

# Adaptive Edge Orientation Analysis

Eric Van Reeth<sup>1,2</sup>, Pascal Bertolino<sup>2</sup>, Marina Nicolas<sup>1</sup>, Jean-Marc Chassery<sup>2</sup>

<sup>1</sup>STMicroelectronics, 12 rue Jules Horowitz, Grenoble, FRANCE

<sup>2</sup>GIPSA-LAB, 961 rue de la Houille Blanche, Grenoble, FRANCE

## ABSTRACT

This paper presents a method that detects edge orientations in still images. Edge orientation is a crucial information when one wants to optimize the quality of edges after different processings. The detection is carried out in the wavelet domain to take advantage of the multi-resolution features of the wavelet spaces, and locally adapts the resolution to the characteristics of edges. Our orientation detection method consists of finding the local direction along which the wavelet coefficients are the most regular. To do so, the image is divided in square blocks of varying size, in which Bresenham lines are drawn to represent different directions. The direction of the Bresenham line that contains the most regular wavelet coefficients, according to a criteria defined in the paper, is considered to be the direction of the edge inside the block. The choice of the Bresenham line drawing algorithm is justified in this paper, and we show that it considerably increases the angle precision compared to other methods as for instance, the method used for the construction of bandlet bases. An optimal segmentation is then computed in order to adapt the size of the blocks to the edge localization and to isolate in each block at most one contour orientation. Examples and applications on image interpolation are shown on real images.

**Keywords:** Wavelets, Bandlets, Curvelets, Edge Direction, Image Analysis, Multi-resolution

## 1. INTRODUCTION

Image processing has become in the last decade a very attractive field of research. Most of the processings concern image denoising, image rescaling, compression or deblurring. More and more advanced concepts have been introduced in order to make these processings converge towards the best image quality. Numerous studies have shown that image quality is strongly correlated to the quality of the image edges. This is the reason why a large number of processings adapt themselves to the edges, and try to improve or preserve the quality of edges as much as possible. Later on, some techniques have focused on taking into account the edge orientation to take advantage of the regularity along the edges. Thanks to these approaches, edges can be seen as long and regular structures instead of discontinuities, which is simpler to process. Among these techniques, we can name the work of Mallat and Le Pennec who created the bandlets,<sup>1</sup> Starck Cands and Donoho who introduced the ridgelets/curvelets,<sup>2</sup> or Do and Vetterli who invented the contourlets.<sup>3</sup> All these techniques focus on using the regularity of wavelet coefficients in some directions, in order to improve different processings (denoising, inpainting, compression). Other approaches have been developed, like the work of Perona<sup>4</sup> which builds a geometrical flow (non-wavelet based) by iteratively diffusing gradient directions. This last approach has been used by Moloney and Jiang<sup>5</sup> for interpolation applications.

The approach that we present here is wavelet based, and combines multi-resolution features of wavelets with an adaptive segmentation method of the image, to isolate a unique edge direction in each block. The first section of this paper deals with the resolution adaptation that is performed on the image. The second part explains the segmentation method that is performed in order to describe at best the geometrical features of the objects inside the image. An innovative method is then introduced to detect the directions of the edges in blocks, by finding the direction that maximizes the homogeneity of wavelets coefficients. Examples of direction descriptions are shown at the end of the paper. Moreover, applications on image interpolations are also shown. However, the interpolation method will not be discussed here, since it is not the point of this paper. It is however useful to look at such applications since it proves the directions we have detected are coherent with the image content.

---

Further author information: send correspondence to Eric Van Reeth. E-mail: eric.vanreeth@st.com

## 2. OVERVIEW OF THE METHOD

The general method is shown in Figure 1. The resolution adaptation consists of dividing the image into macro-blocks and choosing in each macro-block the resolution of the "A Trous" wavelet transform at which the edges are best represented. The size of the macro-blocks must be chosen with regards to the size of the objects inside the image, so that the resolution choice will reasonably be based on one edge of one object.

Then, a direction analysis is performed to detect the dominant direction in each macro-block. The regularity of the wavelet coefficients of the macro-block is tested to check if there are other directions inside the macro-block. In the case where there are other directions, a segmentation of the macro-block into four sub-blocks of equal sizes is performed. In each sub-block, the direction detection process is iterated until a direction is isolated in one block or until the minimum block size is reached.

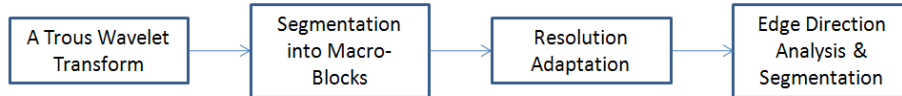


Figure 1. Global process of the whole direction detection method.

## 3. RESOLUTION ADAPTATION

### 3.1 "A trous" Wavelet Algorithm

Wavelets are a great tool to study image contours because they separate high and low frequencies and above this, they offer flexibility regarding resolution adaptation.<sup>6</sup> A complete description of wavelet theory and multi-resolution spaces can be found in.<sup>7</sup> Briefly, when performing a discrete wavelet transform of an image  $I$ , the image is separated in two spaces : the *approximation* space  $A$ , and the *detail* space  $D$ . The approximation space contains the low frequencies of  $I$  whereas the detail space contains the high frequencies, including the edges of  $I$ . By iterating this transform on approximation images, we obtain a frequency description of the image for different resolutions. The transform we use is a dyadic transform, meaning that the resolution is divided by two at each scale. If  $j$  is the scale of the wavelet transform, the resolution of the transform spaces decreases with a factor of  $2^{-j}$  for  $j = 1, \dots, J$ .

In our study, we use Mallat's algorithm called the "A trous" transform, which has the advantage of holding the same image size across the different scales. Thus, it introduces redundancy at low resolutions which we use as additional information to detect the orientations of edges. An example of a discrete dyadic wavelet decomposition is shown in Figure 2. We observe on this example that at the finest resolution, the detail image contains fine details (fine pant stripes) whereas at lower resolution it contains coarser details (pant border).

### 3.2 Local Resolution Choice

In our study, we use this property in our study to locally adapt the scale of the wavelet transform to the contour characteristics, and thus make the processing at the optimal local resolution. Here is the description of the process. Let  $D_{2^{-j}}$  be the detail image at resolution  $2^{-j}$  (or scale  $j$ ). Note that our study will only be performed on detail images since our study concerns edges, which are high frequency structures that only appear in the detail space. We build a partition of the images  $D_{2^{-j}}$  by dividing them into macro-blocks. Let  $\Omega_{2^{-j}}^n$  be the  $n$ -th macro-block of the partition of the image  $D_{2^{-j}}$ . For each macro-block  $\Omega_{2^{-j}}^n$ , we choose the optimal scale  $J_{opt}(n)$  as the one that maximizes the amplitude of the wavelet coefficients.

$$J_{opt}(n) = \arg \max_{j=1, \dots, J} [\sup(\Omega_{2^{-j}}^n) - \inf(\Omega_{2^{-j}}^n)] \quad (1)$$

An illustration of this resolution adaptation can be seen in Figure 3. In this example, the macro-blocks  $\Omega_{2^{-j}}^n$  are  $16 \times 16$  pixels squares and we choose among 3 scales :  $j = 1, 2, 3$ . We easily that "low frequency" edges (hair and face) are represented with the coarsest scale ( $j = 3$ ) whereas "high frequency" edges (scarf) are represented with the finest scale ( $j = 1$ ). This fast and simple approach leads to better results when detecting edges directions since edges are then represented at their optimal resolution.

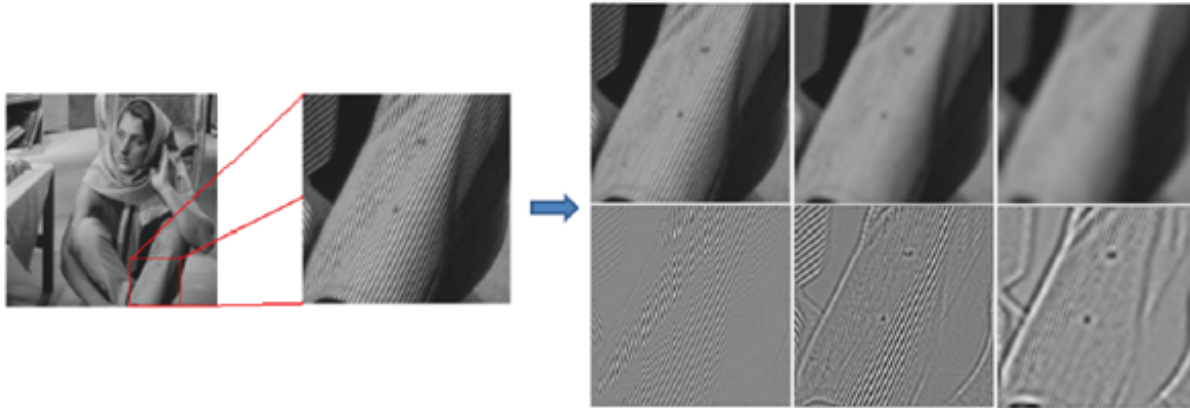


Figure 2. Example of a three level dyadic wavelet decomposition using the Mallat "A trous" algorithm. Resolution is decreasing from left to right. Approximation images are on top and Detail images are at the bottom.

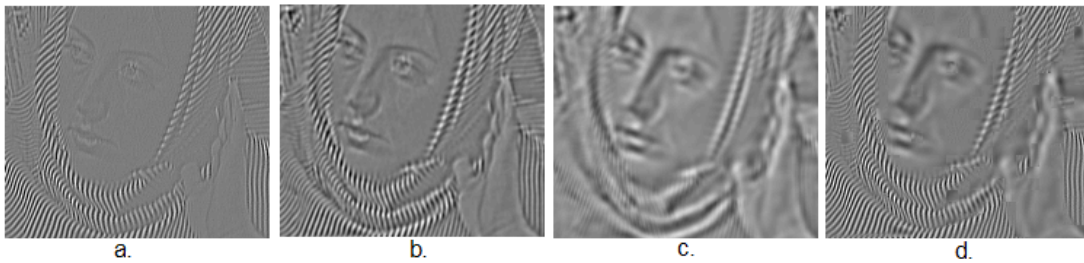


Figure 3. a) Detail Image  $D_{2-1}$ . b) Detail Image  $D_{2-2}$ . c) Detail Image  $D_{2-3}$ . d) Resolution adapted image.

## 4. IMAGE SEGMENTATION

In part 3, we showed that edges can have very different characteristics, going from low frequency transitions to high frequency textures. However, other characteristics have to be taken into account to detect their directions : their size (long, short), their neighborhood (alone or close to each others) and their shape (curved or straight). To compute their directions, we choose to isolate straight parts of edges in square blocks of appropriate size, in order to give a unique edge direction for each block. Big blocks will then be defined for long and straight edges. On the other side, small blocks will be used for parts of the image where edges are concentrated and have different directions. Thus, this section describes a method that adapts the block sizes to isolate in each block at most one edge direction.

### 4.1 Quad-tree segmentation

The quad-tree segmentation is a well-known method that can perform this task. Our approach is similar to the quad-tree segmentation. We analyze the homogeneity of a criterion inside each macro-block created in section 3, of size  $(N \times N)$  by comparing it to a reference value (threshold) :

- if the macro-block is considered homogeneous compared to the threshold, it is not divided.
- if the macro-block is considered not homogeneous, it is divided into four blocks of size  $(N/2 \times N/2)$ , and the process is iterated on these 4 blocks individually.
- the process stops when no more division should be made (the minimum block size is reached or each block contains at most one edge direction).

In our case, the homogeneity criterion is the regularity of wavelet coefficients in a certain direction  $\theta$  (the determination of  $\theta$  is explained in Part 5). A tolerance factor  $\alpha$  is also introduced to make the difference between

homogeneous and non-homogeneous blocks. Let  $G_{ort}$  be the integral over the block of the absolute values of the vertical and horizontal gradients. Let  $G_{dir}$  be the integral over the block of the absolute values of gradients along the direction  $\theta$ .

If we have :

$$G_{dir} \leq \frac{G_{ort}}{\alpha} \quad (2)$$

i.e. if the directional gradient is weak enough ( $\alpha$  was experimentally set to a value determined around 5) compared to the orthogonal gradients, the block is considered to be regular in the direction  $\theta$ , and the block is not divided.

On the other side, if we have :

$$G_{dir} > \frac{G_{ort}}{\alpha} \quad (3)$$

then the block is divided into four sub-blocks of same size. Note that we try to keep blocks as big as possible since we will see in Part 5 that the precision we obtain on the angle value increases with the size of the block. However small blocks are needed to capture quick changes of orientations.

## 4.2 Example of segmentation

An example of segmentation performed with this algorithm is shown in Figure 4. We can verify that at most one

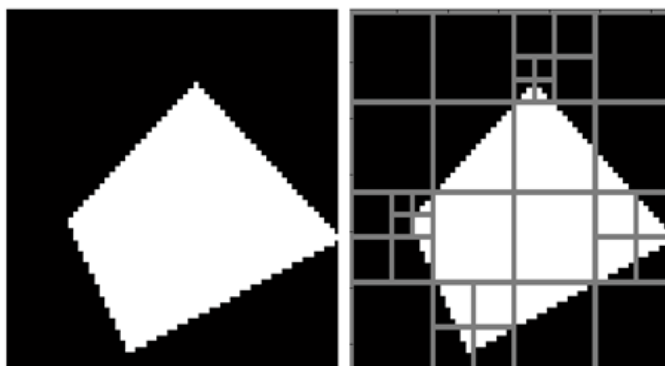


Figure 4. Left: Original image. Right: Adaptive segmentation performed with the described algorithm.

contour direction is present inside each block. In this example, the maximum block size authorized was  $16 \times 16$  pixels, and the smallest was  $4 \times 4$ .

Note that the segmentation is shown here over the original image for clarity, but this segmentation is actually obtained from the wavelet transform image with resolution adapted wavelet coefficients.

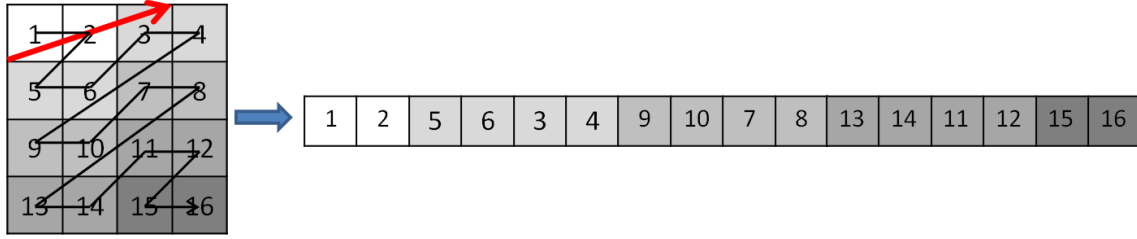
## 5. DIRECTION DETECTION METHOD

The method presented here consists in finding the dominant direction in square blocks, composed of wavelet coefficients.

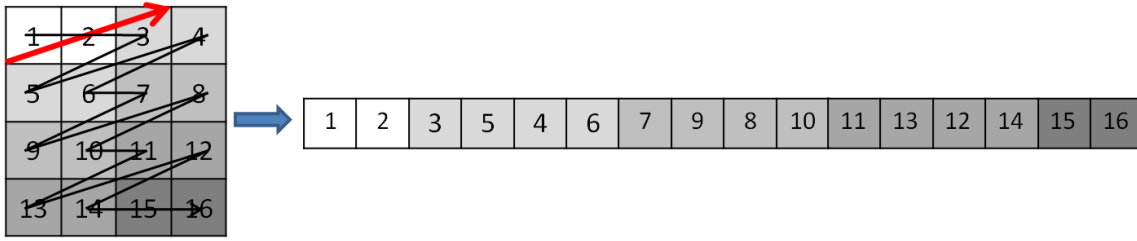
To do so, we search for the Bresenham line which links the wavelet coefficients in the most regular way. The direction of the Bresenham line that maximizes the regularity of the wavelet coefficients is considered as the dominant direction of the block. The following paragraphs describe the extraction of Bresenham lines from a square block and the definition of the regularity criterion.

### 5.1 Pixels sequences

To design Bresenham lines inside a square block, a 2-D  $\rightarrow$  1-D transform is performed. The 2-D block is transformed into a 1-D sequence :  $\bar{S}(\theta)$  of consecutive pixels (except line jumps), containing all the pixels of one block and following the Bresenham line drawing algorithm along the direction  $\theta$ . See Figure 5(a) for an example of such a transform. The Bresenham line design algorithm is chosen because it designs a digital line by minimizing pixel



(a) Sequence  $\bar{S}(\theta)$  created with the Bresenham algorithm



(b) Sequence  $\bar{S}(\theta)$  created with the Bandlet algorithm

Figure 5. Comparison of two different ways of creating sequences.

after pixel the geometric error compared to the continuous line of the same direction. This gives the insurance that the created sequence links neighboring pixels in the optimal way to represent the direction  $\theta$ .

Moreover, it is interesting to notice that the number of digital lines that can be drawn in a square block increases with the size of the block. The angular resolution increases with the size of the block since we can discriminate a larger number of angles. Also, working on a larger number of pixels increases the robustness of the direction computation by reducing the influence of noise or parasite directions (aliasing). This is why the algorithm tries to work on large blocks whenever it is possible.

## 5.2 Definition of the regularity criteria along a direction

We describe here the computation of a homogeneity criteria of the pixels sequences. Let  $d(\theta)$  be a 1-D vector containing the absolute values of the gradient along  $\bar{S}(\theta)$ . For a sequence that is  $N^2$  pixels long, we have :

$$d(\theta) = \sum_{i=1}^{N^2-1} |\bar{S}(\theta)[i+1] - \bar{S}(\theta)[i]| \quad (4)$$

$d(\theta)$  can be seen as a measure of irregularity of the pixel sequence because it sums the absolute values of the gradient along the sequence. Thus, we look for the direction  $\theta$  that minimizes  $d(\theta)$  as the sequence  $\bar{S}(\theta)$  corresponding to the direction of the contour, is the one that contains the fewest discontinuities.

As described earlier, the sequences  $\bar{S}(\theta)$  are composed of wavelet coefficients that have different resolutions depending on the choice of  $J_{opt}$ , performed in Part 3. By taking advantage of this resolution adaptation, we can improve the homogeneity criteria with an approach similar to the one performed in Part 3.

By computing the 1D orthogonal wavelet transform of  $\bar{S}(\theta)$  (Haar transform for example), we can optimally emphasize the discontinuities inside  $\bar{S}(\theta)$ , since we know their resolution (given by  $J_{opt}$ ). More precisely, the scale  $J_{Haar}$ , of the 1D orthogonal wavelet transform performed on  $\bar{S}(\theta)$  is chosen at:

$$J_{Haar} = J_{opt} - 1 \quad (5)$$

Note that no wavelet transform is performed for  $J_{Haar} = 0$ , i.e.  $J_{opt} = 1$ . Thanks to this resolution adaptation, sequences containing many discontinuities (sequences that do not follow the direction of the edge) will be more

”penalized” when  $d_{WT}(\theta)$  is computed, which makes the discrimination of the optimal direction easier.

At the end,  $d(\theta)$  is computed as follows:

$$d_{WT}(\theta) = \sum_{i=1}^{N^2-1} |WT_{J_{Haar}}^{Haar}(\bar{S}(\theta)[i+1]) - WT_{J_{Haar}}^{Haar}(\bar{S}(\theta)[i])| \quad (6)$$

In Figure 6,  $d_{WT}$  is plotted for a block containing a straight oblique edge. We observe a minimum of  $d_{WT}$  at 78 degrees which corresponds to the direction of the contour.

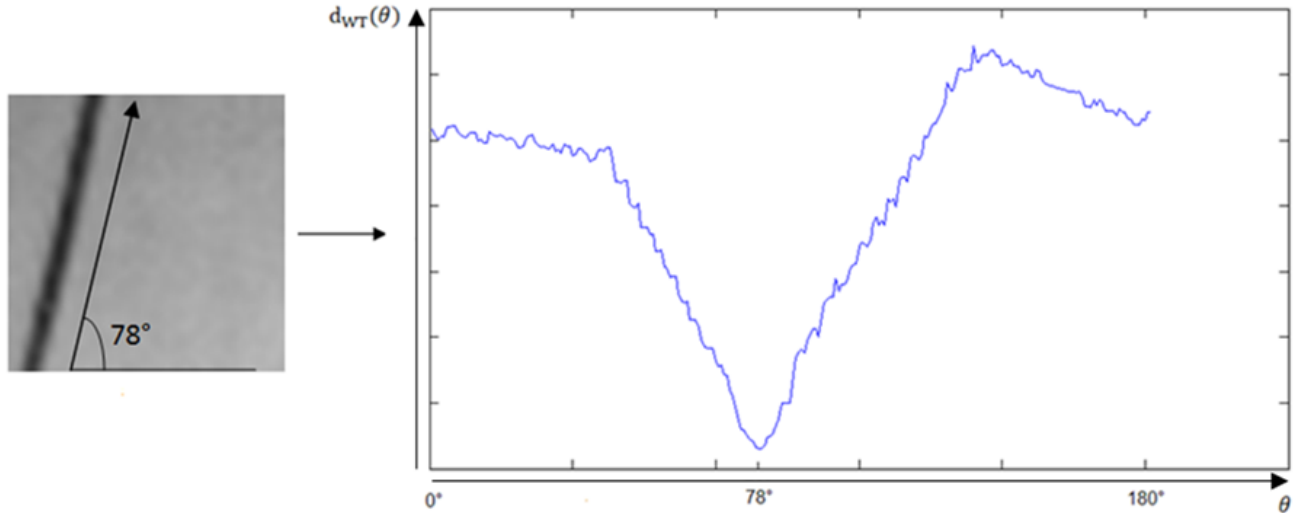


Figure 6. Left: block to be analyzed. Right: Plot of  $d_{WT}(\theta)$ .

One can find in Figure 7 a complete scheme that summarizes the segmentation method as well as the direction detection method performed inside each macro-block.

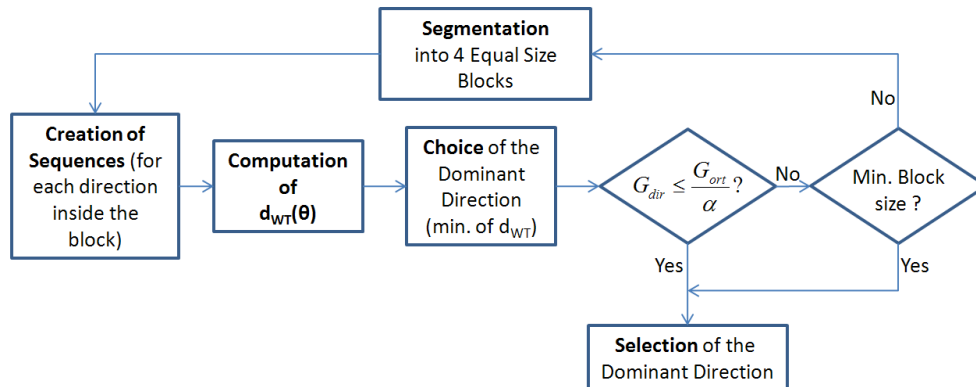


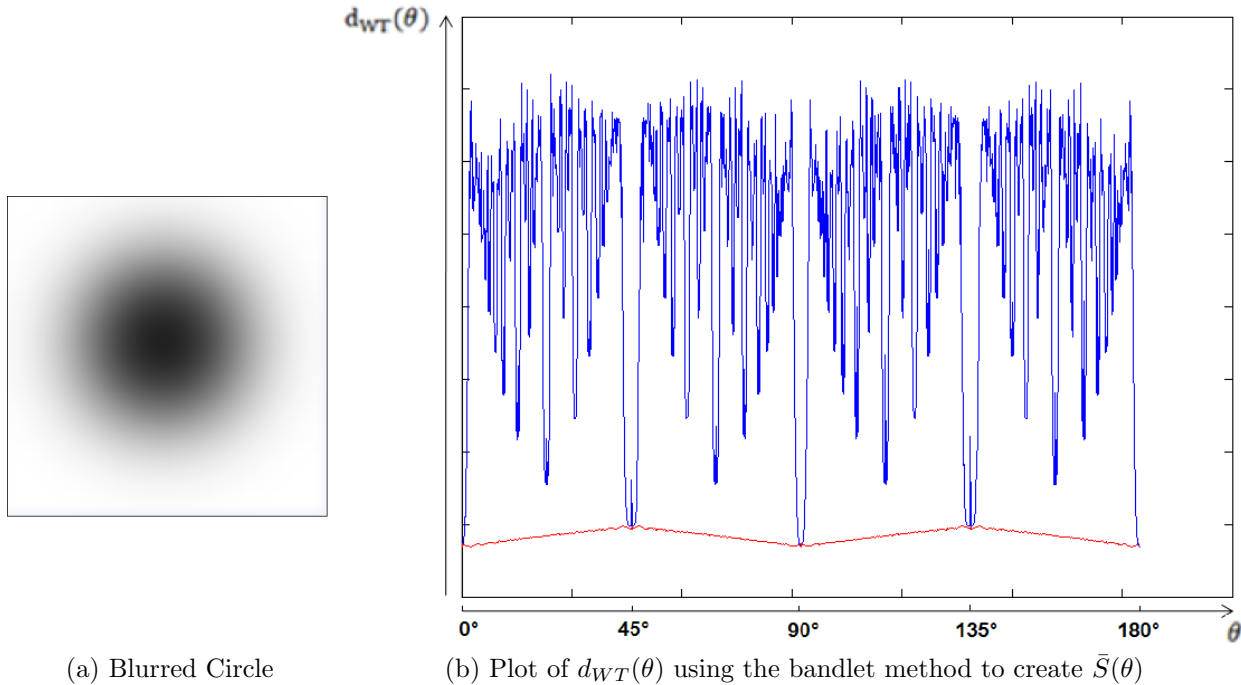
Figure 7. Complete algorithm of the direction detection method and the segmentation performed inside a macro-block

### 5.3 Why chaining neighboring pixels?

A close concept can be found in the construction of bandlet bases introduced by Peyre and Mallat<sup>89</sup> to detect directions inside blocks. However, direction detection is performed in a different manner which is described in.<sup>10</sup>

Basically, instead of chaining neighbor pixels of the block along a Bresenham line, a rotation matrix is applied on the matrix containing the pixel position in the block. Let  $\theta$  be the rotation angle. Then, the pixels are chained by order of proximity to the continuous lines of angle  $\theta$ .

Although this method is theoretically the best way to create pixels sequences that follow a given direction, one can notice that using this method as a direction detector is not optimal since it tends to detect what we call the "basic" directions (i.e.  $\pi/4, \pi/2, 3\pi/4, \pi$ ) more than others. This can be observed in Figure 8, where the direction detection is applied to a blurred circle. We expect that no directions will be privileged and that  $d_{WT}(\theta)$  will be flat, showing that all directions are equivalent. On the contrary we can observe that low peaks



(a) Blurred Circle

(b) Plot of  $d_{WT}(\theta)$  using the bandlet method to create  $\bar{S}(\theta)$

Figure 8. Comparison of the bias introduced by the bandlet method (blue) and our method (red).

of  $d_{WT}(\theta)$  are obtained for the 4 basic directions, and more generally for values factor of  $\frac{k\pi}{2^n}$ , for  $k = 1, 2, \dots, 2^n$ . This indicates that pixels along these angles are seen as more regular as for other directions, which explains why basic directions tend to be detected too often.

Our reason for this bias can be that for basic angles, the chained pixels inside  $\bar{S}(\theta)$  are geographically closer to each other than for other directions. This implies that inside a block (finite domain), more accurate chainings can be performed for basic directions. On the opposite, more inappropriate chainings have to be performed for other directions, which explains the higher values found in the computation of  $d_{WT}(\theta)$ .

One can observe in Figure 8 that our method also has a small bias but which is far weaker than the one inserted in the bandlet method.

Note that for the basic angles, values of  $d_{WT}(\theta)$  are identical for both our method and the bandlet method since these angles will be represented by the chaining of neighboring pixels in the bandlet method as well.

## 6. RESULTS

Here are some results obtained using the whole process described in this paper. The results of the decomposition and the direction analysis are shown on Figure 9. This example contains many different directions. Our algorithm succeeds in following the directions of high frequency contours (stripes) as well as lower frequency ones (arm) thanks to the resolution adaptation that represents the image edges at the optimal resolution. The detected directions also seem to fit the actual contour directions thanks to the accuracy of the direction detection

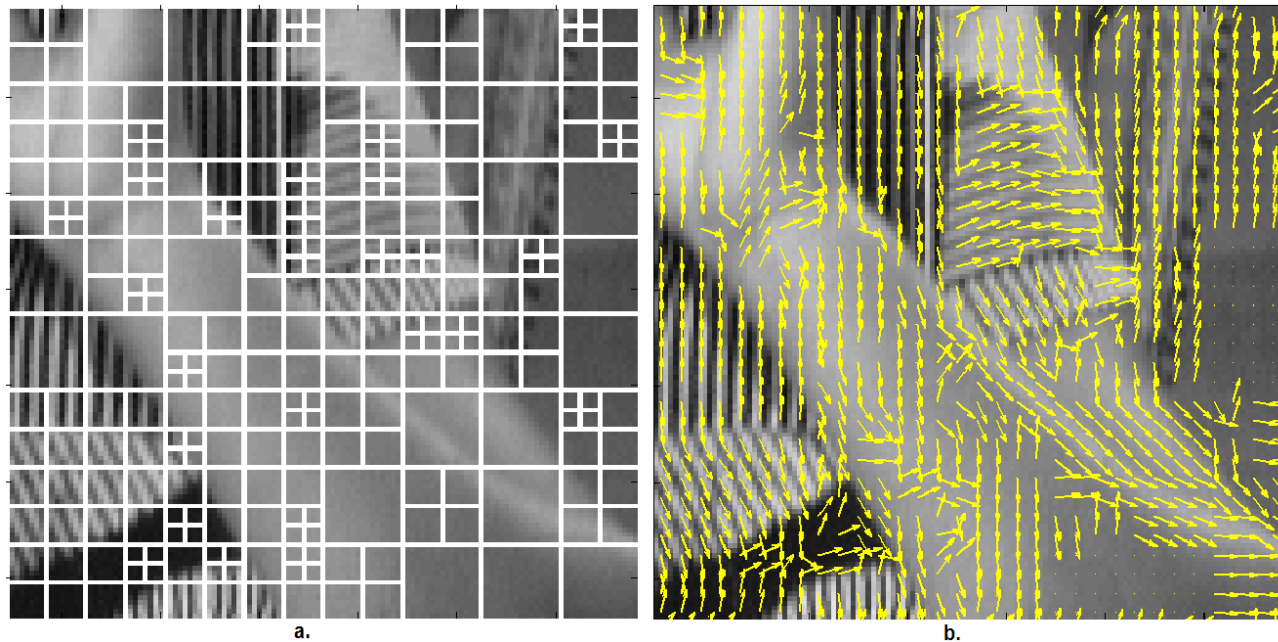


Figure 9. a) Decomposition performed on the image. b) Associated directions.

method. Finally quick changes of edges orientations can be handled thanks to the computation of an appropriate segmentation. Some remarks can however be made about this method:

- The block locations are fixed. We can only adapt the division inside the block, so it sometimes happens that even inside the smallest block size, there is still more than one direction.
- Weaker contours can be masked by well contrasted contours inside a block. Thus, the chosen direction will be the one of the strong contour without any further division.
- This method is quite demanding in computing resources since all the directions are tested per block, before choosing the optimal one. We could think of introducing statistical parameters, based for instance on the detected directions of neighbor blocks, to reduce the number of calculations.

A comparison with the orientation detection method of Perona (as described in<sup>5</sup>) can be seen in Figure 11, from the image part selected in Figure 10. We can observe that our flow is much more regular and closer to the real directions of the image content. The precision gain is obtained by working on blocks that are big enough to capture the actual direction of the contour, and that contain resolution adapted wavelet coefficients. This gives us more information to detect the edge orientation.

## 7. APPLICATIONS

Many applications can benefit from our multi-resolution directional description of images. Among them:

- Image enhancement that comprises deblurring, denoising, anti-aliasing
- Inpainting in particular in images that include long regularities
- Image compression that can be improved when compressing objects according to their geometrical characteristics
- Image interpolation (zoom) can also be improved by interpolating along the contour directions to avoid most of the classical interpolation artifacts (blur, ringing, staircase effect).

We show the interest of our directional decomposition by applying it to image interpolation (Figure 12). The figure shows the efficiency and the accuracy of our direction detection technique. The example shown is a four times interpolation performed from Figure 10. As expected, a reduction of the staircase and aliasing artifacts can be noticed in our interpolation thanks to the accuracy of the direction detection.

## 8. CONCLUSION

A complete method for the detection of edges directions has been presented in this paper. It provides a good description of any kind of edges thanks to the resolution adaptation. Moreover, it provides a high precision on the angle value since the directions of edges are captured inside blocks that are as big as possible. And finally, a novel non-biased direction detection method is explained based on the digital line drawing algorithm of Bresenham. We showed some applications on image interpolation, and our research now focuses on this point, but it is important to keep in mind that many other applications can benefit from such a directional analysis.

## REFERENCES

- [1] LePennec, E., *Bandelettes et representation geometrique des images*, PhD thesis, Polytechnique (2002).
- [2] Starck, J., Candes, E., and Donoho, D., “The curvelet transform for image denoising,” *IEEE Transactions on image processing* **11**(6), 670–684 (2002).
- [3] Do, M., “The contourlet transform: an efficient directional multiresolution image representation,” *IEEE Transactions on image processing* **14**, 2091–2106 (2005).
- [4] Perona, P., “Orientation diffusions,” *Image Processing, IEEE Transactions on* **7**(3), 457–467 (1998).
- [5] Jiang, H. and Moloney, C., “A new direction adaptive scheme for image interpolation,” *ICIP* **3**, 369–372 (2002).
- [6] Lindenberg, T., [*Principles for Automatic Scale Selection*], Academic Press (1998).
- [7] Mallat, S., [*A Wavelet Tour of Signal Processing*], Academic Press (1999).
- [8] Peyre, G., “Geometrical image estimation with orthogonal bandlet bases,” (2007).
- [9] Mallat, S. and Peyre, G., “A review of bandlet methods for geometrical image representation,” *Numerical Algorithms* **44**(3), p. 205–234 (2007).
- [10] Peyre, G., “Bandlet toolbox, [www.mathworks.com/matlabcentral/fileexchange](http://www.mathworks.com/matlabcentral/fileexchange),” (2007).

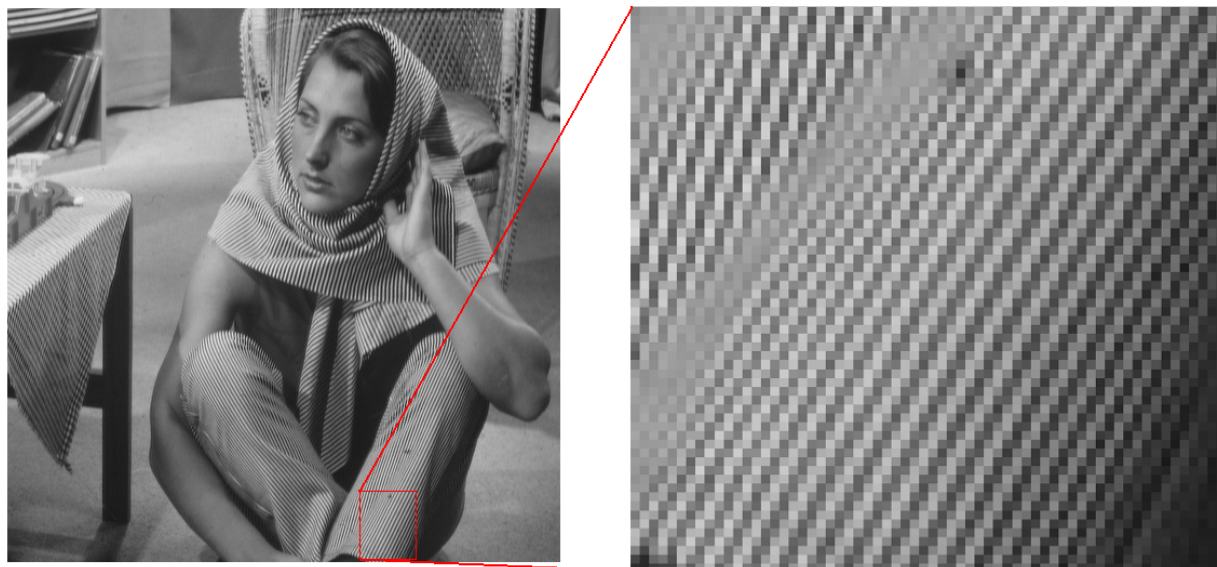
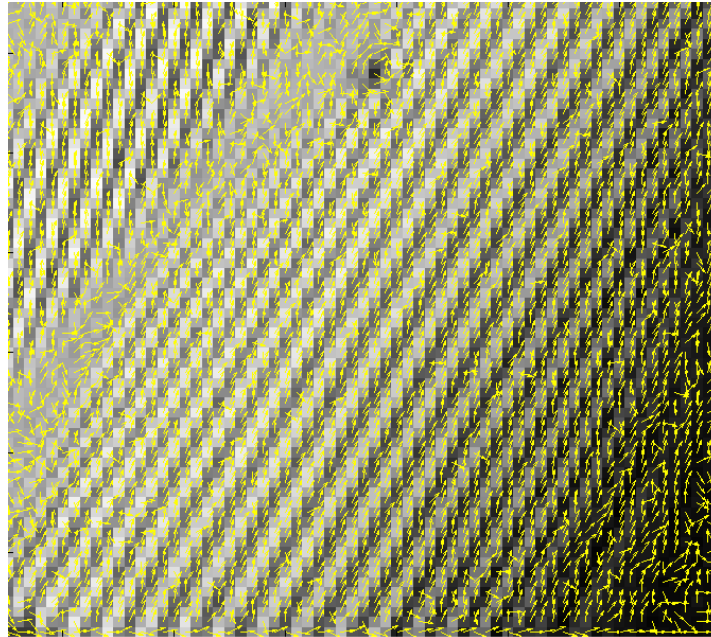
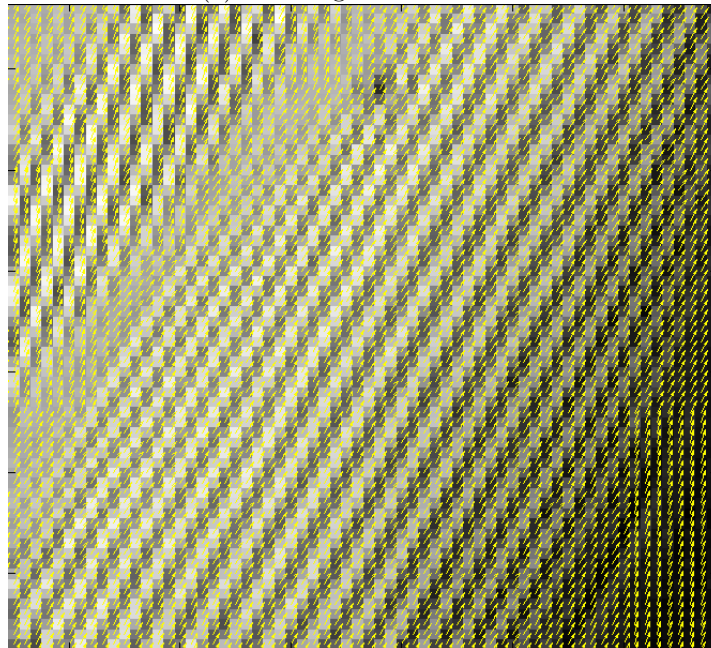


Figure 10. Selection of a part of the image Barbara to perform a direction analysis.

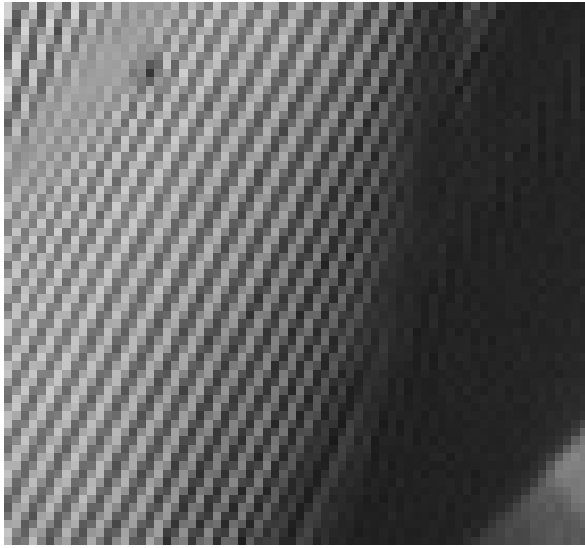


(a) Perona geometrical flow

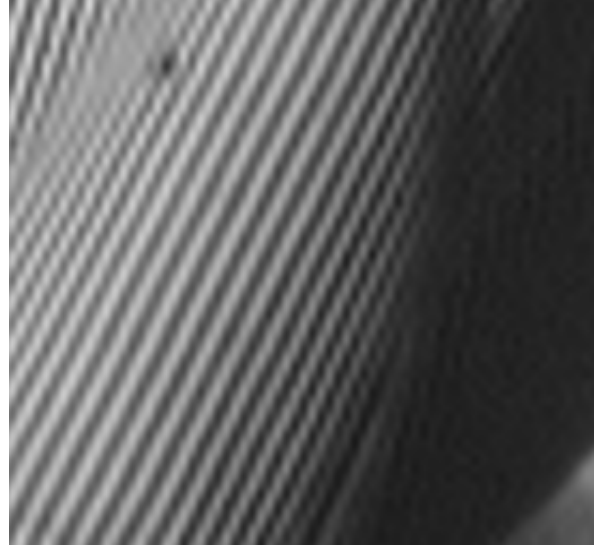


(b) Our geometrical flow

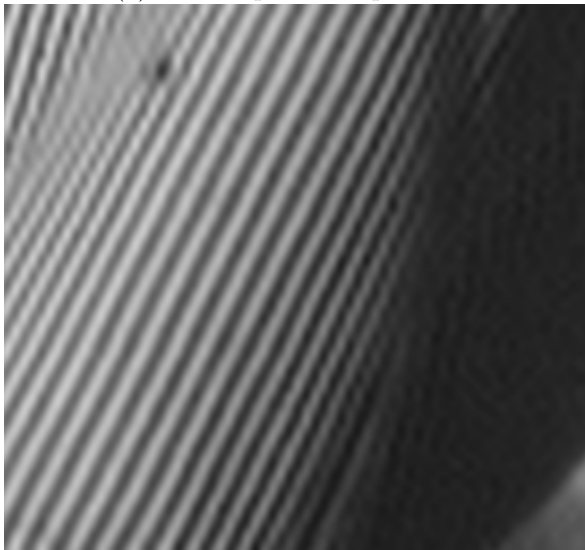
Figure 11. Comparison of our detection method (a), with Perona's method (b).



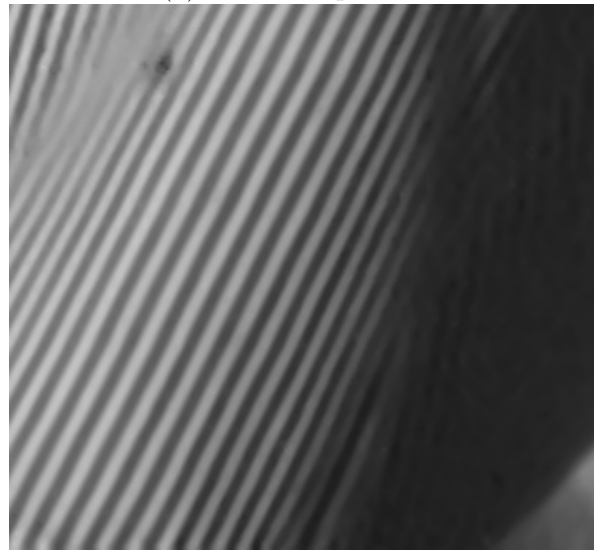
(a) Nearest pixel interpolation



(b) Cubic interpolation



(c) Spline interpolation



(d) Our directional interpolation

Figure 12. Comparison of different interpolation techniques.