

Image Interpolation Based on a Multi-resolution Directional Map

Eric Van Reeth^{1,2}, Pascal Bertolino², Marina Nicolas¹

¹STMicroelectronics, 12 rue Jules Horowitz, Grenoble, FRANCE

²GIPSA-lab, 961 rue de la Houille Blanche, Grenoble, FRANCE

ABSTRACT

This paper describes an interpolation method that takes into account the edge orientation in order to avoid typical interpolation artifacts (jagging, staircase effects...). It is first based on an edge orientation estimation, performed in the wavelet domain. The estimation uses the multi-resolution features of wavelets to give an accurate and non-biased description of the frequency characteristics of the edges, as well as their orientation. The interpolation is then performed, using the edge orientation estimation, to improve a reference interpolation (cubic-spline for instance). This improvement is carried out by filtering the edges with a gaussian kernel along their direction in order to smooth the contour in the direction parallel to the edge, which avoids disturbing variations across them (jagging and staircase effects). This technique also keeps the sharpness of the transition in the direction perpendicular to the contour to avoid blur.

Results are presented on both synthetic and real images, showing the visual impact of the presented method on the quality of interpolated images. Comparisons are made with the usual cubic-spline interpolation, and with other edge-directed interpolation techniques to discuss the choices that have been made in our method.

Keywords: Wavelets, Gabor filtering, Edge Direction, Image Analysis, Multi-resolution, Image interpolation, Upscaling

1. INTRODUCTION

Image interpolation has been widely studied in the past twenty years due to its huge number of applications (image zooming, display on flat TV panels, etc...). These techniques can be divided in two categories, which are non-adaptive and adaptive methods. Non-adaptive methods such as bilinear, bicubic, polyphase or spline interpolations are usually fast and easy to implement. However, they introduce a lot of well known artifacts such as jagging, staircase effects, blur or ringing on image edges.

Adaptive methods adapt their processing depending on the properties of the pixels, especially the ones on edges. In a previous article,¹ we introduced a method that detects the orientation of edges. The edge direction is indeed a primordial characteristic since the gray level variations parallel to the edges have to be smooth, whereas it is important to keep sharp transitions across the edge to avoid blur.

This paper focuses on an edge-directed interpolation technique, based on our estimation of edge orientations. The first part briefly reviews this method. In this part, we study how wavelets can improve the analysis, and how challenging it is to be accurate both on the edge position and on the edge direction in the discrete domain. Then, we will detail how the edge direction is used in our interpolation method. Finally, results and comparisons with existing edge directed interpolations will be shown.

2. EDGE ORIENTATION ESTIMATION

The goal of this estimation is to give an accurate approximation of the edge direction, and of the edge position. Our estimation of the edge direction is performed in the wavelet domain, in order to take advantage of the multi-resolution features of this transform. The method is indeed partly based on a local optimization of the resolution in order to fit the frequency of the gray level variation of the edge.

The method contains the following steps:

Further author information: send correspondence to Eric Van Reeth. E-mail: eric.vanreeth@gipsa-lab.grenoble-inp.fr

- The image is transformed at three different scales with an isotropic undecimated wavelet transform (IUWT)² that keeps the same image size along the scales.
- A partition of the original image is done, by dividing it into blocks of equal size.
- In each block the scale that has the strongest wavelet coefficient amplitude is selected. This scale is the one that best represents the frequency of the transition.
- In each block the predominant direction is estimated by finding the direction of the 8-connexity discrete line along which the variation of the wavelet coefficients is minimal. If the variation along one direction is small enough compared to other directions, then there is a high chance that the possible contour is parallel to this direction.
- If more than one direction have small variations, the block is said to contain several edge orientations. Then, the block is divided with a quad-tree algorithm, in four equal-size smaller blocks. The direction estimation is then iterated in each sub-block until at most one edge direction is detected in each sub-block, or until the minimum block size is reached.
- At the end, the directional map is obtained as a partition of blocks which indicate the optimal block resolution as well as the possible edge orientation. An example of a directional map is given in Figure 1.

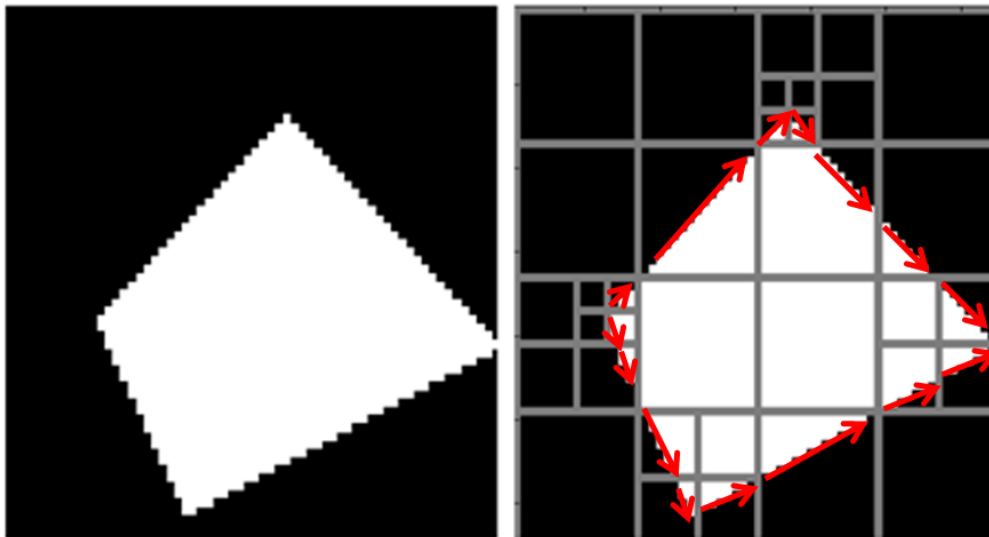


Figure 1. Orientation estimation and quad-tree division of a synthetic image.

Other edge direction estimators have been studied^{3,4,5,6} The approach that inspired ours was developed in,⁷ for the construction of bandlet bases. To define the orientation along which bandlets are designed, a quad-tree which isolates at most one direction per block is built. However, they first estimate directions in small blocks which are possibly merged into one big block if the amount of variation in the big block is smaller than the sum of the variations in the small blocks. This process implies edge orientation estimations in every small blocks and in every bigger blocks. On the opposite, our method avoids direction estimations in all small blocks since we start the estimation in big blocks that are only divided if needed. Another advantage of our approach is that starting the orientation estimation in big blocks allows us to have a high precision on the first estimated angle value, whereas the angle resolution is lower in smaller blocks.

Of course, using small blocks allows to have a better spatial accuracy. They adapt better to sudden changes of orientation (corners). The size adaptation of the quad-tree allows to have a high precision on the angle value of long straight edges whereas it can also captures discontinuities in the edge orientation.

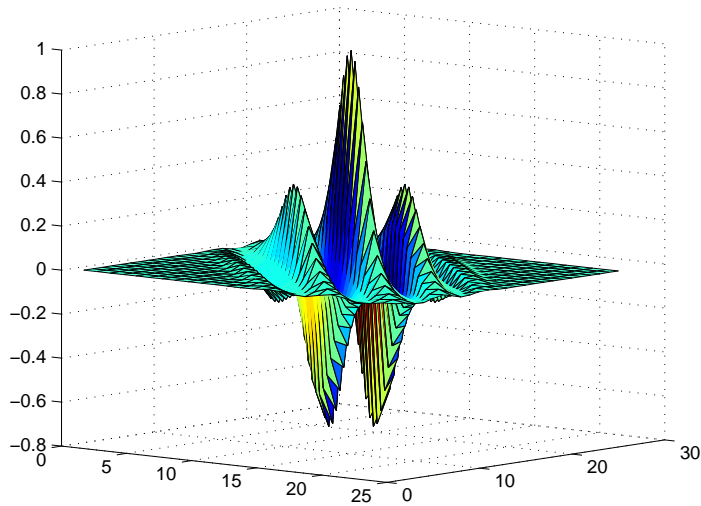


Figure 2. Example of a Gabor filter.

3. EDGE-DIRECTED INTERPOLATION

The principle of our interpolation is to use the local directional information given by the quad-tree to ensure the smoothness in the direction of the edge. This is done by combining a isotropic interpolation (cubic-spline) with a directional correction (filtered version of the isotropic interpolation). The combination of these two methods is essential to obtain a nice rendering of the directional edges (filtered version) and to preserve the quality of textures and edge-free regions (isotropic interpolation). This is an important aspect of our interpolation since we want to ensure a nice global rendering of the image, and not only on edges.

Most directional approaches suffer from degradations on edge-free regions⁸⁹¹⁰ because they interpolate the whole image based on local directions, creating false edges in uniform regions. Yet, assuming the local regularity of gray levels in one direction is a very strong assumption and this information should be handled with great care. Following this idea, Wang and Ward have developed an interesting technique based on the detection of ridges (straight edges) in images,¹¹ which allows them to interpolate directionally only pixels situated on straight edges and avoid the apparition of false edges.

In our approach, we use the output of a Gabor filter to locate the pixels belonging to edges having the direction given by the directional map. The output of the Gabor filter is used as a mask which separates pixels to be directionally filtered and the others. This process is detailed hereafter.

3.1 Gabor Filtering

Gabor filters are often used in directional analysis. The impulse response is a two-dimensional gaussian modulated by a sinusoidal plane. The main parameters of this filter are the scales of the two axis of the gaussian envelope: (σ_a, σ_b) , the angle of the filter: θ , the frequency and the phase of the sinusoid: (ω_0, P_0) . Figure 2 shows an example of a Gabor filter in the spatial domain.

To locate pixels that have to be directionally interpolated, each block of the quad-tree segmentation is filtered with a Gabor filter. The Gabor filter is positioned according to the orientation detected inside the block. The filter output will be high on pixels having this orientation and low on the others, and thus can be used as a mask. In order to be more precise on the detection, the filter is not directly applied to the original image but to the finest scale of an isotropic undecimated wavelet transform (IUWT) that is described in.¹² The interest of such a transform is that edges appear more clearly, being less disturbed by background variations, and that spatial information is kept. Let $G(x, y)$ be a 2D discrete Gabor filter, $B_N(x, y)$ the N-th block of the image, $W_N(x, y)$ the IUWT of B_N and $X(x, y)$ the output of the convolution:

$$X(x, y) = W_N(x, y) * G(x, y) \quad (1)$$



Figure 3. The *cameraman* image.

An example of this filtering is shown in Figure 4 on a part of the *cameraman* image shown in Figure 3. In this example, the Gabor filter is positioned in the direction of the white bar of the tripod (around 75°). The output (Figure 4(b)) shows that mainly edges having the same direction as the bar, are detected.

3.2 Edge-directed Gaussian Filtering

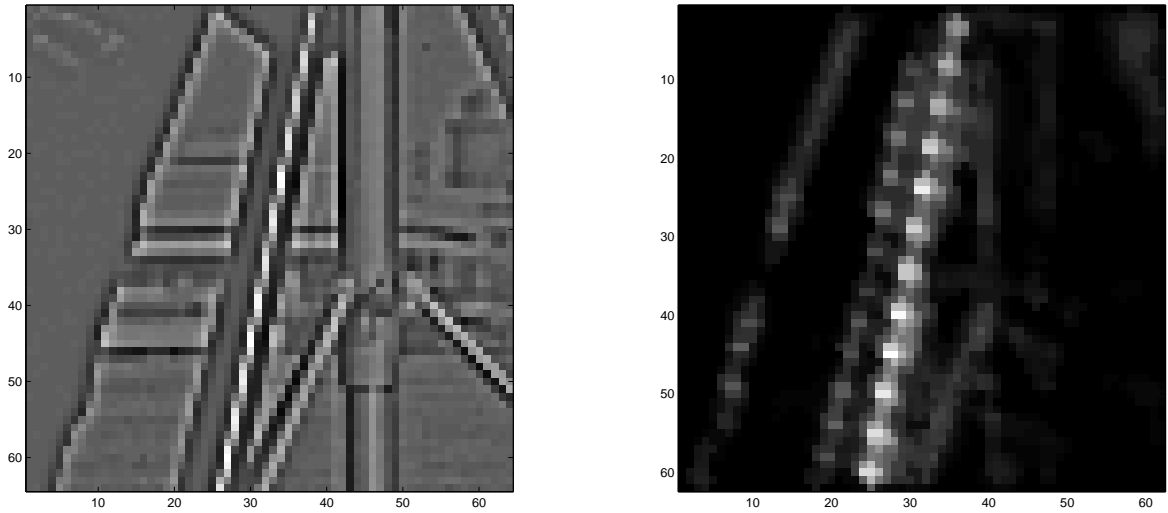
Cubic-spline interpolation is often considered as one of the best low complexity general purpose interpolator. This is due to its acceptable behavior on most natural images and the introduction of relatively few visually disturbing artefacts, and this is the reason why we use it as the reference interpolation. However the restricted size and the square shape of the two-dimensional interpolation support fail to correctly interpolate oriented edges.

Next paragraph shows that directional edges can be reconstructed *a posteriori* by filtering pixels in the direction of the edge. By doing so, the filter connects pixels that contain the edge direction information and thus gives a nice edge aspect.

3.2.1 Construction of the filter

The filter we build is a two-dimensional gaussian function oriented at θ , the detected direction of the edge. The two other parameters of the gaussian σ_θ and σ_{θ^\perp} are tuned such as the gaussian is stretched along θ and very thin in the perpendicular direction θ^\perp : $\sigma_\theta \gg \sigma_{\theta^\perp}$. The reason why the filter is shaped this way is that we want to avoid blur which would be introduced by filtering pixels across the edges. Several filters are represented in Figure 5 for various values of σ_{θ^\perp} .

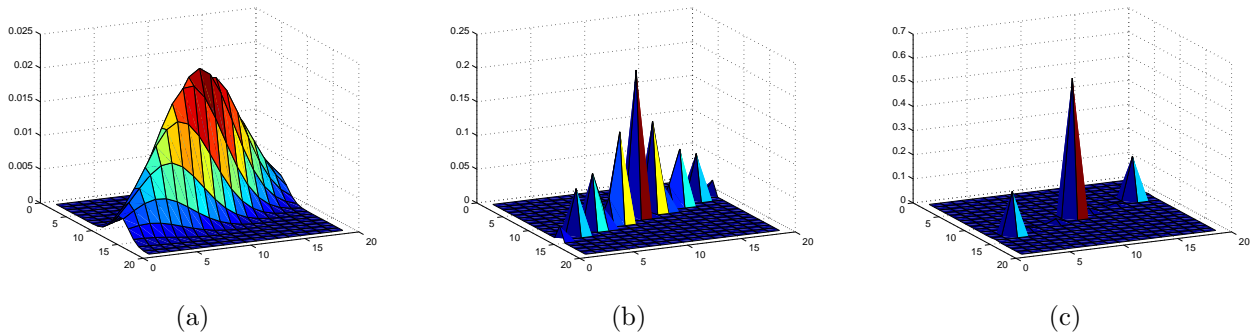
In the discrete space, the concept of orientation is different from the continuous space. Oriented lines can be represented with a parametric formulation:¹³ $D(a, b, \mu, \omega)$, where μ is the shift, ω the thickness and (a, b) are



(a)

(b)

Figure 4. (a) Finest scale of the IUWT. (b) Output of the oriented Gabor filter that will be used as a mask to balance the directional and the isotropic interpolation.



(a)

(b)

(c)

Figure 5. Gaussian filters for different values of the parameter $\sigma_{\theta\perp}$. (a) $\sigma_{\theta\perp} = 1.5$. (b) $\sigma_{\theta\perp} = 0.15$. (c) $\sigma_{\theta\perp} = 0.05$.

co-prime such as $\arctan\left(\frac{a}{b}\right) = \theta$ with θ the angle of the line in radian. Discrete lines with different values of thickness are shown in Figure 6. The thinnest discrete line (Figure 6(c)) is the most precise discrete representation of an angle, in the sense that two consecutive pixels lie exactly in the direction of the line. Thus, all pixels are the same distance away from each other. This distance is given in the cartesian plane by the coordinates (a, b) as previously defined. We can notice that when a continuous straight line is sampled to be displayed, pixels that are (a, b) away from each other have the exact same gray level values. These pixels have to be considered together in order to interpolate a straight line without losing the orientation information. We use this principle in the Gaussian directional filtering.

It is interesting to note that when $\sigma_{\theta\perp}$ decreases, the gaussian filter behaves in a similar way as the discrete line when ω decreases (Figure 5). For small values of $\sigma_{\theta\perp}$, the filter is composed of coefficients that lie on a thin discrete line oriented at θ . Consequently, pixels that belong to this line will be filtered together, which will retrieve the initial direction information lost during the cubic-spline interpolation. The thinner the filter, the better the precision on the angle value. However, some extreme orientations require a very large support to draw the corresponding thinnest discrete line. A compromise has to be found on the thickness of the filter to obtain reasonable filter sizes.

In our case, $\sigma_{\theta\perp}$ is set to 0.1 (between Figure 5(b) and (c)) and σ_{θ} is set to 10. This value keeps the filter thin enough to make sure that the gaussian does not introduce blur and that the direction information is retrieved.

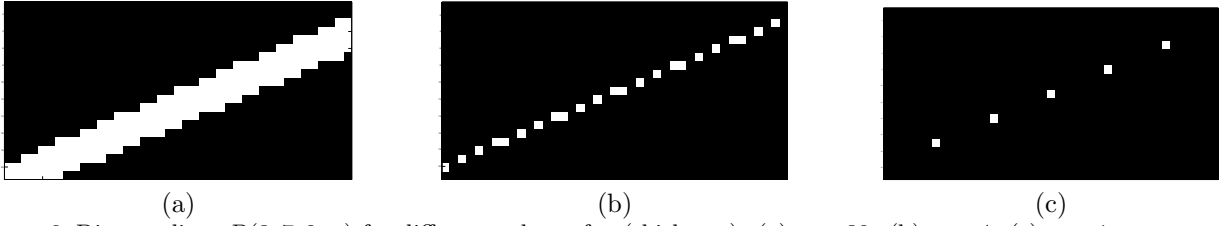


Figure 6. Discrete lines $D(3, 7, 0, \omega)$ for different values of ω (thickness). (a) $\omega = 30$. (b) $\omega = 4$. (c) $\omega = 1$ represents the thinnest discrete line.

On the other hand, it is big enough to ensure that the gaussian filter has at least some significant values for all directions, in a relatively small neighborhood.

In practice, the filter is truncated to keep only the significant coefficients closest to the center. These are enough to reconstruct the direction information, and increasing the filter size would introduce some artefacts (mainly echo and blur on non-edge parts). The result of such a filtering on the image shown in Figure 7. In this example, the filter is set in the direction of the central tripod bar of the camera. It is clear that edges that have this direction are well reconstructed. On the other hand, the background and other edges are highly affected by the filtering. To avoid these deteriorations, we use the mask defined in part 3.1 that will balance the directional interpolation and the isotropic interpolation.

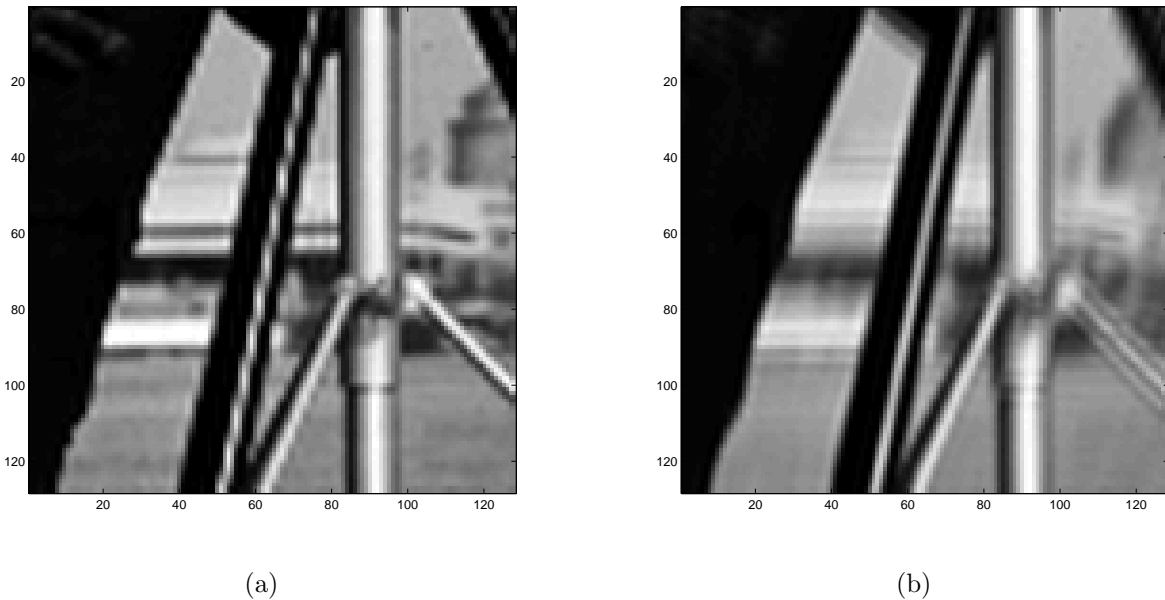


Figure 7. Example of directional filtering on a 2×2 cubic-spline interpolation. (a) Isotropic (cubic-spline) interpolation. (b) Directional interpolation.

3.3 Implementation Steps

- Build the quad-tree in order to isolate in each block at most one edge direction.

Then in each block of the quad-tree:

- Build the mask $X(i, j)$ using a Gabor filtering that detects the correctly oriented edges. $\bar{X}(x, y)$ is the normalized and interpolated version of $X(i, j)$ that fits the size of the interpolated block.

- Interpolate the block using the cubic-spline interpolation. Let $B_{isotropic}(x, y)$ be the isotropic interpolated block.
- Filter $B_{isotropic}(x, y)$ in the detected direction with a gaussian filter. Let $B_{dir}(x, y)$ be the directionally interpolated block.
- Combine the two interpolations using the mask to create the final interpolation of the block: $B_{interp}(x, y)$.

$$B_{interp}(x, y) = B_{dir}(x, y) \times \bar{X}(x, y) + B_{isotropic}(x, y) \times (1 - \bar{X}(x, y)) \quad (2)$$

In order to avoid blocking and border effects, an overlap between blocks is introduced in the Gabor and gaussian filterings.

4. RESULTS

Figure 8 (d) shows a close-up of a 4×4 (vertical and horizontal) interpolation performed with our method on the same part of the *cameraman* image as before. Comparison with a cubic-spline interpolation (Figure 8 (a)), with the the Edge Adapted interpolation⁹ (Figure 8 (b)), and with the NEDI⁸ (Figure 8 (c)) are shown. The Edge Adapted method is based on a pre-defined directional map that estimates the edge orientation and uses this information to balance the weights of the neighboring pixels during the interpolation. The NEDI is based on the geometric duality that exists between the low-resolution covariance and the high-resolution covariance. This propriety is used to adapt the interpolation at a higher-resolution.

It can be seen that Figures 8 (a) and (b) have jaggy edges, especially on the straight edges of the camera tripod. On the other hand, figures (c) and (d) have a nice rendering of these edges, showing the impact of an efficient edge directed interpolation.

However, the NEDI generates unnatural edges both in the background and in textured parts. On the opposite, our method keeps the correct aspect of the texture and background given by the cubic-spline interpolation, while improving the visual aspect of edges. This improvement is due to the local adaptation of our interpolation, guided by the Gabor filtering which efficiently balances the effect of the edge directed interpolation and of the cubic-spline interpolation. This local adaptation makes our method efficient for all kind of images.

An other example is shown on a part of the *Lighthouse* image shown in Figure 9. The results are shown in Figure 10, where we can observe that our method is the only one that reconstructs the direction of the lines properly. Indeed, the support of our interpolation is stretched in the direction of the edge in order to capture the pixels that contain the orientation information, something that cannot be done with limited interpolation support methods.

5. CONCLUSION

We have presented an interpolation method based on the combination of a isotropic and a directional interpolation that is adapted to most image contents. Unlike most directional interpolation techniques, it keeps the aspect of textures and edge-free areas and gives visually pleasing results. The Gabor masking makes sure that the directional interpolation is only applied on edges having the selected orientation, which allows us to take the risk of filtering pixels that are relatively far from each other without creating artefacts. Thus, edges of most orientations can be reconstructed in the interpolated image. On the downside, our method takes longer time to interpolate images than classical interpolation methods. This is mainly due to both Gabor and gaussian filterings that have to be done in each block of the quad-tree. However, compromises on the angle precision can be made and look-up tables containing pre-defined filter coefficients can be built, to bring the computation time back to more classical values. Finally, it can also be noticed that no additional tuning of parameter is required, which makes this interpolation straight forward to apply.

REFERENCES

- [1] Van Reeth, E., Bertolino, P., Nicolas, M., and Chassery, J. M., "Adaptive edge orientation analysis," *Wavelet Applications in Industrial Processing VII* **7535**(1), 753505+, SPIE (2010).

- [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] Perona, P., "Orientation diffusions," *Image Processing, IEEE Transactions on* **7**(3), 457–467 (1998).
- [4] Jafari Khouzani, K. and Soltanian Zadeh, H., "Radon transform orientation estimation for rotation invariant texture analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**, 1004–1008 (2005).
- [5] Michelet, F., Costa, J.-P. D., Laviaille, O., Berthoumieu, Y., Baylou, P., and Germain, C., "Estimating local multiple orientations," *Signal Processing* **87**, 1655–1669 (2007).
- [6] Bigun, J. and Hans du Buf, J., "Geometric image primitives by complex moments in gabor space and the application to texture segmentation," in [*IEEE Computer Society Conference on Computer Vision and Pattern Recognition*], 648 –650 (15-18 1992).
- [7] Peyre, G. and Mallat, S., "Discrete bandelets with geometric orthogonal filters," *ICIP* **1**, I – 65–8 (sep. 2005).



(a) 4× Cubic-spline interpolation.



(b) 4× Edge Adapted interpolation.



(c) 4× NED Interpolation.



(d) Our 4× interpolation.

Figure 8. Comparison of interpolation results. See electronic version for accurate comparison.

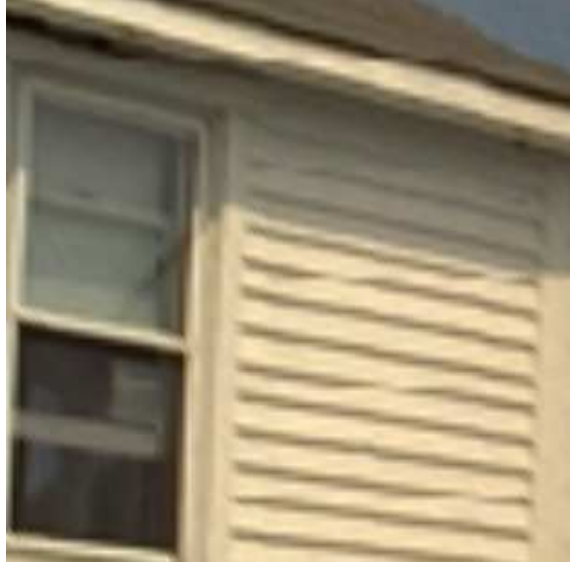


Figure 9. The *lighthouse* image.

- [8] Li, X. and Orchard, M., “New edge-directed interpolation,” *ICIP* **10**(10), 1521–1527 (2001).
- [9] Jiang, H. and Moloney, C., “A new direction adaptive scheme for image interpolation,” *ICIP* **3**, III-369 – III-372 vol.3 (2002).
- [10] Muresan, D., “Fast edge directed polynomial interpolation,” *ICIP* **2**, II- 990–993 (2005).
- [11] Wang, Q. and Ward, R., “A new orientation-adaptive interpolation method,” *ICIP* **16**, 889 –900 (april 2007).
- [12] Starck, J., Fadili, J., and Murtagh, F., “The undecimated wavelet decomposition and its reconstruction,” *ICIP* **16**, 297–309 (Feb. 2007).
- [13] Reveilles, J.-P., *Gomtrie discrete, Calcul en nombres entiers et algorithmique*, PhD thesis, Universit Louis Pasteur, Strasbourg, France (1991).



(a) 4× Cubic-spline interpolation.



(b) 4× Edge Adapted interpolation.



(c) 4× NED Interpolation.



(d) Our 4× interpolation.

Figure 10. Comparison of interpolation results. See electronic version for better visualization.