MASTER COPY – FOR REPRODUCTION PURPOSES

AD A218 857

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION Center of Excellence in AI University of Pennsylvania	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION U. S. Army Research Office
6c. ADDRESS (City, State, and ZIP Code) Dept. of Computer & Information Science 200 S. 33rd Street Philadelphia, PA 19104-6389		7b. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211

8a. NAME OF FUNDING / SPONSORING ORGANIZATION U. S. Army Research Office	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211	10. SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)

Image Wavelet Decomposition and Applications (MS-CIS-89-22)

12. PERSONAL AUTHOR(S)

N. Treil, S. Mallat and R. Bajcsy

13a. TYPE OF REPORT Interim technical	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) Revised December 1989	15. PAGE COUNT 8

16. SUPPLEMENTARY NOTATION The view, opinions and/or findings contained in this report are those
of the author(s) and should not be construed as an official Department of the Army position,
policy, or decision, unless so designated by other documentation.

17.	COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Computer vision, image processing

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The general problem of computer vision has been investigated for more than twenty years and is still one of the most challenging fields in artificial intelligence. Indeed taking a look at the human visual system can give us an idea of the complexity of any solution to the problem of visual recognition. This general task can be decomposed into a whole hierarchy of problems ranging from pixel processing to high level segmentation and complex objects recognition.

Contrasting an image at different representations provides useful information such as edges. First, we introduce an example of low level signal and image processing using the theory of wavelets which provides the basis for multiresolution representation. Like the human brain, we use a multiorientation process which detects features independently in different orientation sectors. So, we end up contrasting images of the same orientation but of different resolutions to gather information about an image.

We then develop an interesting image representation using energy zero crossings. We show that this representation is experimentally complete and leads to some higher level applications such as edge and corner finding, which in turn provides two basic steps to image segmentation. We also discuss the possibilities of feedback between different levels of processing.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR   83 APR edition may be used until exhausted.
All other editions are obsolete.
SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

UNIVERSITY of PENNSYLVANIA

Image Wavelet Decomposition
and Applications

N. Treil, S. Mallat
R. Bajcsy

MS-CIS-89-22
GRASP LAB 177

Department of Computer and Information Science
School of Engineering and Applied Science
Philadelphia, PA 19104-6389

PENN

Image Wavelet Decomposition
and Applications

N. Treil, S. Mallat
R. Bajcsy

MS-CIS-89-22
GRASP LAB 177

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

April 1989

ACKNOWLEDGEMENTS.

Abstract

The general problem of computer vision has been investigated for more than twenty years and is still one of the most challenging fields in artificial intelligence. Indeed taking a look at the human visual system can give us an idea of the complexity of any solution to the problem of visual recognition. This general task can be decomposed into a whole hierarchy of problems ranging from pixel processing to high level segmentation and complex objects recognition.

Contrasting an image at different representations provides useful information such as edges. First, we introduce an example of low level signal and image processing using the theory of wavelets which provides the basis for multiresolution representation. Like the human brain, we use a multiorientation process which detects features independently in different orientation sectors. So, we end up contrasting images of the same orientation but of different resolutions to gather information about an image.

We then develop an interesting image representation using energy zero crossings. We show that this representation is experimentally complete and leads to some higher level applications such as edge and corner finding, which in turn provides two basic steps to image segmentation. We also discuss the possibilities of feedback between different levels of processing.

Contents

1 Introduction.

The general problem of computer vision has been investigated for more than twenty years and is still one of the most challenging fields in artificial intelligence. Just by taking a look at the human visual system can give us an idea of the complexity of any solution to the problem of visual recognition. This general task can be decomposed into a whole hierarchy of problems ranging from pixel processing to high level segmentation and complex objects recognition. First, we introduce an example of low level signal and image processing using the theory of wavelets and then develop interesting image representation. We show that this representation is experimentally complete and leads to some higher level applications such as edge and corner finding. This, in turn, provides two basic steps to image segmentation. We also discuss the possibilities of feedback between different levels of processing.

2 The multiresolution concept.

2.1 Motivation.

The information provided by an image lies in the local variations of the image intensity. From the image contrast, we can extract some important features such as the edges of the structures embedded in the image. At the border of such structures, the image intensity is likely to suddenly change. The local variations have therefore a high amplitude.

However not all of the local variations have the same relevance to the understanding of a scene. For example, suppose that we are looking at a far away house. Moving closer to it would make us distinguish successively the doors and windows, then the bricks of the walls and the tiles, then the texture of these bricks and tiles. Separating the details appearing at each resolution would enable us to establish a hierarchy between these pieces of information. It would also allow a scale-invariant interpretation of the image. This scale invariance is a well known property of the human visual system.

As it is not possible to define a priori an optimal resolution for analyzing images, several researchers have developed pattern matching algorithms which process the image at different resolutions. In some sense, image's details at a coarse resolution provide the "context" of the image which helps in the analysis of finer details. For example, it is difficult to recognize that a small rectangle inside an image is the window of a house if we did not previously recognize the house "context". It is therefore natural to first analyze the image details at a coarse resolution and then increase the resolution using a coarse to fine processing strategy. Another interesting approach is to study details' propagation through different resolutions. A feature which appears at several resolutions is certainly very noticeable in the image and is likely to be part of an object's contour. A feature appearing only at finer resolutions is a

high frequency variation of small amplitude. It probably corresponds to some texture. On the other hand, a feature which appears only at coarse resolutions is a smooth variation like shading.

2.2 The theory of wavelets.

Recently Stephane Mallat has introduced such a multiresolution representation of signals based on the theory of wavelets [1], using a finite set of resolutions namely the powers of 2 (a definite advantage for implementation). The basic idea is to separate the higher half and the lower half of the spectrum of a signal by using a second order band-pass filter and a low pass filter, to subsample the image corresponding to the lower half of the spectrum and to iterate the process. This is roughly equivalent to dividing the spectrum in successive bands $[\frac{\pi}{2}, \pi]$, $[\frac{\pi}{4}, \frac{\pi}{2}]$, $[\frac{\pi}{8}, \frac{\pi}{4}]$ ... $[\frac{\pi}{2^{i+1}}, \frac{\pi}{2^i}]$ and extracting independently the details corresponding to these bands. Suppose that our original signal has 2^j samples. It is then said to be at resolution 2^j. The result of the first band-pass filtering will give us the difference of information between resolution 2^j and resolution 2^{j-1}. The result of the next band-pass filtering will give us the difference of information between resolution 2^{j-1} and resolution 2^{j-2} and so on. The theory of wavelets shows that we can obtain filters which are very well localized both in the Fourier and spatial domain —thus preserving the locality of information— and that we can achieve a representation which is complete.

In order to obtain a representation of the signal which translates when the signal translate, Stephane Mallat[2] has designed another type of representation of the band-pass filtered images based on their zero-crossings and the energies between consecutive zero-crossings. It is called an Energy-zero-crossings representation. This representation has proved to be experimentally complete for signals [1] and in this article we also show that it is complete for images. It is the basis of the higher level applications developed in this article.

3 The multiorientation concept.

The purpose of a multiorientation process is to detect features independently in different orientation sectors. Although this increases the amount of computation, it also structures the data organization of the feature detection processes. Let us take a simple example. Suppose we carry out isotropic edge detection on an image which contains a rectangle. It would be pretty difficult to recognize that the closed region we have detected is actually a rectangle; one obstacle among others is that most edge detectors smooth corners[3]. If on the other hand we detect horizontal edges and vertical edges independently, and then use a composition process to find closed regions from the interactions between these edges, we will define an object with two horizontal sides two vertical sides and four corners ... obviously a rectangle ! In both case the result of the analysis is a closed region but by using multiorientation we

have learned something about the shape of the region while building it. Of course this is a very simplistic example and the composition processes are far from being straightforward, but this gives a hint on the idea of any image analysis method based on decomposition: gaining knowledge while recomposing. We can of course think of gaining knowledge about the organization of a scene by using a composition process on multiresolution as well.

It is interesting to note that multiorientation is one of the features of the human vision system[4]. Indeed there exist brain cells which respond specifically to stimuli within a certain orientation and also at a certain frequency. This means that the brain performs a multiresolution multiorientation decomposition of the visual input. There are as many as 30 main directions, where the divisions are finer around the horizontal and vertical axes. There also exist cells which respond specifically to corners. We can guess from this that the brain performs some composition processing as part of the whole visual recognition system.

4 The decomposition process.

In this section we describe in detail the decomposition process leading to the Energy-zero-crossings representation. Our purpose is not to explain the theory of wavelets. The choice of the filters as well as certain properties like the completeness of the representation have very sound mathematical bases which will not be discused here.

4.1 Monodimensional dyadic wavelet decomposition.

Let S^j be a signal at resolution 2^j and W^k be the signal of the differences between S^k and S^{k-1} for k less than or equal to j. Let $U_{\frac{\pi}{2}}$ be the low-pass filter with a cutoff frequency of $\frac{\pi}{2}$ and $G_{\frac{\pi}{2}}$ be the high-pass filter with a cutoff frequency of $\frac{\pi}{2}$. From the general idea of multiresolution, a simple processing would be:

1. Filter S^j by $G_{\frac{\pi}{2}}$. The result is W^j

2. Filter S^j by $U_{\frac{\pi}{2}}$ and sample by 2. The result is S^{j-1}

3. Filter S^{j-1} by $G_{\frac{\pi}{2}}$. The result is W^{j-1}

4. Filter S^{j-1} by $U_{\frac{\pi}{2}}$ and sample by 2. The result is S^{j-2} and so on...

This process is summarized in Figure 1.

$$S^j \longrightarrow \boxed{G_{\frac{\pi}{2}}} \longrightarrow W^j$$

$$\boxed{U_{\frac{\pi}{2}}}$$
$$\boxed{\downarrow 2}$$

$$S^{j-1} \longrightarrow \boxed{G_{\frac{\pi}{2}}} \longrightarrow W^{j-1}$$

$$\boxed{U_{\frac{\pi}{2}}}$$
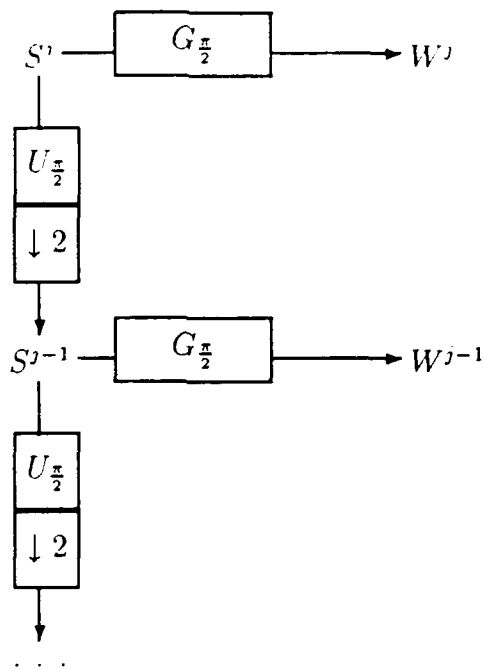$$\boxed{\downarrow 2}$$

$$\cdots$$

Figure 1: *Conceptual process.*

The final goal of the decomposition is to obtain zero-crossings which point out some information to us. *In other word we would like them to correspond to abrupt changes in the signal intensity.* The energy associated with these zero-crossings would provide a measurement of the abruptness of the variation. The other constraint is that we want to build a multiresolution representation which is complete. This leads to two important features of the decomposition process:

- To achieve the completeness of the Energy-zero-crossings representation, we need a good precision on the zero-crossings. Thus we need to **expand** the results of the consecutive band-pass filterings before computing the positions of the zeros. This is done by performing a spline interpolation.

- The filters have to fit the requirements of the theory of wavelets. After computation they appear to perform a rather smooth cutoff. Thus we cannot sample by two the result of the low-pass filtering without loosing information[5]. A solution to this problem is to wait for several iterations of the process[1] before starting the actual sampling.

From what we have just said, supposing that we want to expand the $(W^i)_{i \in [j-n...j]}$ by a factor 4 (typical expansion factor) the algorithm becomes:

1. Filter S^j by $G_{\frac{\pi}{2}}$. The result is W^j

---

[1] We call **iteration** the list of operations yielding the isolation of the spectrum part between $\frac{\pi}{2^{j+1}}$ and $\frac{\pi}{2^j}$

2. Expand W^j by a factor 4

3. Filter S^j by $U_{\frac{\pi}{2}}$. The result is S^{j-1} (but it still has 2^j samples !)

4. Filter S^{j-1} by $G_{\frac{\pi}{4}}$. The result is W^{j-1}

5. Expand W^{j-1} by a factor 2

6. Filter S^{j-1} by $U_{\frac{\pi}{4}}$. The result is S^{j-2}

7. Filter S^{j-2} by $G_{\frac{\pi}{8}}$. The result is W^{j-2}

8. Filter S^{j-2} by $U_{\frac{\pi}{8}}$ and sample by 2. The result is S^{j-3}

9. Filter S^{j-3} by $G_{\frac{\pi}{8}}$. The result is W^{j-3}

10. Filter S^{j-3} by $U_{\frac{\pi}{8}}$ and sample by 2. The result is S^{j-4} and so on...

This is summarized in Figure 2. In Appendix A we give the algorithm for an expansion factor of 2 instead of 4.
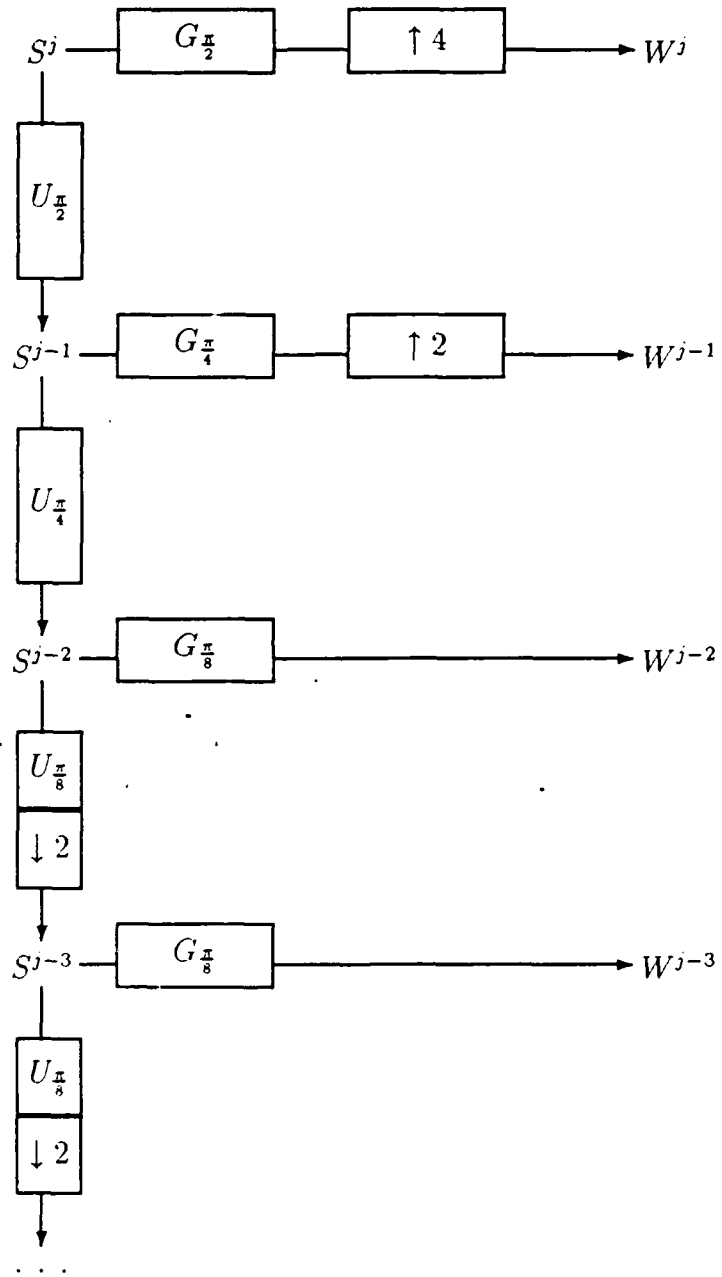
Figure 2: *Real process.*

As we can now see, we need two other kinds of filters with a cutoff frequency of $\frac{\pi}{4}$ and $\frac{\pi}{8}$ respectively. One way to handle this is to periodize by 2 and 4 the filters in the Fourier domain (i.e. inserting zeros between coefficients in the spatial domain[5]). Indeed if we suppose that we have a low pass filter $U_{\frac{\pi}{2}}$ with a cut off frequency of $\frac{\pi}{2}$ as shown in Figure 3.
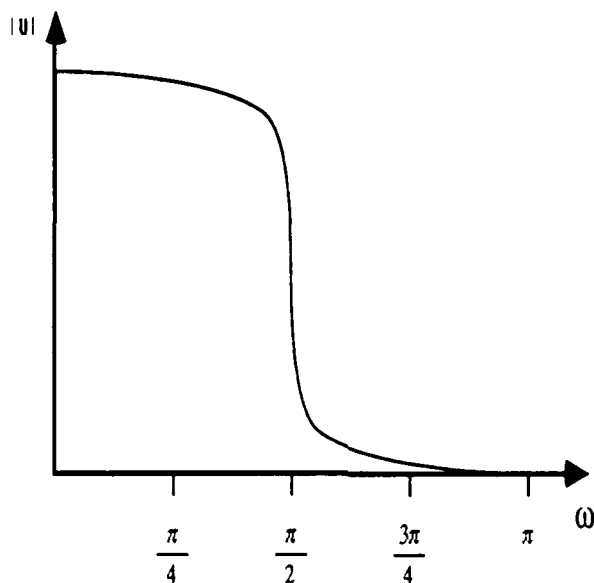

Figure 3: *Low pass filter* $U_{\frac{\pi}{2}}$.

If we periodize this filter by two as shown in Figure 4 we obtain a filter $U_{\frac{\pi}{4},\frac{3\pi}{4}}$ which behaves as the combination of a low-pass filter with a cut off frequency of $\frac{\pi}{4}$ and a high-pass filter with a cut off frequency of $\frac{3\pi}{4}$. However we use this filter after having already filtered the signal by $U_{\frac{\pi}{2}}$. The spectrum of the signal is then nearly null for frequencies higher than $\frac{\pi}{2}$ so the filter $U_{\frac{\pi}{4},\frac{3\pi}{4}}$ really behaves as a low-pass filter as its high-pass capabilities are applied to the null part of the spectrum.

This periodization technique reveals quite efficient and the distortion which might occur —due to the fact that our filters are far from being ideal— is negligible.

We can now compute the zero-crossings and energies on the $(W^i)_{i\in[j-n..j]}$. The integration method we used for energy computation is based on the spline interpolation[6] and is exact for cubic spline functions.
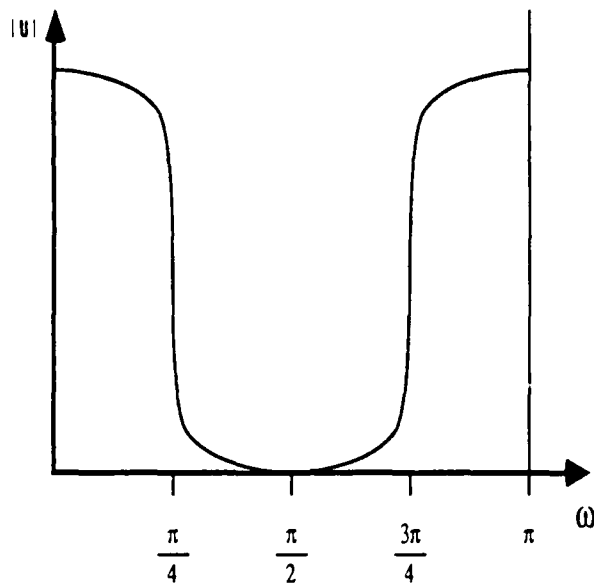
Figure 4: *Low pass filter $U_{\frac{\pi}{2}}$ periodized by 2.*

4.2 Bidimensional dyadic wavelet decomposition.
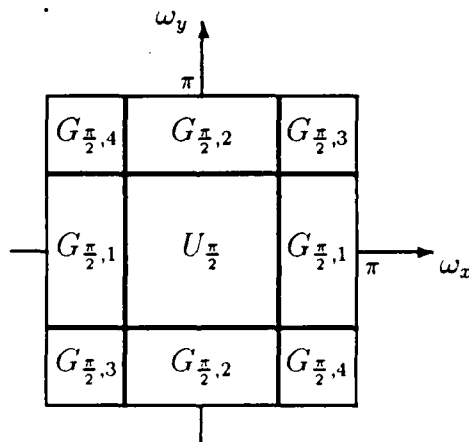
Figure 5: *Splitting of the spectrum using wavelet filters.*

The bidimensional multiresolution multiorientation decomposition process we implemented reveals to be much more complex. First of all, the square shape of pixels limits immediately the number of possible orientations one can handle without encountering tremendous implementation problems. Thus we chose the four main directions: horizontal, vertical and the two separate diagonals. This means that for each resolution we use four distinct directional high pass filters and a low pass filter. In terms of spectrum Figure 5 shows what parts of

10

the spectrum are isolated by each filter $G_{\frac{\pi}{2},1}$, $G_{\frac{\pi}{2},2}$, $G_{\frac{\pi}{2},3}$, $G_{\frac{\pi}{2},4}$ and $U_{\frac{\pi}{2}}$. The $\left(G_{\frac{\pi}{2},k}\right)_{k \in [1..4]}$ filters are high-pass filters along one direction and low-pass filters along the perpendicular direction. The zero-crossings will be computed on the corresponding filtered images along the high-pass filtering direction. In order to save time and memory space, the filtered images will also be sampled along the low pass filtering direction. Notice that the $G_{\frac{\pi}{2},1}$, $G_{\frac{\pi}{2},2}$ and $U_{\frac{\pi}{2}}$ filters are separable. The $G_{\frac{\pi}{2},1}$ filter can be obtained by filtering by the one dimensional filter $G_{\frac{\pi}{2}}$ along the lines and by the one dimensional filter $U_{\frac{\pi}{2}}$ along the columns. The $G_{\frac{\pi}{2},2}$ filter can be obtained by filtering by the one dimensional filter $U_{\frac{\pi}{2}}$ along the lines and by the one dimensional filter $G_{\frac{\pi}{2}}$ along the columns. The two dimensional filter $U_{\frac{\pi}{2}}$ can be obtained by filtering by the one dimensional filter $U_{\frac{\pi}{2}}$ along the lines and the columns.

Of course the two constraints mentioned in section 4.1 are still in effect. Thus we must expand the filtered images along the high-pass filtering direction before computing the zero-crossings, and we must also wait for several decomposition steps before starting to sample along the low-pass filtering direction. To these two constraints we must add a third one which is that our diagonal filters $G_{\frac{\pi}{2},3}$ and $G_{\frac{\pi}{2},4}$ are not separable. As a consequence, they must be of reasonable size. This is rendered quite complicated by the fact that they are not 2π periodic in both directions. Indeed let us consider Figure 6 which shows the main support[2] of the fourier transform of wavelet filter $G_{\frac{\pi}{2},3}$.
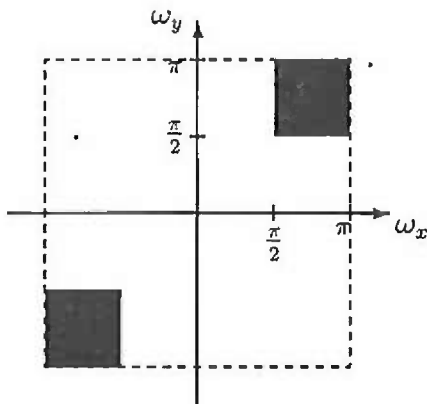
Figure 6: *Diagonal filter theoretical support.*

If we mentally periodize the pattern with a period of 2π along both axes we realize easily that there are some points of abrupt changes on the border of the dashed square. This means that if we perform a reverse discrete Fourier transform on this filter in order to define it in the spatial domain, the coefficients we obtain do not decrease rapidly at all (like those of a cardinal sinus). Using this kind of filter would lead to a very expensive convolution. An alternative to this is to shrink the spectrum of the image to make it fit between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$

---

[2]Idealy, the filter should be 1 inside the area in black and 0 elsewhere. However our filter is smooth to reduce its number of coefficients and to fit the requirements of the theory of wavelets. So the area in black corresponds rather to the filter's **main support** which is the area of the Fourier space where its value is nearer to 1 then to 0.

on each axis, and to filter it with a filter whose Fourier transform's main support is given by Figure 7 thus isolating the correct part of the spectrum. Shrinking the spectrum can be done, in terms of spatial operation, by interpolating the image by a factor 2 using a spline interpolation[5]. The filter of Figure 7 doesn't have any discontinuities when you periodize the pattern with a period of 2π. As a consequence, it will have a small number of coefficients in the spatial domain. The coefficients of this filter are given in Appendix B.
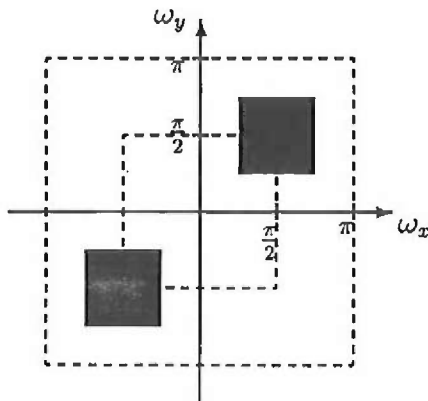

Figure 7: *Diagonal filter actual support.*

We have to fit these three requirements within the same algorithm. Supposing we want to expand the band-pass filtered images by 4 for zero-crossings computation the decomposition is made using the following steps:

Let S^j be an image at resolution 2^j and W_1^k W_2^k W_3^k and W_4^k be the image details at resolution 2^k with k less than or equal to j in the horizontal, vertical, first diagonal and second diagonal directions respectively. Let $\left(G_{\frac{\pi}{2^i},k}\right)_{k\in[1..4]}$ be the directional band-pass filters whose general shapes are given in Figure 5 but which isolate the parts of the spectrum between $\frac{\pi}{2^i}$ and $\frac{\pi}{2^{i-1}}$ (instead of between $\frac{\pi}{2}$ and π). Let $U_{\frac{\pi}{2^i}}$ the low-pass filter of cutoff frequency $\frac{\pi}{2^i}$.

1. Expand S^j by a factor 2 along both horizontal and vertical axes using spline interpolation. This will confine its spectrum between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. We will refer to the resulting image as S_e^j.

2. Filter S_e^j by the $\left(G_{\frac{\pi}{4},k}\right)_{k\in[1..4]}$ and sample by 2 along the low-pass direction . The results are the $\left(W_k^j\right)_{k\in[1..4]}$.

3. Expand the $\left(W_k^j\right)_{k\in[1..4]}$ by 2 along the band pass filtering direction using spline interpolation.

4. Filter S_e^j by the low-pass filter $U_{\frac{\pi}{4}}$. We call the result S^{j-1} (but it still has 2^{j+1} samples !).

12

5. Filter S^{j-1} by the $\left(G_{\frac{\pi}{8},k}\right)_{k\in[1..4]}$ and sample by 4 along the low-pass direction . The result is the $\left(W_k^{j-1}\right)_{k\in[1..4]}$.

6. Filter S^{j-1} by the low-pass filter $U_{\frac{\pi}{8}}$ and sample by 2 along each axis . The result is S^{j-2}.

7. Filter S^{j-2} by the $\left(G_{\frac{\pi}{8},k}\right)_{k\in[1..4]}$ and sample by 4 along the low-pass direction . The result is the $\left(W_k^{j-2}\right)_{k\in[1..4]}$.

8. Filter S^{j-2} by the low-pass filter $U_{\frac{\pi}{8}}$ and sample by 2 along each axis . The result is S^{j-3} and so on ...

In Appendix A we give the algorithm for an expansion factor of 2 instead of 4.

We can now compute the Energy-zero-crossings representation on the set of detail images $(W_k^i)_{(i,k)\in[j-n..j]\times[1..4]}$ along the band-pass filtering directions. The integration we used method for energy computation is based on the spline interpolation and is exact for cubic spline functions.

4.3   Energy-zero-crossings representation.

Let us consider the monodimensional case first. building an energy-zero-crossings representation from a dyadic wavelet representation consists in coding the detail signals not in terms of pixels but in term of zero-crossings and energies between adjacent zero-crossings. Indeed if the filters are properly chosen so as to make them proportional to the second derivative of a smoothing function, the energy-zero-crossings representation directly points out to us where the information is, and gives us a measure of the information's importance. The two components of this representation are dealt with at different levels. First the detail signal is transformed into a chained list of **energy patches** which are a representation of the energy of each arch[3] of the signal. The information retained for each energy patch is the following:

- let S be the signal and x_0 and x_1 two adjacent zero-crossings.

- let e be the energy of the arch between x_0 and x_1:

$$e = \sqrt{\int\int_{x_0}^{x_1}[S(x)]^2 dx}$$

- let ε be the sign of S between x_0 and x_1.

---

[3]We call **arch** the portion of the signal between adjacent zero-crossings.

- let l the arch length defined by:

$$l = x_1 - x_0$$

- let a be defined by:

$$al = \varepsilon e^2$$

- then a and l are the characteristic parameter describing the energy patch.

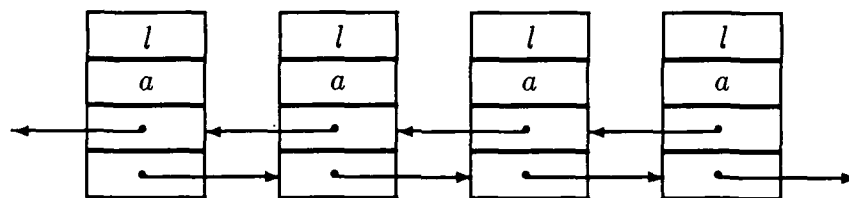Then the signal representation looks like:

Figure 8: *Energy list structure.*

Having defined what an energy patch is we can now take the following structure for the zero-crossings: A *zero-crossing is defined by its abscissa, the energy patch on its right and the energy patch on its left.* All this is summarized in figure 9.
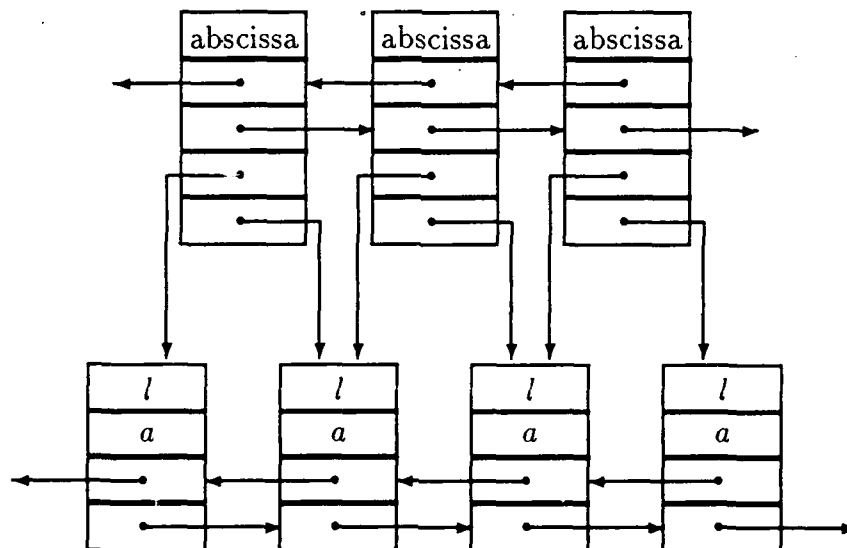
Figure 9: *Energy-zero-crossings structure.*

The two dimensional Energy-zero-crossings representation is a direct generalization on the one-dimensional case. Each detail image has a band-pass filtering direction and a low-pass filtering direction. We compute the energies and zero crossings along the band-pass filtering direction treating these lines as one dimensional signals. For the $(W_1^i)_{i \in [j-n..j]}$ the zero-crossings are computed along the lines, for the $(W_2^i)_{i \in [j-n..j]}$ they are computed along the

columns, and for the $(W_3^i)_{i\in[j-n..j]}$ (resp. $(W_4^i)_{i\in[j-n..j]}$) they are computed along the positive (resp. negative) diagonal[4]. Figure 10 shows the Energy-zero-crossings representation of the dyadic decomposition of the image of Figure 13 (section 5.2) for the horizontal and vertical orientations at the first three resolutions 2^{j-1}, 2^{j-2} and 2^{j-3}.

---

[4] We call **positive diagonal** the diagonal that goes from bottom left to top right, and **negative diagonal** the diagonal that goes from top left to bottom right
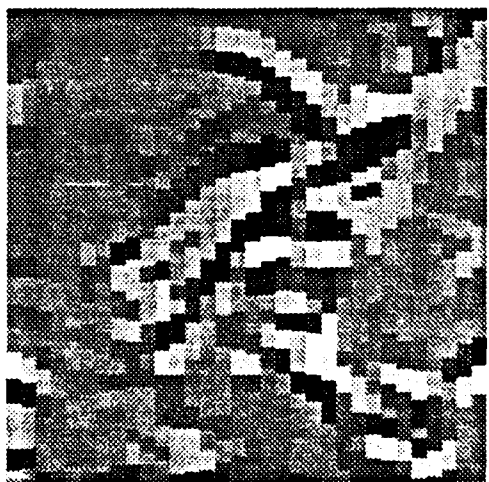
Figure 10: *Energy-Zero-Crossings representation (horizontal & vertical).*

5 Completeness of the representations.

5.1 Dyadic wavelet transform.

Stephane Mallat has previously demonstrated that the dyadic wavelet transform was complete for signals and images. However only the monodimensional simulation had been implemented. In this paper we show results concerning a bidimensional implementation of the dyadic reconstruction process. The reconstruction is based on the following property of the U and G filters:

$$|G(\omega)|^2 + |U(\omega)|^2 = 1$$

or, in two dimensions:

$$\sum_{i=1}^{4} |G_i(\omega_x, \omega_y)|^2 + |U(\omega_x, \omega_y)|^2 = 1$$

This means that if S^j is a signal and if we obtained S^{j-1} and W^j by filtering by U and G respectively, the original signal can be obtained by filtering S^{j-1} by U, by filtering W^j by G and by adding up the results. This is summarized in the following equation:

$$S^j = S^{j-1} * u + W^j * g$$

Where u and g are the impulse responses of filters U and G. In fact this is not exactly true. Indeed we must remember that we did certain operations of spline expansion and sampling to obtain S^{j-1} and W^j. Thus we must reverse these operations . The reverse of a spline interpolation can be performed using a reduction/projection as shown by Stephane Mallat[1], and the reverse of a decimation is an spline interpolation[5].

5.1.1 One dimensional algorithm.

We can now give the one dimensional reconstruction process. Supposing that we have decomposed signal S^j as described in Figure 2 and obtained W^j, W^{j-1}, W^{j-2}, W^{j-3} and S^{j-4} we can reconstruct S^j this way:

1. Expand S^{j-4} by a factor 2. The result is S^{j-4}_{exp}.

2. Filter S^{j-4}_{exp} by $U_{\frac{\pi}{8}}$.

3. Filter W^{j-3} by $G_{\frac{\pi}{8}}$.

4. Add up the results. This gives S^{j-3}.

5. Expand S^{j-3} by a factor 2. The result is S^{j-3}_{exp}.

6. Filter S^{j-3}_{exp} by $U_{\frac{\pi}{8}}$.

7. Filter W^{j-2} by $G_{\frac{\pi}{8}}$.

8. Add up the results. This gives S^{j-2}.

9. Filter S^{j-2} by $U_{\frac{\pi}{4}}$.

10. Reduce W^{j-1} by a factor 2. The result is W^{j-1}_{red}

11. Filter W^{j-1}_{red} by $G_{\frac{\pi}{4}}$.

12. Add up the results. This gives S^{j-1}.

13. Filter S^{j-1} by $U_{\frac{\pi}{2}}$.

14. Reduce W^{j} by a factor 4. The result is W^{j}_{red}

15. Filter W^{j}_{red} by $G_{\frac{\pi}{2}}$.

16. Add up the results. This gives S^{j}.

5.1.2 Two dimensional algorithm.

For images, the algorithm is a little more complicated. The major obstacle to reconstruction is the square grid. Let us first describe the basic steps. Supposing that we have decomposed signal S^j and obtained the $\left(W^{j}_{k}\right)_{k\in[1..4]}$, $\left(W^{j-1}_{k}\right)_{k\in[1..4]}$, $\left(W^{j-2}_{k}\right)_{k\in[1..4]}$ and S^{j-3} we can reconstruct S^j this way:

1. expand S^{j-3} by 2 along each axis. The result is S^{j-3}_{exp}.

2. Expand the $\left(W^{j-2}_{k}\right)_{k\in[1..4]}$ by 4 along the low-pass filtering direction. The results are the $\left(W_{exp\ k}^{\ j-2}\right)_{k\in[1..4]}$.

3. Filter the $\left(W_{exp\ k}^{\ j-2}\right)_{k\in[1..4]}$ by the $\left(G_{\frac{\pi}{8},k}\right)_{k\in[1..4]}$ and S^{j-3}_{exp} by $U_{\frac{\pi}{8}}$.

4. Add up the results. This gives S^{j-2}.

5. expand S^{j-2} by 2 along each axis. The result is S^{j-2}_{exp}.

6. Expand the $\left(W^{j-1}_k\right)_{k\in[1..4]}$ by 4 along the low-pass filtering direction. The results are the $\left(W_{exp}{}^{j-1}_k\right)_{k\in[1..4]}$.

7. Filter the $\left(W_{exp}{}^{j-1}_k\right)_{k\in[1..4]}$ by the $\left(G_{\frac{\pi}{8},k}\right)_{k\in[1..4]}$ and S^{j-2}_{exp} by $U_{\frac{\pi}{8}}$.

8. Add up the results. this gives S^{j-1}.

9. Reduce the $\left(W^{j}_k\right)_{k\in[1..4]}$ by 2 along the band-pass filtering direction and expand them by 2 along the low-pass filtering direction. The results are the $\left(W_{exp}{}^{j}_k\right)_{k\in[1..4]}$.

10. Filter the $\left(W_{exp}{}^{j}_k\right)_{k\in[1..4]}$ by the $\left(G_{\frac{\pi}{4},k}\right)_{k\in[1..4]}$ and S^{j-1} by $U_{\frac{\pi}{4}}$.

11. Add up the results. this gives S^j_e, which is S^j expanded by a factor 2.

12. Reduce S^j_e by 2 along each axis. This gives S^j.

The critical steps here are steps 2, 6 and 9 for the images corresponding to the two diagonal directions. These images have been obtained by sampling along the diagonal lines. Due to the square shape of the pixels, it is very difficult to interpolate again along the diagonal lines to reverse the sampling. The problem is that we have an array with only certain diagonals containing pixels, and we want to fill in the empty pixels. Let us consider an image sampled by four along the "negative" diagonal[5].

The process we have chosen is to interpolate by four along the negative diagonal, as shown in Figure 12 (notice that the new pixels do not fit the square grid) and to fill in the remaining empty pixels by linear interpolation along the positive diagonal. For example in Figure 12, value at pixel "c" would be equal to the mean of value at pixel "a" and value at pixel "b".

Of course this interpolation is not at all accurate since we perform a linear interpolation —which is a rather poor interpolation— along the direction containing the high frequencies. However, this error appears only on the diagonal lines, for which the human visual system has less sensitivity. Thus it is very difficult to detect this error except when looking at the image of the differences between the original picture and the reconstructed picture.

---

[5] We call **positive diagonal** the diagonal that goes from bottom left to top right, and **negative diagonal** the diagonal that goes from top left to bottom right
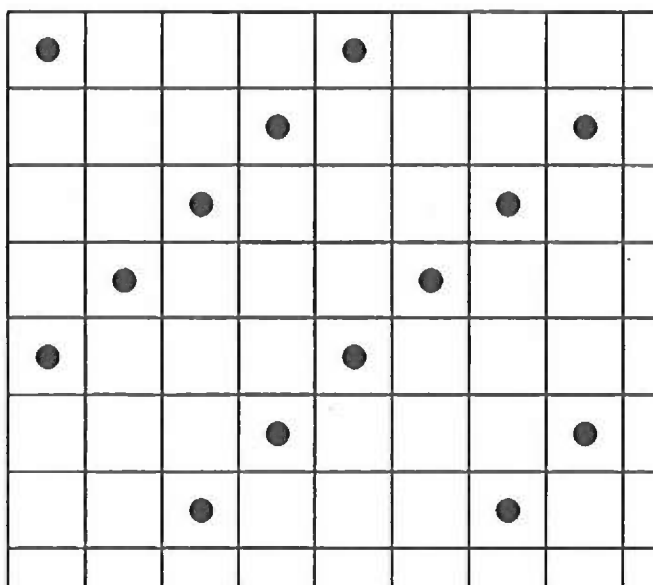
Figure 11: *Image sampled along the negative diagonal.*

5.2   Completeness of the Energy-zero-crossings representation.

The Energy-zero-crossings representation was proved to be experimentally complete for one dimensional signals. Let us remember how the reconstruction works in one dimension. Having decomposed a signal S^j into a set of details $(W^i)_{i \in [j-n..j]}$ and computed a set of zero-crossings and energies $(Z^i)_{i \in [j-n..j]}$, the method is the following:

1. Take any signal S_1^j.

2. Decompose S_1^j into a set of details $(X_1^i)_{i \in [j-n..j]}$ using the usual decomposition method given in section 4.1.

3. Match the zero-crossings and energies of the $(X_1^i)_{i \in [j-n..j]}$ with the $(Z^i)_{i \in [j-n..j]}$. This is described in detail in [1]. The basic principle is:

   (a) Add a piecewise linear continuous function to each X_1^i in order to create zero-crossings at the right location.

   (b) Suppress spurious zero-crossings by adding other piecewise linear functions.

   (c) Adjust the energies by multiplying each arch between two consecutive zero-crossings by a proper coefficient.

   This gives the $(Y_1^i)_{i \in [j-n..j]}$.

4. Reconstruct a signal S_2^j from the $(Y_1^i)_{i \in [j-n..j]}$ using the usual dyadic reconstruction method given in section 5.1.1.
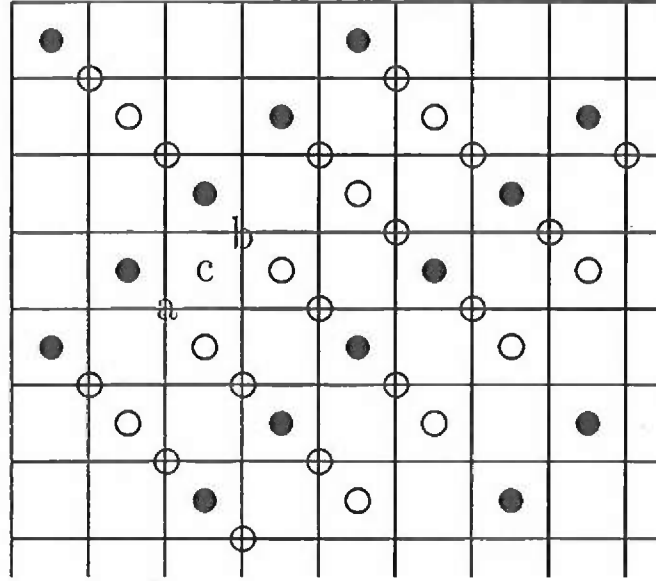
Figure 12: *Diagonal interpolation.*

5. Decompose S_2^j into a set of details $(X_2^i)_{i \in [j-n..j]}$.

6. Match the zero-crossings and energies of the $(X_2^i)_{i \in [j-n..j]}$ with the $(Z^i)_{i \in [j-n..j]}$.

7. reconstruct a signal S_3^j from the $(Y_2^i)_{i \in [j-n..j]}$ and so on ...

The S_k^j converge rapidly towards the original signal S^j. In order to reduce the number of iterations, we take as the S_1^j signal the dyadic reconstruction of a set of piecewise constant signals $(X_0^i)_{i \in [j-n..j]}$ which energies and zero-crossings already match the $(Z^i)_{i \in [j-n..j]}$. Five or six iterations usually give a satisfactory reconstructed signal.

The extension to images is quite immediate. We computed the energies and zero crossings for each line along the band-pass filtering direction treating these lines as one dimensional signals. Thus we can directly generalize the above mentioned algorithm to images. The reconstruction process works very well on images, but the only problem is that border effects are not totally controlled as they are in case of one dimensional processes. Figure 13 shows on the right the image reconstructed from the Energy-zero-crossings representation of the image on the left. The reconstruction took 5 iterations. The reconstruction process is quite accurate, and since the human visual system is very resistant to distortion, the differences are hardly noticeable to the naked eye. Figure 14 shows the enhanced image of the differences between the two images. As a comparison, the pixel values in the original image use the full range [0,255] whereas black and white colors in Figure 14 correspond to values -10 an +10 respectively. Notice that errors occur mainly along the diagonal contours at all frequencies. This is due to the diagonal interpolation method as explained in section 5.1.2.

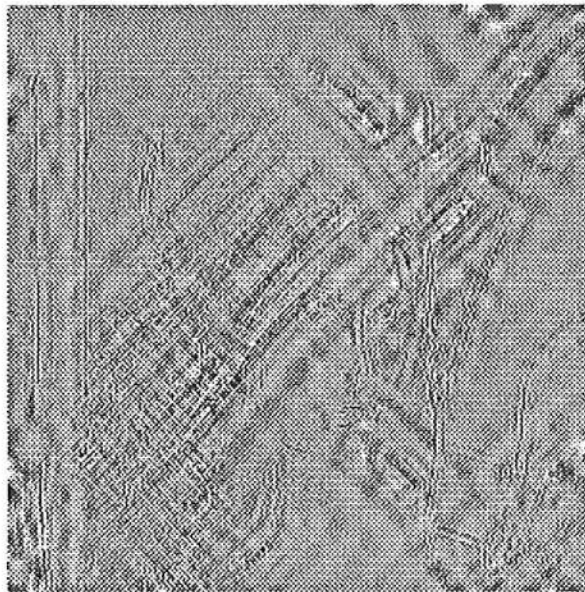Figure 13: *Original image and reconstruction from the Energy-Zero-Crossings representation.*

Figure 14: *Differences between the original image and its reconstruction.*

6 Applications to higher level vision processes.

6.1 Edge detection.

The Energy-zero-crossings representation, together with the multiorientation feature, is a useful model for edge finding algorithms. Indeed the zero-crossings point out where local variations are and the energies give a measurement of the intensity of these variations.

Let us emphasize some terminology points here. It is important to define the meaning of the word "direction" in what follows. supposing that we have applied a horizontal band-pass filter and a vertical low-pass filter to an image, the direction of the edges we will look for is the vertical direction (vertical edges are horizontal variations) thus we will refer to the vertical direction as the **main edge direction** whereas we will call the image an **image of horizontal variations**.

6.1.1 Basic edge-detection algorithm.

As a start we can simply decide that a zero-crossing belongs to an edge if the energies on the right and on the left of it are greater than a certain threshold. Growing an edge from this point is then easy because we know in which direction we must look. Figure 15 gives us an example of what a vertical edge looks like in terms of zero-crossings of an image of horizontal variations.

Having detected in row i a zero-crossing which has sufficient energy to possibly belong to an edge, we look in row $i + 1$ for the nearest zero-crossing having the same sign[6]. Two situations can then occur:

1. The energies on the right and/or left of the chosen zero-crossing are under the threshold. Then the edge cannot be defined further. This situation arises at the end of actual edges, or in textured regions where there are some energetical zero-crossing but which do not form organized patterns from row to row.

2. The energies on the right and left of the chosen zero-crossing are above the threshold. We can chain the zero-crossings together and search row $i + 2$ for the following zero-crossing of the edge.

---

[6]By convention the **sign** of a zero-crossing is the sign of the arch on its right
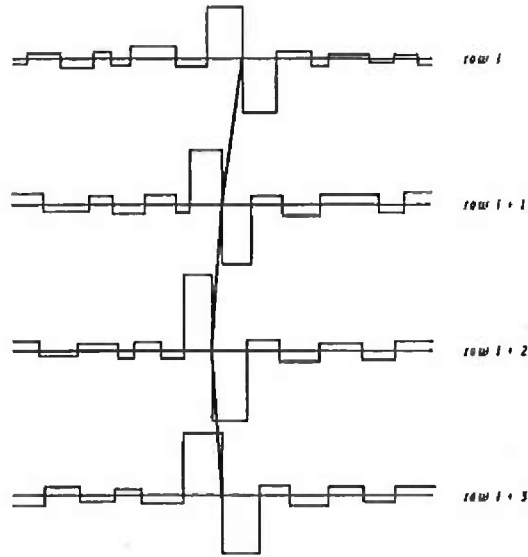
Figure 15: *Edge detection using zero-crossings.*

This process is called **edge-growing**. Similarly we can investigate rows $i - 1$, $i - 2$ and so on to extend our edge. In our implementation, we have again used the zero-crossing structure described in Figure 9 except that our zero-crossings are not linked along one line, but between consecutive lines.

6.1.2 Algorithm improvement.

Of course this algorithm is far too simple to give satisfactory results. First of all, if we use a single threshold, we often miss part of the edges we define due to contrast variations. However we must remember that we do not examine each zero-crossing independently to decide whether or not it is part of an edge. Our approach is to look for zero-crossings which would extend previously detected edges. Thus we use **connectivity** as a criterion. Having realized that connectivity is part of the decision process, we can then reduce the threshold constraint while growing the edge. So we now have two different thresholds. The first one is used to detect starting points for our edge growing loop, and the other one, which is weaker, is used while actually growing the edges.

The next technical problem is that although now we can detect connected "chains" of zero-crossings, not all these chains correspond to actual edges in the image. The reason for this is the following: The theory of wavelet provides us with a large class of possible filters. We then put some requirements for the filter we choose:

- It has to be proportional to the second derivative of a smoothing function, so that the zero-crossings mark inflection points.

- It has to be localized in space so that we can accurately locate these inflection points. This is also important for computation speed.

The trade-off to this is that our filter has "bounces", instead of being nicely smooth like for example the laplacian of a gaussian. Figure 17 shows the zero-crossings corresponding to two step edges of Figure 16 for resolutions 2^j and 2^{j-1}. We can see that for each edge there is a main zero-crossing and a minor one on each side. If the energies of these minor zero-crossing happen to be above the threshold, we will detect some "echos" surrounding our edge.
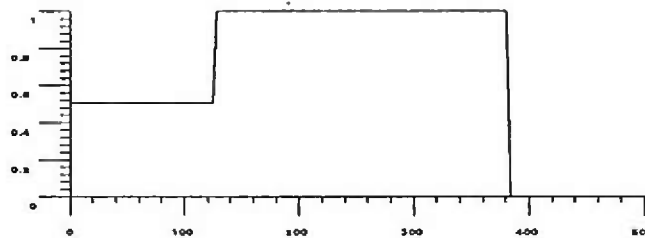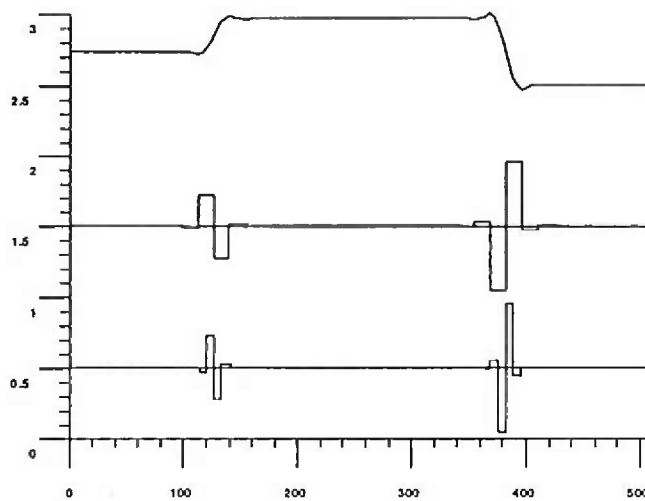

Figure 16: *Step edge.*

Figure 17: *Zero-crossings of a step edge.*

However let e_{max} be the maximum of the left and right energy for one of these minor zero-crossings, and e_{min} be the minimum, the value of the ratio $\frac{e_{min}}{e_{max}}$ is characteristic of the

edge type. If this edge is an echo, this ratio is usually 15 to 20 % depending on the filter we chose, whereas for real edges, its value is above 40 %. This is a simple way of discriminating echo edges from actual ones. Figure 18 and Figure 19 shows the edges of the image in Figure 13 as they appear at resolutions 2^j and 2^{j-1}. Notice that some edges are redundant whereas some appear only at a the coarser or finer resolution. Notice also that the junction between orientations is smooth if the actual edge is spatially smooth (i.e. if its curvature is small).

Figure 18: *Edges of image "lady" in 4 orientations at resolution* 2^{j-1}.

Figure 19: *Edges of image "lady" in 4 orientations at resolution 2^{j-2}.*

6.1.3 Edge finding and multiorientation.

It is possible that an edge will appear on several images corresponding to different orientation. Indeed we chose filters that were localized in the spatial domain. As a consequence, their cut-off capacities were smothered in the frequency domain. This causes the filters to overlap both between orientations and resolutions. For example if while looking for vertical edges we detect an edge which is bent 30° from the vertical, there are great chances that this edge will also be detected by looking at the zero-crossings of the image of diagonal variations. In fact we can control the amount of redundancy by including constraints on the direction in the edge-growing algorithm. For example we can stop growing the edge if the next potential segment happens to make an angle of more than 30° with the main edge direction, as we expect it to be detected in another orientation anyway. This is important because we might want to build algorithms which link continuous segments, such as circles, having some parts in different orientations.

6.2 Corner detection.

As stated in section 3 if we use a pattern recognition process based on image decomposition and recomposition, multiorientation is a definite advantage for the detection of interesting structural points like corners, "T" junctions and so on. Figure 21 shows the horizontal and vertical edges of the image in Figure 20 representing a model of the Penn campus. It shows that the corners are not affected by our edge detection method. All we have to do is to take edges within two different orientations and to look for crossing points.

The wavelet filters are localized enough to prevent ambiguities on corner detection. Moreover, we can get accurate information about the tangents at the corner location. These corner points are relevant to the image understanding for several reasons. First of all, corners are usually the points of an object's contour which are the most useful to define it's geometry. For example, it much easier to detect motion by observing the corners than by trying to match edges. Also corner points are generally points of interaction between several objects in the scene. A "T" junction for example is a point which which plays a role in the definition of three different objects or regions. so there is a lot of information associated to such a point. Another reason is that unconnected edges very seldomly exist in real scenes. Or if these exist, they often correspond to a lack of information we should take care of. The best example of this are illusory contours. The human visual system reacts to illusory patterns by mentally adding some edge portions to incomplete segments so as to try to define a recognizable shape. If we want to detect unconnected edge segments for further analysis, it is a good idea to first detect edges which are connected by corners, then to deal with the remainder.
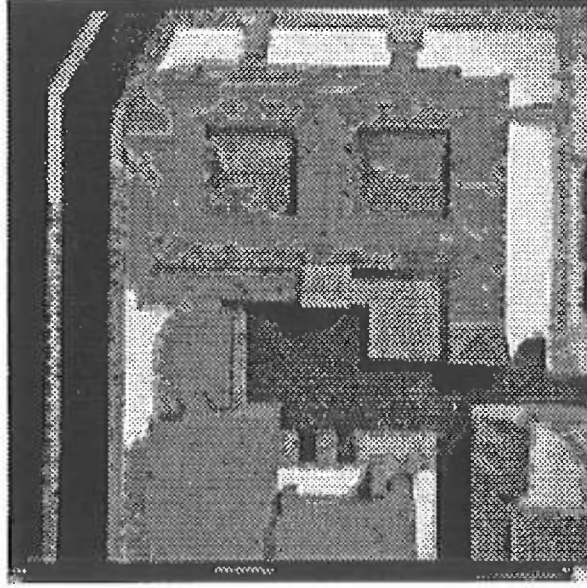
Figure 20: *Image : Penn campus model.*

6.2.1 Algorithm.

Our algorithm quite straightforward. Given L_1 and L_2 two edges (i.e. chained lists of points) in different orientations, let p_1 and p_2 be the first points of L_1 and L_2 respectively:

1. move along L_1 to find the point p'_1 such that the distance $d(p'_1, p_2)$ is locally minimum. This means that if b and a are the points before and after p'_1, both $d(a, p_2)$ and $d(b, p_2)$ are greater than $d(p'_1, p_2)$.

2. move along L_2 to find the point p'_2 such that the distance $d(p'_1, p'_2)$ is locally minimum.

3. move along L_1 forward or backward from p'_1, depending on which direction decreases the distance, to find the point p''_1 such that $d(p''_1, p'_2)$ is locally minimum. If p''_1 equals p'_1 then p'_1 and p'_2 are locally the closest points between L_1 and L_2 and should be examined as possibly defining a corner. If p''_1 is different from p'_1 then:

4. move along L_2 forward or backward from p'_2, depending on which direction decreases the distance, to find the point p''_2 such that $d(p''_1, p''_2)$ is locally minimum. If p''_2 equals p'_2 then p''_1 and p''_2 are locally the closest points between L_1 and L_2 and should be examined as possibly defining a corner. If p''_2 is different from p'_2 then:

5. move along L_1 forward or backward from p''_1 to find the point p'''_1 such that $d(p'''_1, p''_2)$ is locally minimum and so on...

Once we have obtained points q_1 and q_2 which are the points of L_1 and L_2 locally closest to each other, we decide according to $d(q_1, q_2)$ whether these points define a corner. if $d(q_1, q_2)$ is smaller than a certain threshold, we compute the exact position of the point where L_1 and
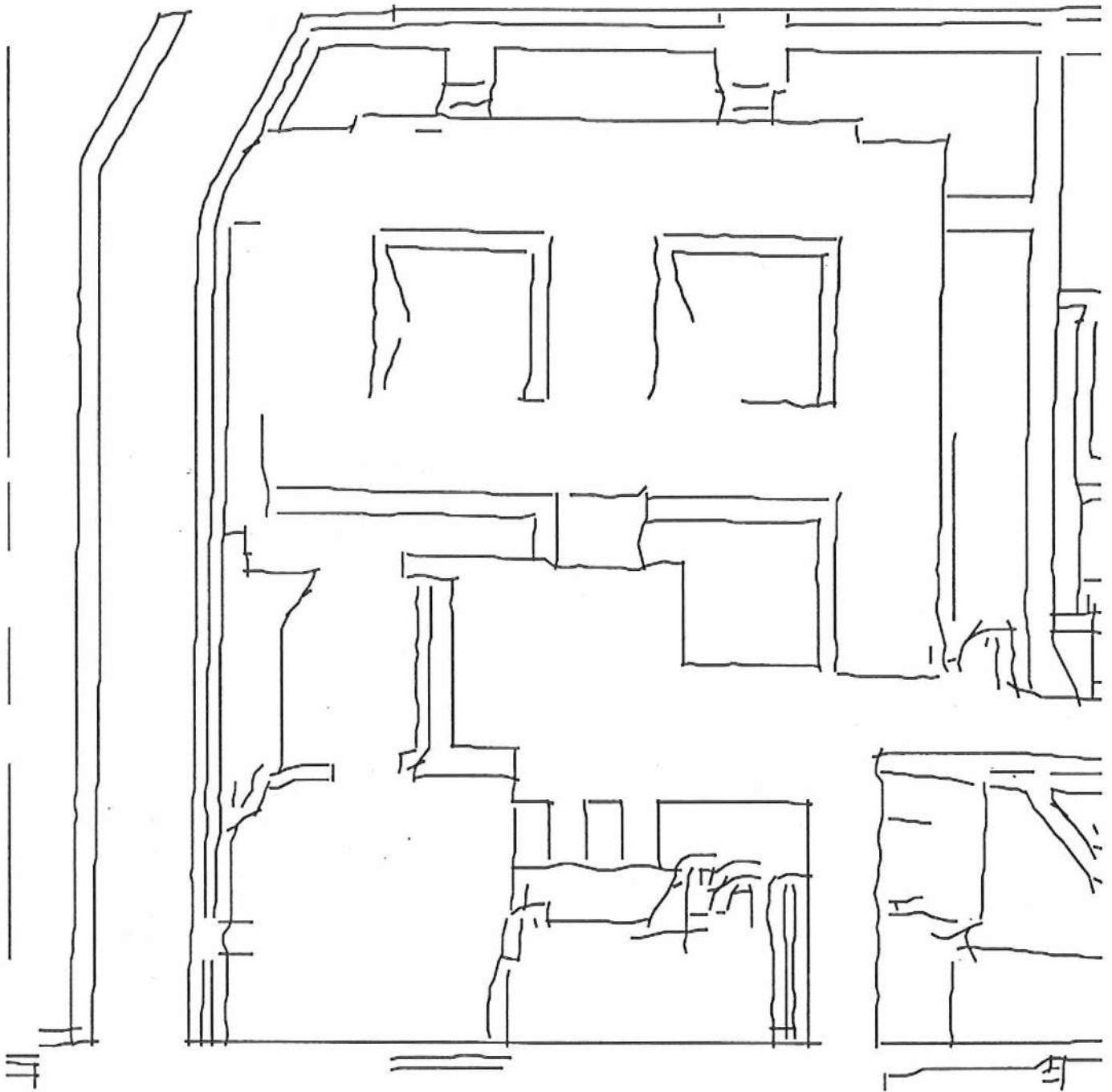
Figure 21: *Edges of image "Penn campus" in 2 orientations at resolution* 2^{j-1}.

L_2 cross and the tangents to the edges at this point. If the edges do not actually cross we extend the last segment in order to extrapolate the corner point as shown in Figure 22.
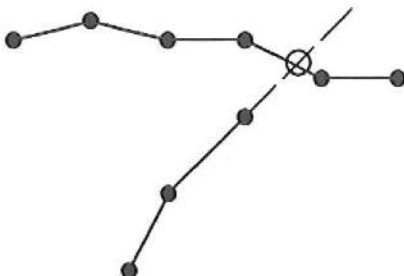

Figure 22: *Extrapolation of a corner point.*

The value of the threshold is not at all empirical. If we examine our decomposition process, we realize that at a any resolution , and for any orientation, the spatial resolution on the band-pass filtering direction is four times greater than on the low-pass filtering direction[7]. Let us we take the scale of the original image to compute point coordinates. The position uncertainity is then four times greater on the low-pass filtering direction. We take as a threshold the uncertainity corresponding to the low-pass filtering direction's resolution. Let us take an example: Suppose that we decompose a signal S^j with 2^j samples into a set of details $(W_k^i)_{(i,k)\in[j-n..j]\times[1..4]}$ with an expansion factor of 4. If we extract the vertical edges from the zero-crossings of W_1^{j-1}, knowing that W_1^{j-1} has 2^{j-1} rows and 2^{j+1} columns, we can expect an uncertainity of 2 on the vertical coordinates of the points which define these vertical edges. Similarly we can expect an uncertainity of 2 on the horizontal coordinates of the points of the horizontal edges. So the two points defining a possible corner can be as far as $2\sqrt{2}$ from one another due to the combination of these uncertainities. We will then take $2\sqrt{2}$ as a threshold value and assume that two points define an actual corner if the distance between them is smaller than that. Figure 23 shows the corners detected from the edges shown in Figure 21.

6.2.2 Corner resolution.

Of course, an interesting question is to know what is the minimum angle a corner must have in order to be detected. Part of the answer is that a corner will automatically be detected if the two edges that define it appear in separate orientations. Thus the performance of our corner detection process is linked directly to the number of different orientation we choose to separate, as well as, to the amount of redundancy between them. However as we said before, it is very difficult to implement more than four different orientations.

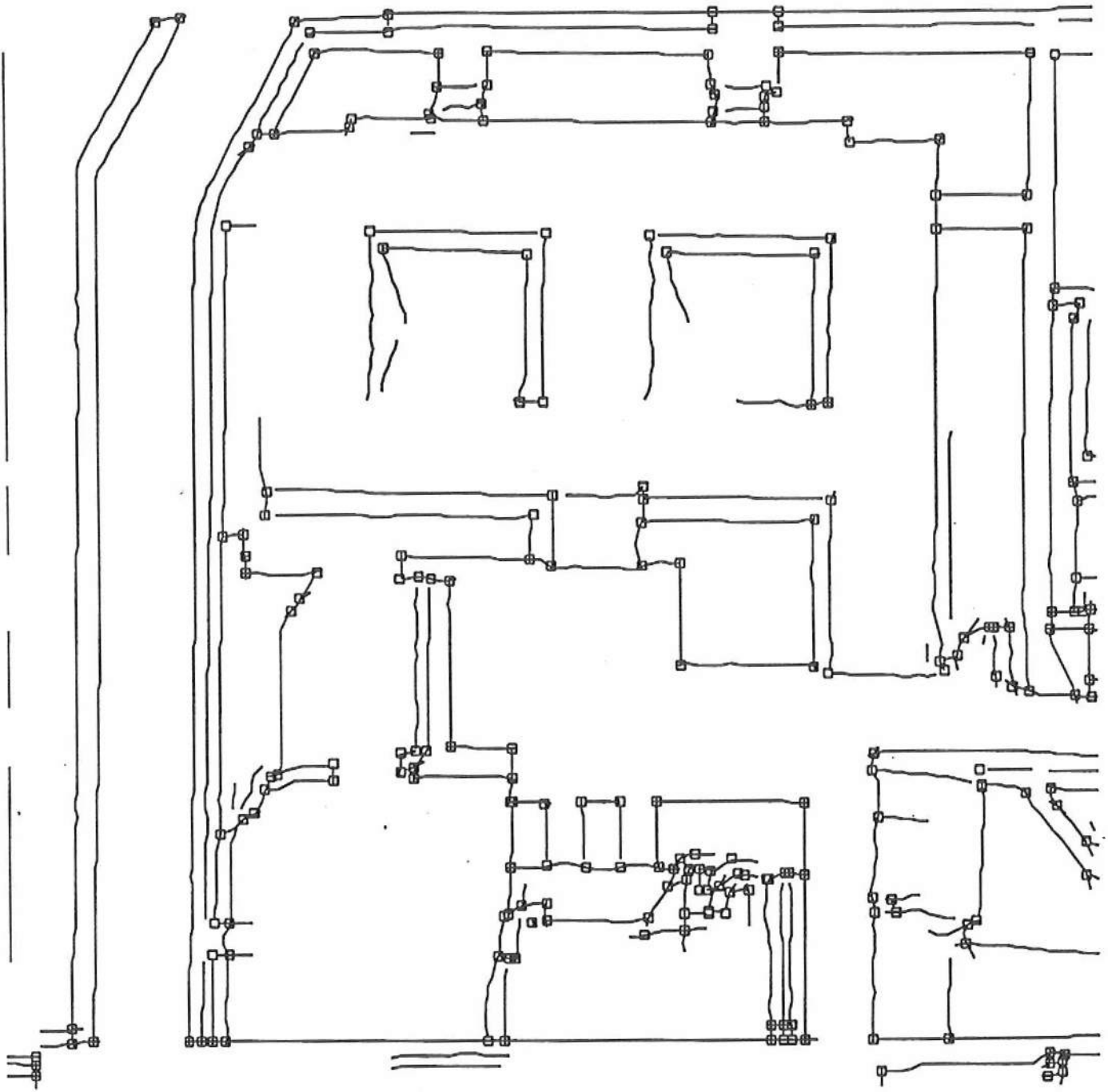[7]Or twice greater, depending on the expansion factor mentioned in section 4.1.

Figure 23: *Corners of image "Penn campus" at resolution* 2^{j-1}.

An alternative could be to use only one separable filter corresponding to a horizontal band-pass filter and a vertical low pass filter, for example, and to apply it to several rotation interpolations of the original image. Of course while performing the rotation interpolation we will loose high frequencies. However we can still perform edge and corner detection at all resolutions except the first one.

6.3 Unconnected lines detection: an example of feed-back.

As stated earlier, unconnected line segments occur very seldom in natural scenes. However, many edge detectors which are not based on region growing, including ours, fail to connect some edges. Knowing that unconnected patterns are very unlikely, we can localize the areas where our edge detection process gives incomplete results, and tune the process differently in those areas. Also, we have the alternative of using different methods for this particular image portion such as using another picture with a different lighting condition, or artificially enhancing the contrast.

Detecting edges with loose ends is done simply by looking at the two ends of each edge and to see whether they have been used to define corners. If either of them is not, it is considered as loose end. As we now try to make an adaptive process by locally tuning the parameter of our edge detector, it is useful to regroup these loose end points into clusters. This way we can apply some feed back at different levels. Either we change the image acquisition system in the area corresponding to a particular cluster, or we locally change the threshold and try to extend and connect the lines within a cluster. The clustering algorithm is based on the minimum spanning tree computation and is given in Appendix C. Results are shown in Figure 24. The circled area correspond to regions where the edges are generally less obvious.

A   Dyadic decomposition with an expansion by 2.

A.1   One dimensional algorithm.

Let S^j be a signal at resolution 2^j and W^j be the signal of the differences between S^j and S^{j-1}. Let $U_{\frac{\pi}{2^i}}$ be the low-pass filter with a cutoff frequency of $\frac{\pi}{2^i}$ and $G_{\frac{\pi}{2^i}}$ be the high-pass filter with a cutoff frequency of $\frac{\pi}{2^i}$. supposing that we want to expand the $(W^i)_{i \in [j-n..j]}$ by a factor 2 the algorithm is:

1. Filter S^j by $G_{\frac{\pi}{2}}$. The result is W^j
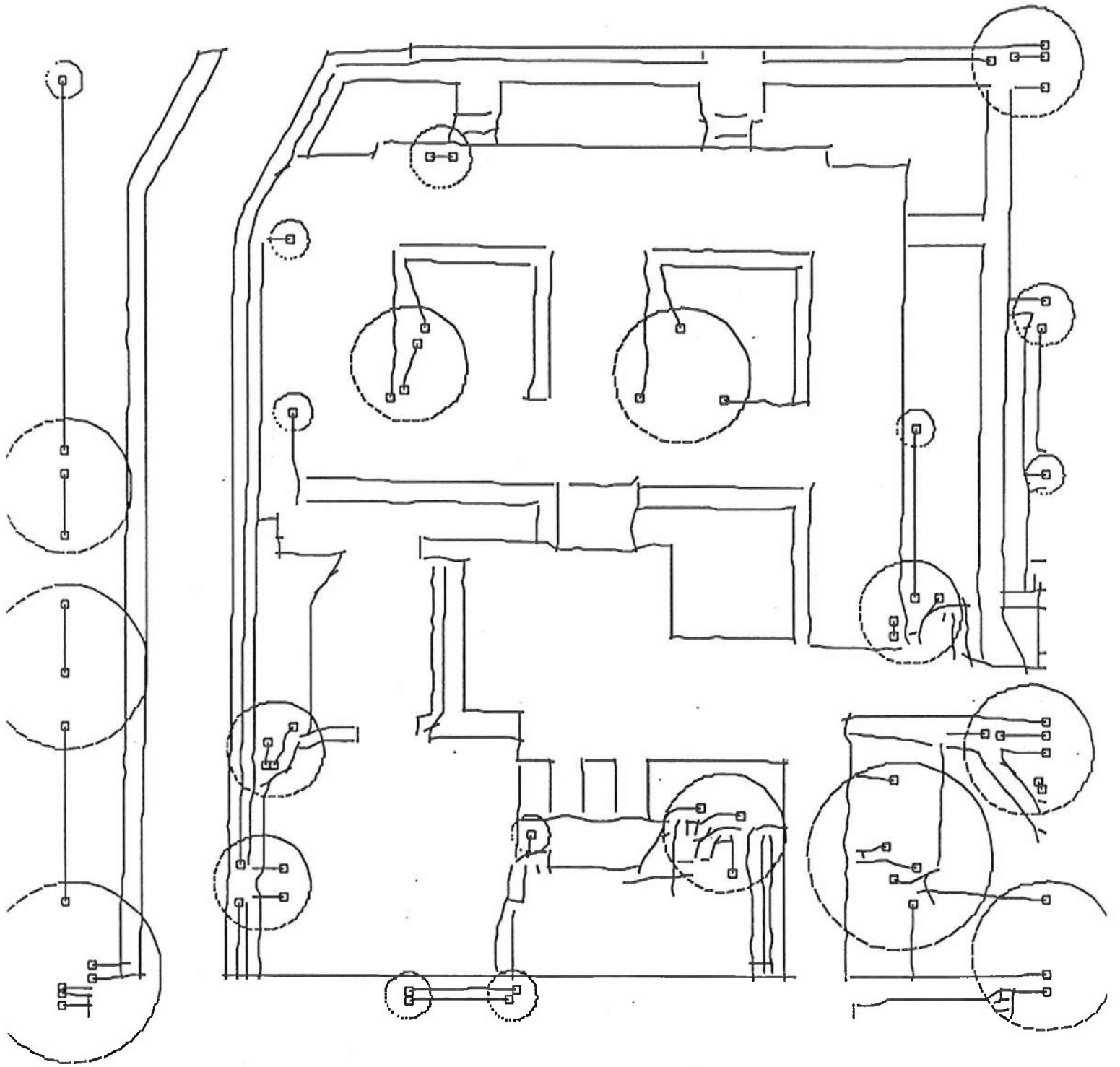
2. Expand W^j by a factor 2

Figure 24: *Corners of image "Penn campus" at resolution 2^{j-1}.*

3. Filter S^j by $U_{\frac{\pi}{2}}$. The result is S^{j-1} (but it still has 2^j samples !)

4. Filter S^{j-1} by $G_{\frac{\pi}{4}}$. The result is W^{j-1}

5. Filter S^{j-1} by $U_{\frac{\pi}{4}}$ and sample by 2. The result is S^{j-2}

6. Filter S^{j-2} by $G_{\frac{\pi}{4}}$. The result is W^{j-2}

7. Filter S^{j-2} by $U_{\frac{\pi}{4}}$ and sample by 2. The result is S^{j-3}

8. Filter S^{j-3} by $G_{\frac{\pi}{4}}$. The result is W^{j-3}

9. Filter S^{j-3} by $U_{\frac{\pi}{4}}$ and sample by 2. The result is S^{j-4} and so on...

A.2   Two dimensional algorithm.

Let S^j be an image at resolution 2^j and W_1^k W_2^k W_3^k and W_4^k be the image details at resolution 2^k with k inferior or equal to j in the horizontal, vertical, first diagonal and second diagonal directions respectively. Let $\left(G_{\frac{\pi}{2^i},k}\right)_{k \in [1..4]}$ be the directional band-pass filters whose general shapes are given in Figure 5 but which isolate the parts of the spectrum between $\frac{\pi}{2^i}$ and $\frac{\pi}{2^{i-1}}$ (instead of between $\frac{\pi}{2}$ and π). Let $U_{\frac{\pi}{2^i}}$ the low-pass filter of cutoff frequency $\frac{\pi}{2^i}$.

1. Expand S^j by a factor 2 along both horizontal and vertical axes using spline interpolation. This will confine its spectrum between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. We will refer to the resulting image as S_e^j.

2. Filter S_e^j by the $\left(G_{\frac{\pi}{4},k}\right)_{k \in [1..4]}$ and sample by 2 along the low-pass direction . The results are the $\left(W_k^j\right)_{k \in [1..4]}$.

3. Filter S_e^j by the low-pass filter $U_{\frac{\pi}{4}}$ and sample by 2 along each axis . The result is S^{j-1} (but it has 2^j samples).

4. Filter S^{j-1} by the $\left(G_{\frac{\pi}{4},k}\right)_{k \in [1..4]}$ and sample by 4 along the low-pass direction . The result is the $\left(W_k^{j-1}\right)_{k \in [1..4]}$.

5. Filter S^{j-1} by the low-pass filter $U_{\frac{\pi}{4}}$ and sample by 2 along each axis . The result is S^{j-2}.

6. Filter S^{j-2} by the $\left(G_{\frac{\pi}{4},k}\right)_{k \in [1..4]}$ and sample by 4 along the low-pass direction . The result is the $\left(W_k^{j-2}\right)_{k \in [1..4]}$.

7. Filter S^{j-2} by the low-pass filter $U_{\frac{\pi}{4}}$ and sample by 2 along each axis . The result is S^{j-3} and so on ...

B Coefficients of wavelet filters.

Below are the coefficients of filters $U_{\frac{\pi}{2}}$ and $G_{\frac{\pi}{2}}$. This filters are symmetrical with respect to 0. To obtain the filters $U_{\frac{\pi}{4}}$ and $G_{\frac{\pi}{4}}$ we periodized the filters by inserting one 0 between each coefficients as explained at the end of section 4.1. To obtain the filters $U_{\frac{\pi}{8}}$ and $G_{\frac{\pi}{8}}$ we periodized the filters by inserting three 0 between each coefficients.

$U_{\frac{\pi}{2}}$ filter	
abscissa	value
0	0.587132
1	0.274067
2	-0.056679
3	-0.021478
4	0.013405
5	-0.003085

$G_{\frac{\pi}{2}}$ filter	
abscissa	value
0	-0.590628
1	0.267548
2	0.047401
3	-0.020079
4	-0.001451
5	0.002588
6	-0.000750

These are the coefficients of filter $G_{\frac{\pi}{4},3}$ shown in Figure 7. This filter is symmetrical with respect to the point (0,0). To obtain the filters $G_{\frac{\pi}{8},3}$ we periodized this filter by inserting one 0 between each coefficient along the lines and columns.

	$G_{\frac{\pi}{4},3}$ diagonal filter								
	-4	-3	-2	-1	0	-1	2	3	4
0	-0.01216	-0.00131	-0.07572	0.00370	0.17657	0.00370	-0.07572	-0.00131	-0.01216
1	0.0	-0.01478	0.00172	0.12439	0.00370	-0.11448	-0.00476	0.01598	0.0
2	0.0	0.0	0.03834	0.00172	-0.07572	-0.00476	0.03815	0.0	0.0
3	0.0	0.0	0.0	-0.01478	-0.00132	0.01598	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	-0.01273	0.0	0.0	0.0	0.0

C Clustering algorithm.

Our clustering algorithm is designed to regroup points in roughly circular shaped clusters of a given maximum radius. This algorithm uses a second parameter which is the maximum distance between two points in a cluster.

Let us suppose we have a set of points $(P_i)_{i \in [1..n]}$. Let d_{max} be the desired maximum distance between two points in a cluster and r_{max} be the desired maximum radius of clusters.

The algorithm is the following:

1. Compute the minimum spanning tree of the $(P_i)_{i \in [1..n]}$. The result is a set of arcs $(A_i)_{i \in [1..m]}$.

2. Remove every arc greater than d_{max}. This yields several sub-trees. Points which belong to the same sub-tree form a cluster. The result is thus a set of clusters $\left(C_i^{(0)} \right)_{i \in [1..k_0]}$

3. For each cluster $C_i^{(0)}$ compute the center of gravity g_i of all the points in it , then compute the distances between g_i and each point in the cluster. Let r_i be the maximum of these distances.

4. For each cluster $C_i^{(0)}$, if r_i is greater than r_{max} then break the longest arc of the corresponding sub-tree. This yields two "sub-subtrees" defining two sub-clusters $C_i^{(1)}$ and $C_i^{(2)}$.

5. Repeat the two previous steps on the $C_i^{(j)}$ until no further subdivision occurs. The result is a set of clusters C_i fulfilling the desired requirements.

References

[1] Stephane Mallat. *Multiresolution Representations and Wavelets*. PhD thesis, University of Pennsylvania, Department of Computer and Information Science, School of Engineering and Applied Science, August 1988.

[2] Stephane Mallat. *Dyadic wavelet energy zero-crossings*. Technical Report MS-CIS-88-30, University of Pennsylvania, Department of Computer and Information Science, School of Engineering and Applied Science, 1988. To appear as an invited paper in IEEE Trans. on Information Theory.

[3] Eric p. Krotkov. *Results in finding edges and corners in images using the first directionnal derivative*. Technical Report MS-CIS-85-14, University of Pennsylvania, Department of Computer and Information Science, School of Engineering and Applied Science, March 1985.

[4] F. Campbell and J. Kulikowski. Orientation selectivity of the human visual system. *Journal of Physiology*, 197:437–441, 1966.

[5] R. C. Crochiere and L. R. Rabiner. Interpolation and decimation in signal processing. *Proceeding on the IEEE*, 69, March 1981.

[6] Philip J. Davis and Philip Rabinowitz. *Methods of Numerical Integration*, chapter 2.3.1, page 62. Computer Science and Applied Mathematics, 19??

List of Figures