/8 \), it is \( PSNR = 44.3 \) and \( \gamma = 0.020 \).

In Fig. 13 in the bottom row, you see the full original image and on the right the reconstruction by EPWT with \( M = 512 \).

In the upper row, on the left you see the pixels of \( I_{\text{shade}} \) in white (in black the remaining pixels), while on the right you see the pixels of \( I_{s,\text{shade}} \) in white.

References


Figure 1: Lys glacier; original image.

Figure 2: Lys glacier; result with $M = 256$. 
Figure 3: On the left: detail of the data. On the right: result with $M = 256$.

Figure 4: On the left: detail from the data. On the right: result with $M = 256$. 


Figure 5: On the left: detail from the data. On the right: result with $M = 512$.

Figure 6: On the left: original image. On the right: result with $M = 512$. 
Figure 7: On the left: original image. On the right: result with $M = 512$. 
Figure 8: On the left: original image. On the right: result with $M = 256$. 
Figure 9: On the left: original image. On the right: result with $M = 64$. 
Figure 10: On the left: original image. On the right: result with $M = 512$. 

14
Figure 11: Data from the Tonga Trench.

Figure 12: Tonga Trench. result with $M = 64$. 

15
Figure 13: Tonga Trench on the left. On the right result with $M = 512$. 