Multi-Oriented Multi-Resolution Edge Detection

MS-CIS-90-13
GRASP LAB 205

Laurent Peytavin

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

February 1990

Acknowledgements:
AirForce grant AFOSR 88-0244, AFOSR 88-0966, Army/DAAG 29-84-K-0061, NSF-CER/DCR82-19196 A02, NASA NAG5-1045, ONR SB-35923-0, NIH1 R-1-NS-23636-01, NSF INT85-14199, DARPA N00014-88-K-0630, NATO grant No. 0224/85, DuPont Corp. by Sandia 75, Post Office, IBM Corp. and LORD Corp.

To my parents.

Acknowledgements.

I am particularly grateful to D. Ruzena Bajcsy who guided me and supported me all through this project. I am also very grateful to Aleš Leonardis who contributed to lots of ideas and concepts of this edge detector. His experience in image processing helped me to enlarge my vision of the subject and eventually orientated this work so that it could fit a more elaborated image segmentation project. I am grateful to Stephane Mallat for the time he spent introducing me to the theory of wavelets, and giving me some important hints for my work. I would like to thank Sang W. Lee who helped me to use his work on color constancy, and Helen Anderson who helped me patiently in so many ways. I received lots of help and a good introduction to the lab and to my project from Howard Choset. I also will not forget the help I received from: Ray McKendall, Dmitry Cherkassky, Ulf Cahn Von Seelen, Jasna Maver and John Bradley.

This work was in part supported by the following contracts and grants: Airforce grant AFOSR 88 0244,AfOSR 88-0966, Army/DAAG-29-84-K-0061, NSF-CER/DCR82-19196 Ao2, NASA NAG5-1045, ONR SB-35923-0, NIH 1-RO1-NS-23636-01,NSF INT85-14199, ARPA N0014-88-K-0630,NATO grant No.0224/85, DuPont Corp. by Sandia 75 1055,Post Office, IBM Corp. and LORD Corp.

Abstract

In order to build an edge detector that provides information on the degree of importance spatial features represent in the visual field, I used the wavelet transform applied to two-dimensional signals and performed a multi-resolution multi-oriented edge detection. The wavelets are functions well-localized in spatial domain and in frequency domain. Thus the wavelet decomposition of a signal or an image provides outputs in which you can still extract spatial features and not only frequency components.

In order to detect edges the wavelet I chose is the first derivative of a smoothing function. I decompose the images as many times as I have directions of detection. I decided to work for the moment on the X-direction and the Y-direction only. Each step of the decomposition corresponds to a different scale. I use a discrete scale $s = 2^j$ (dyadic wavelet) and a finite number of decomposed images. Instead of scaling the filters at each step I sample the image by 2 (gain in processing time). Then, I extract the extrema, track and link them from the coarsest scale to the finest one. I build a symbolic image in which the edge-pixels are not only localized but labelled too, according to the number of appearances in the different scales and according to the contrast range of the edge. Without any arbitrary threshold I can subsequently classify the edges according to their physical properties in the scene and their degree of importance.

This process is subsequently intended to be part of more general perceptual learning procedures. The context should be: none or as little as possible a priori knowledge, and the ultimate goal is to integrate this detector in a feedback system dealing with color information, texture and smooth surfaces extraction. Then decisions must be taken on symbolic levels in order to make new interpretation or even new edge detection on ambiguous areas of the visual field.

Contents

1 Introduction 4

2 Motivations and goals 5
- 2.1 Motivations . . . 5
  - 2.1.1 The multi-resolution concept . . . 5
  - 2.1.2 The multi-orientation concept . . . 5
- 2.2 Goals . . . 6

3 Multiscale decomposition 7
- 3.1 The window Fourier Transform . . . 7
- 3.2 The wavelet Transform . . . 8
- 3.3 The wavelet decomposition . . . 10

4 Coarse to fine tracking 16
- 4.1 The edge behavior . . . 16
  - 4.1.1 Ideal and single edges . . . 16
  - 4.1.2 The edge behavior in real images . . . 20
- 4.2 The tracking process . . . 22
  - 4.2.1 Stage 1: The Edge Tracking and Linking through scale space . . . 22
  - 4.2.2 Stage 2: The Edge Signature Computation . . . 24
  - 4.2.3 Stage 3: The Symbolic Merge . . . 24

5 Interpretation and Edge Classification 27
- 5.1 Physical causes of an edge . . . 27
- 5.2 The Color Images . . . 28

6 Results and Comments 32
- 6.1 Interpretation of the outputs . . . 32
- 6.2 Comments and Conclusion . . . 40

1 Introduction

Multiscale decomposition or multifrequency channel decomposition have been used in many applications in image recognition within the last 10 years. The fact is that evidences in the physiology of the human vision has been gathered showing that the retinal image is decomposed into several spatially oriented frequency channels. Our purpose is not to imitate the human vision. It helps us to understand the motivation of such processings and it allows us to deal with a good definition of what an image segmentation should be and should use as first low-level processings.

We are now convinced that low-level processings must provide information according to the degree of importance the features represent in the vision field. The edge detector is at the very beginning of the long chain of visual recognition. It became interesting to introduce at this stage the notion of scale space in order to provide ordered edges and contours to upper levels.

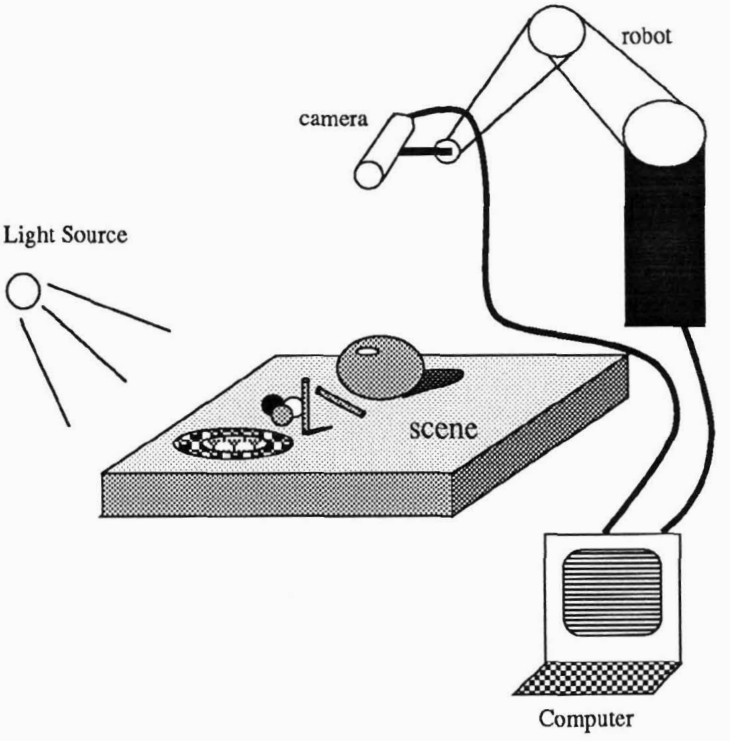

Figure 1: Robot Environment.

2 Motivations and goals

2.1 Motivations

These motivations mostly explain the a-priori trust that D. R. Bajcsy, S. Mallat, N. treil, Aleš Leonardis and I had in going as far as possible into multi-scale decomposition with wavelets. They are related to two fundamental concepts: the multi-resolution concept and the multi-orientation concept.

2.1.1 The multi-resolution concept

Detecting edges from an image is at the very beginning in image segmentation. This is a very important process particularly when we want to deal with images of the real world. Our visual environnement is obviously made with a set of more or less sharp local intensity transitions. Artificial images in medicine for instance provide other kind of objects that have blurred shapes. In the effort of giving vision to robots the edge detection must take an important part.

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 succesively the doors and the 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 (or scale) would enable us to establish a hierarchy between these pieces of information. We want to get rid of details while looking after the "context" (here the house outline) and then focuse our interest on the highest frequency features and edges to improve the recognition.

In other words a good segmentation must integrate this multi-scale classification right at the beginning. Thus the edge detection must respect this first concept.

Besides as I already mentioned it has been proved that some brain cells in the visual cortex respond specifically to stimuli at a certain frequency. The multiresolution frequency channel definitely seems to be the way to approach the perfection of human vision.

2.1.2 The multi-orientation concept

There are two ways to analyse an image. One is to perform isotropic analysis, that is to use isotropic filters, the other one is to give preference to some directions of detection. The first one is simple and does not provide any kind of redundancy. The last one, if not simple, provides accurate detections in a limited sector around the directions. When speaking of contour detection, corners are also well preserved with multi-orientation process when smoothed with isotropic filters.

It is interesting to note that multiorientation is one of the features of the human vision system too. Some brain cells respond to orientation stimuli and perform a 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.

This multi-orientation concept eventually fits the wavelet decomposition very well as we are going to see. Then it makes the coarse-to-fine tracking easier and the edge behavior happens to be a one-dimensional problem.

2.2 Goals

This edge detector has never been conceived as a single process. It is to be integrated in a segmentation system that could perform perceptual learning of real scenes. The primary application is obviously to provide robots with this perceptual learning. For the moment, we do not need to deal with frames and movements. We are not in the context of active vision. In other words the scene must be first well understood. The system is assumed to use as many clues as it can to understand its static visual field. The time constraint is not so heavy.

These considerations did not allow us to build turtle-like algorithms. As a matter of fact the wavelet decomposition is likely to be easy implemented in parallel machines. We worked also on the output of the edge detector in order to give quick, easily and meaningful readable data to upper levels. We kept in mind that the system would come back to do some new interpretation whenever some ambiguous areas are detected.

That is why we chose as output a symbolic image that fits exactly the real one. The grey scale information has disappeared. Instead a code is given to each pixel. In this code we put several pieces of information. Among them there are the degree of details that the edge-pixel represents and the contrast range of the local variation.

We will see that these two pieces of information can be combined to provide nice edge classification directly related to the physical properties of the image, that can help us for example to distinguish small highlights from shadowy contours.

We subsequently intended to apply this detector in parallel on 3-plane color images. We expected the results of such a detection on the brightness image, the hue image and the saturation image of the same scene to give us other clues to extract meaningful contours.

3 Multiscale decomposition

A The multiscale decomposition of an image must provide a set of different signals relevant to specific frequency channels.

B In order to segment and detect edges the decomposed signals must be readable and meaningful. It means that some spatial features must be recognized. It implies that the function we use to perform the decomposition must be limited in the spatial domain.

The Fourier transform does not verify the last condition.
Indeed the family $(e^{j2\pi fx})_j^{x\epsilon R}$ is not band-limited.
We eventually had to find and use other decompositions.

3.1 The window Fourier Transform

Some researchers in computer vision used the window Fourier transform.
(Notation: $\mathbf{L}^2(\mathbf{R})$ denotes the Hilbert space of measurable, square-integrable one dimensional functions).

The window Fourier transform of $f\epsilon\mathbf{L}^2(\mathbf{R})$ is defined by:

$$G_f(\omega, u) = \int_{-\infty}^{+\infty} e^{-i\omega x} g(x-u) f(x) dx.$$

G_f measures locally, around the point u, the amplitude of the sinusoidal component of frequency ω. This decomposition satisfies A and B. But it has several drawbacks when applied to image analysis. The spatial and frequency resolution domain are constant (Fig 2). Once g is chosen the resolution of the decomposed signals does not change. It means that we have to tune very precisely the dimension of g to detect what we want. The major problem is that we do not know the size of the objects in a scene. And above all, the need of good resolution depends upon the frequency to be detected. High frequency features can be detected with large band filters whereas low frequency ones need more accurate filters in the Fourier domain.

In order to avoid the inconvenience of a transform having a fixed resolution in the spatial and frequency domain, Morlet and Grossmann in 1984 [3] defined a decomposition based on dilations and named it the wavelet decomposition.

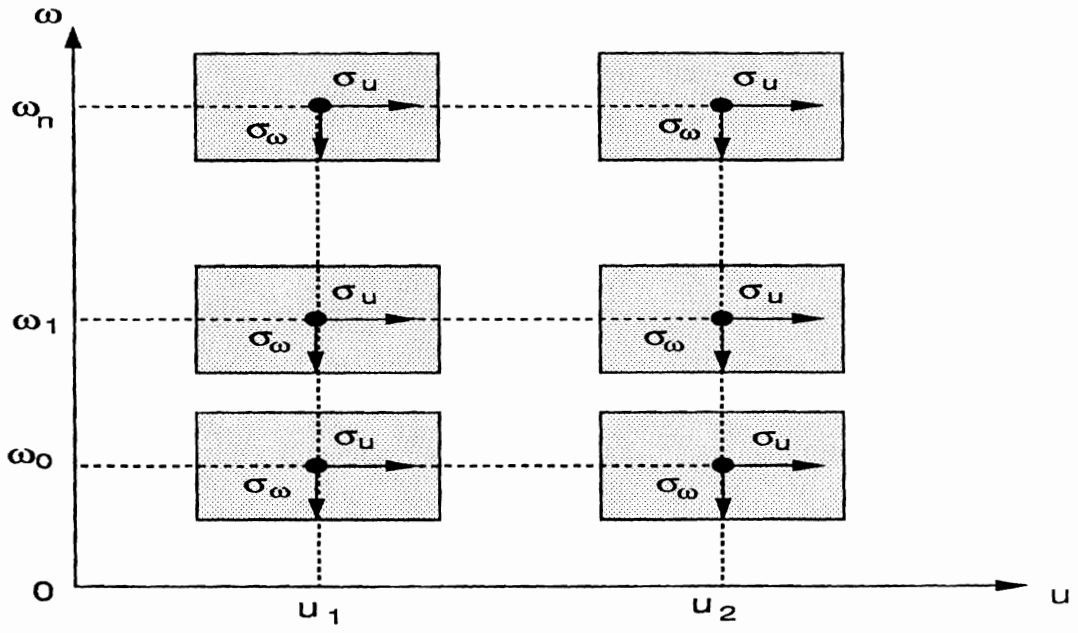

Figure 2: Phase-scale representation: σ_u and σ_ω are the standard deviations of $g(x)$ and of the Fourier transform of $g(x)$.

3.2 The wavelet Transform

The family of wavelet functions comes from the dilation and translation of a unique function $\psi(x)$: $(\sqrt{s}\ \psi(s(x-u)))_{(s,u)\epsilon R^2}$.

The wavelet transform of $f \epsilon \mathbf{L}^2(\mathbf{R})$ is defined by:

$$W_f(s,u) = \int_{-\infty}^{+\infty} f(x)\sqrt{s}\ \psi(s(x-u))dx.$$

It can be rewritten as inner products in $\mathbf{L}^2(\mathbf{R})$

$$W_f(s,u) = f * \tilde{\psi}_s(u) =< f(x), \sqrt{s}\ \psi(s(x-u)) > .$$

Since the Fourier Transform of $\tilde{\psi}_s(x)$ is given by

$$FT(\tilde{\psi}_s(x)) = \frac{1}{\sqrt{s}} FT(\psi)(\frac{\omega}{s})$$

The shape of the resolution cells varies with the scale s. This is illustrated in Fig 3. When the scale s is small, the resolution is coarse in the spatial domain. If the scale s increases, the resolution increases in the spatial domain and decreases in the frequency domain.

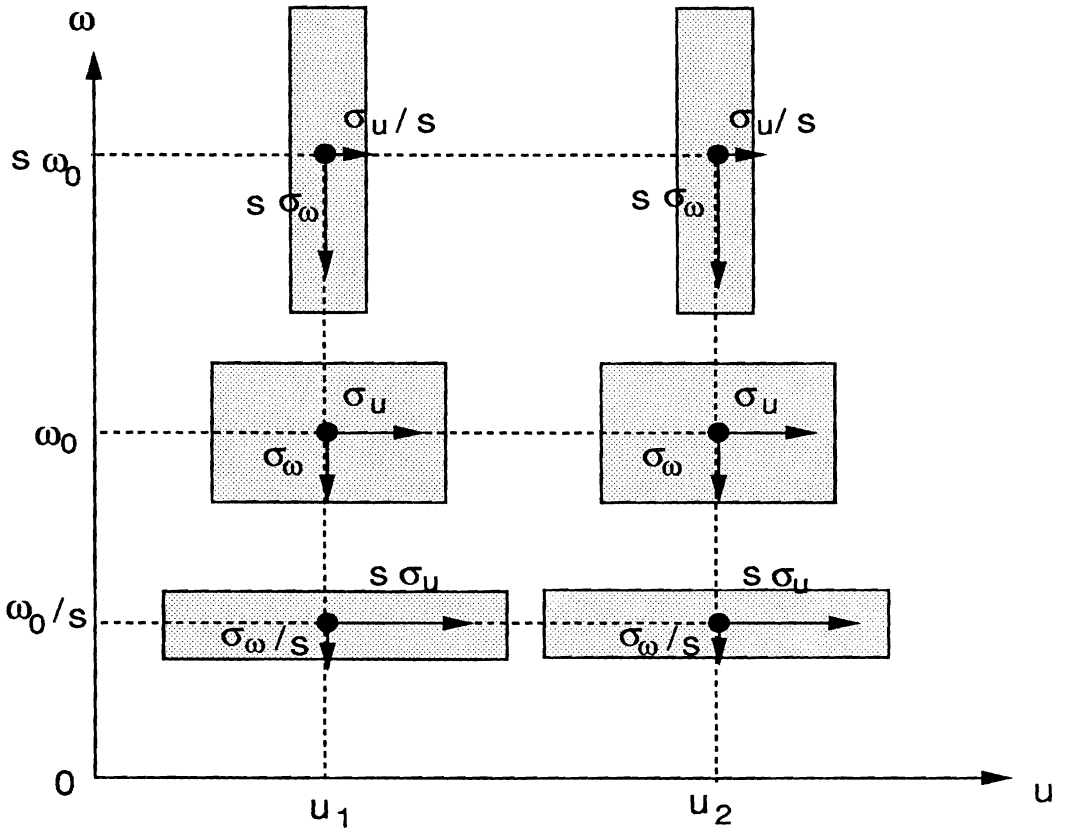

Figure 3: Phase-scale representation: σ_u and σ_ω are the standard deviations of $\psi(x)$ and of $FT(\psi(x))$.

From this point we can define a discrete wavelet transform and a wavelet decomposition for two-dimensional signals. This mathematical work has been made by Stephane Mallat. I recommend to read [2]"Multifrequency Channel Decompositions of Images and Wavelets Models", in which he shows all the properties of the wavelet transform: isometry, orthogonal basis, the ability to characterize the local regularity of a function and so forth...

I will not go into these details. I will only describe what we took from this work in order to build our wavelet decomposition.

3.3 The wavelet decomposition

The implementation of our wavelet transform on images led to a pyramidal multiresolution decomposition.

I must acknowledge that this kind of decomposition have been already developed by Burt [4]. They performed a Laplacian Pyramid with second derivatives of Gaussians. Our approach is very similar to theirs. Indeed we use filters that are first derivatives of gaussian-like functions. It means that the function $\psi(x)$ we chose is the first derivative of a smoothing gaussian-like function. This kind of filtering is indeed the easiest way to detect edges. Each local discontinuity provides a local extremum. However Burt used window filters that smoothed corners. The multi-orientation wavelet decomposition allowed us to get precise localization for each orientation. But the main advantage is rather that the wavelet transform is now based on solid mathematical proofs. It provides efficient algorithms and there is no increase in data storing while the decomposition is processed.

Yet, as I will explain later the behavior of detected edges through scale space does not depend coarsely on the filters we use. The rules that model the edge behavior in scale space use the assumption that the filters are gaussian or have the shape of gaussians. As I already said in "goals" section the main idea was to see how to use the scale space information to classify the edges and how to perform more intelligent segmentation with it. Consequently we consider the wavelet decomposition as an efficient tool only.

In order to have a closer look at the wavelet decomposition I would recommend the reading of N. Treil's paper [6] and S. Mallat's [2]. Now here is what I implemented:

I take the first derivative of a smooting function and make it the wavelet $\psi(x)$. If $L(x)$ is the smoothing function (a simple gaussian for the primary experiments) I can denote $\psi(x)$ by $G(x)$ and I can write:

$$G(x) = \frac{dL}{dx}(x)$$

I use a dicrete scale $s = 2^j$ and a finite number N of decomposed signals. The initial wavelet G will be my finest filter. $f(x)$ is the signal. The decomposition is called **dyadic decomposition** and is written:

$$\sum_{0}^{N-1} f(x) * 2^{-j} G(2^{-j}x) = \sum_{0}^{N-1} f(x) * G_j(x).$$

$(G_j(x))_{0 \leq j \leq N-1}$ are the dilated functions from G. Since G is the first derivative of L this decomposition can be rewritten:

$$\sum_{0}^{N-1} f(x) * 2^{-j} \frac{dL}{dx}(2^{-j}x) = \sum_{0}^{N-1} 2^j \frac{d}{dx}(f(x) * 2^{-j} L(2^{-j}x)).$$

This last formula shows exactly how we detect local variations at different scales. Indeed for each scale j the decomposed signal $f(x) * G_j(x)$ is obtained by smoothing f with $2^{-j}L(2^{-j}x)$ before the derivation. As $2^{-j}L(2^{-j}x)$ is 2^j times larger than L we can extract edges that are 2^j times "coarser" than those detected with L.

In order to deal with images, I use S. Mallat's two dimensional wavelet decomposition schema. Introducing only two directions of detection (X direction and Y direction) I come up to build 3 different two-dimension filters. They are all with separable variables. It allows us to compute line-filtering successively on rows and columns.

I denote $2^{-1}L(2^{-1}x)$ by H and get the three following filters:

- **I** $\quad G(x)L(y)$
- **II** $\quad L(x)G(x)$
- **III** $\quad H(x)H(y)$

I detects discontinuities in the X-direction and smoothes the signal in the Y-direction. This two-dimensional wavelet filter will provide vertical edges.
II detects discontinuities in the Y-direction and smoothes the signal in the X-direction. This two-dimensional wavelet filter will provide horizontal edges.
III smoothes in both directions and cuts half of the spectrum (its cut off frequency is $\frac{\pi}{2}$).

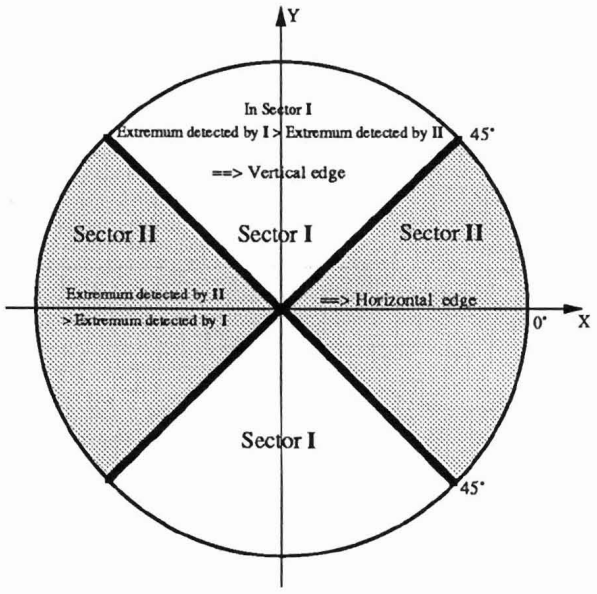

Figure 4: Detection sectors

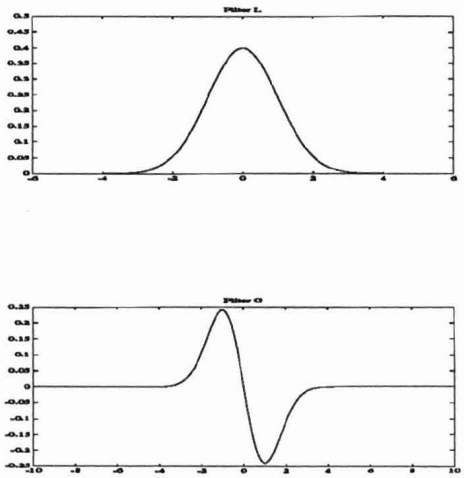

Figure 5: Gaussian model filters L and $G = \frac{dL}{dx}$.

Fig 6 shows the spectrum distribution of these filters. Fig 7 illustrates the actual decomposition. Sampling by 2 at each step does allow us to use the same filters for all the scale and performs an efficient algorithm with no increase of output data.

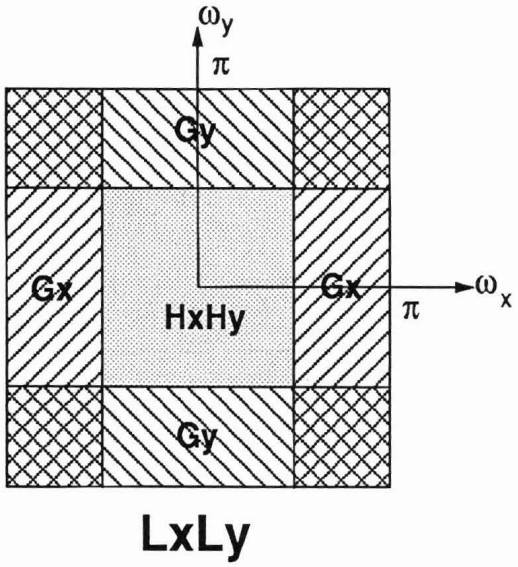

Figure 6: **Spectrum distribution of the filters**

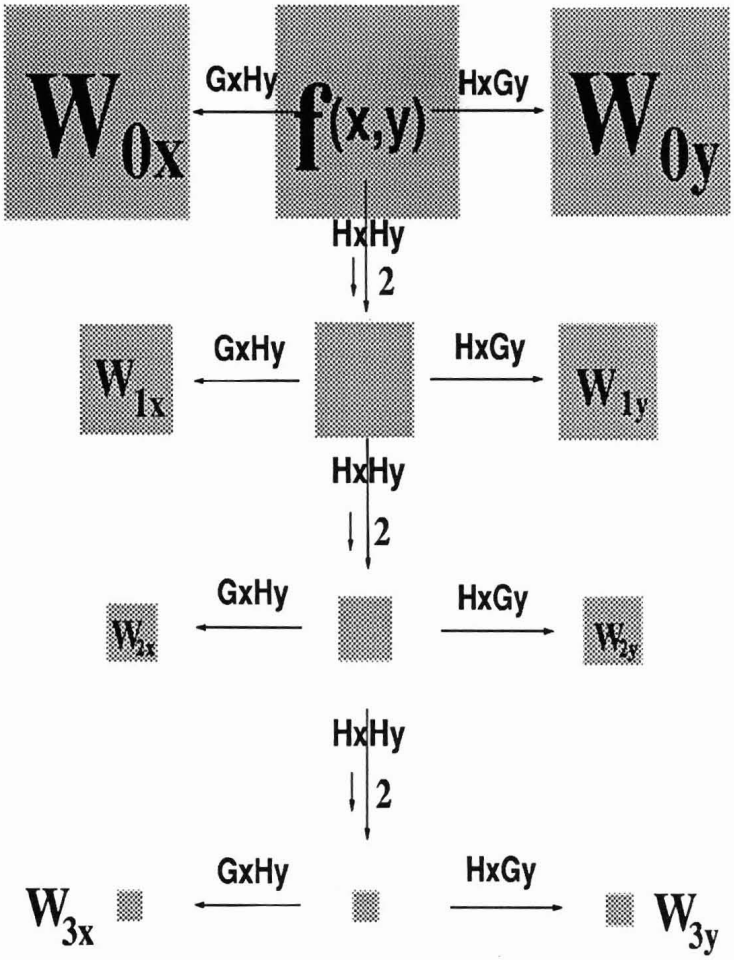

Figure 7: Multi scale decomposition on horizontal and vertical direction.

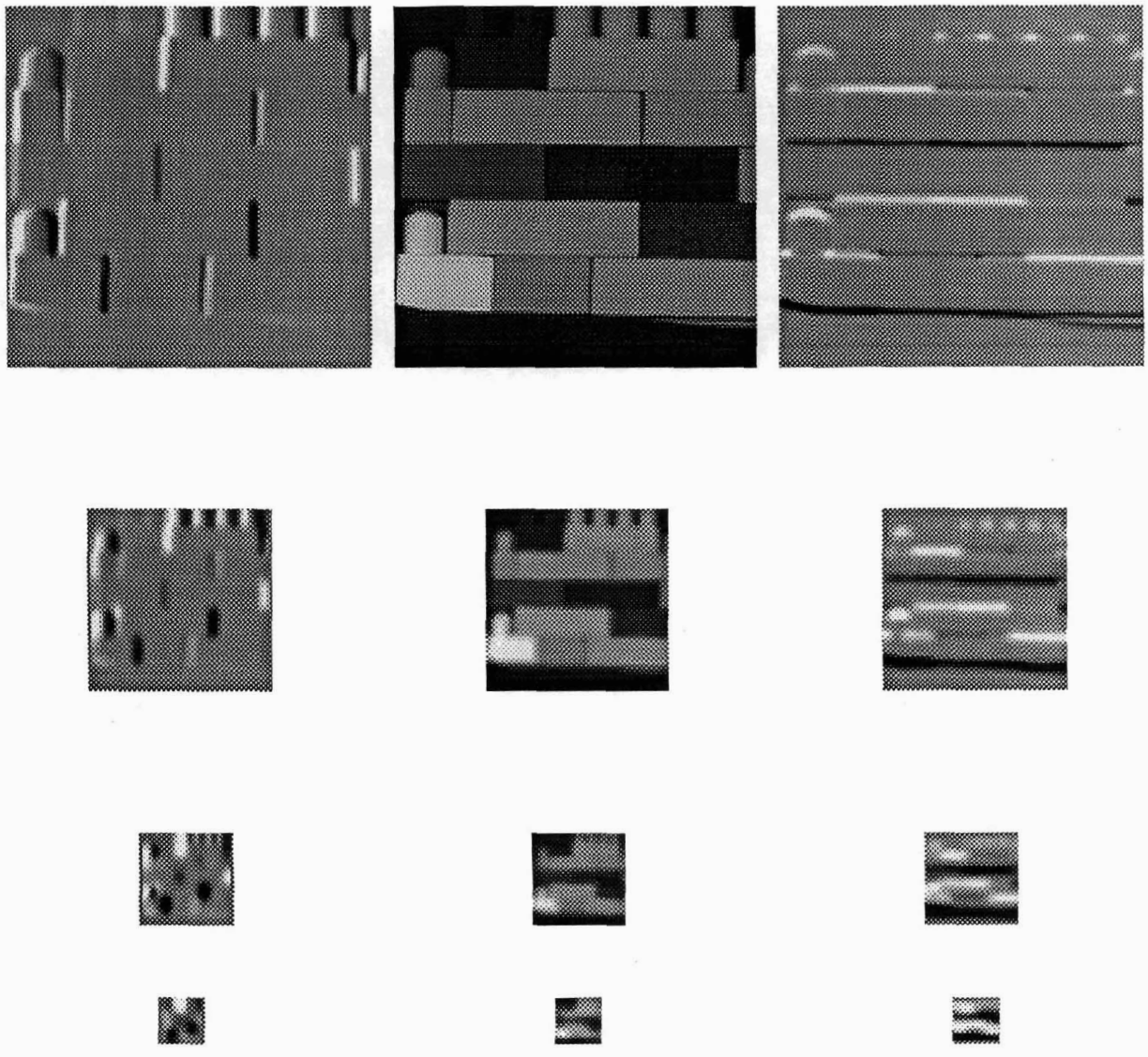

Figure 8: Wavelet decomposition of the image of a wall.

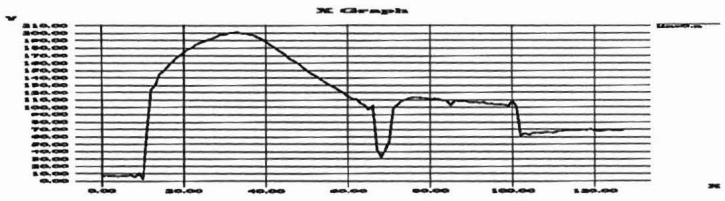

Figure 9: **Scan line** $f(x)$

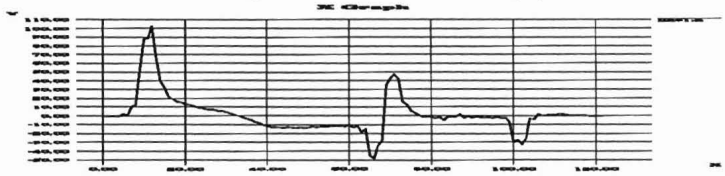

Figure 10: $W_0(f)(x)$

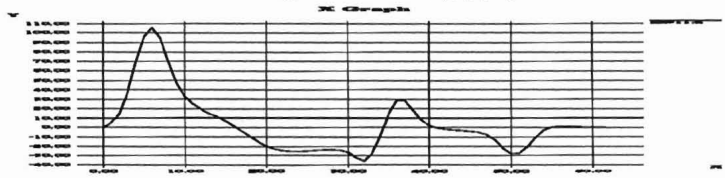

Figure 11: $W_1(f)(x)$

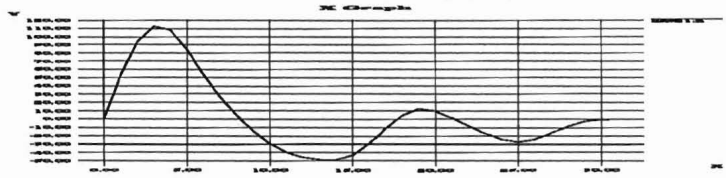

Figure 12: $W_2(f)(x)$

Figure 13: $W_3(f)(x)$

4 Coarse to fine tracking

4.1 The edge behavior

The multi orientation decomposition allows us to track the edges along lines and columns. The problem restricts itself to a one-dimemsional signal processing.

Indeed let us consider one horizontal scan line i. G will be applied on it (one dimension filtering). When all the lines are processed, L is applied on the columns. It smoothes in the perpendicular direction that is in this case the vertical direction. We can easily consider that this effect is negligible compared to the effect of the first derivative filter on the line. Therefore we can follow the result of the successive filtering through the scales. As you can see in Fig 9 to Fig 13, each scan line (upper signal) is related to N scan lines taken from the N wavelet images.
I denote the scan line by a one dimension signal $f(x)$.
I denote the wavelet transform of this scan line at each scale i by $W_i(f)(x)$.

In order to perform a coarse to fine tracking I take all the extrema at the coarsest scale $W_3(f)(x)$ and link them to those of the next scale according to some rules. I repeat the same operation at scale 2 and so forth.. up to the finest. However this operation is not obvious. As we are going to see, the edges interact and their location is shifted as we go down the scales. Therefore the tracking must use a window of detection. In other words the location of the same edge is allowed to move along the line when going down the scales. Besides some edges can merge through scale space too. I studied these problems and tried to reduce their influence.

When this tracking is done, I can take all the edges detected at the finest scale and count how deep they are connected through scale space. The more the edge appeared up the scales, the more likely it corresponds to the outline of a coarse objet and the more important this edge is. On the contrary an edge that is not much connected and thus disappeared quickly in scale space is due to high frequency texture, highlight speckles or just small objects. Thus that will be our measurement of the degree of importance.

4.1.1 Ideal and single edges

In order to get the signature of an edge through scale space, I performed some simulation with ideal edges.

In the following formulas I will not do any sampling in order to simplify the demonstration.

Let us denote the scan line by $f(x)$.

Let us denote the dyadic wavelet transform of $f(x)$ at scale j by $W_{2^j} f(x)$.

$$W_{2^j} f(x) = f(x) * G_j(x) = 2^j \frac{d}{dx}(f(x) * 2^{-j} L(2^{-j} x))$$

Let us consider $f(x)$ as the signal of a single edge along the scan line. Locally, the image intensity can be modeled by the convolution of a discontinuity (or singularity) with a smoothing function. $f(x) = d(x) * l_{\sigma_0}(x)$, where $d(x)$ is singular in x_0 and σ_0 is the standard deviation of the smoothing function.

The wavelet transform of $f(x)$ is given by:

$$W_{2^j} f(x) = 2^j \frac{d}{dx}(d(x) * l_{\sigma_0}(x) * 2^{-j} L(2^{-j}x)).$$

Let us imagine now that L and l are both gaussians.

If the standard deviation of $L(x)$ is σ_1 the standard deviation of $2^{-j}L(2^{-j}x)$ will be $2^j\sigma_1$.

Subsequently we can write:

$$W_{2^j} f(x) = 2^j \frac{d}{dx}(d(x) * l_{\sigma_0}(x) * l_{2^j\sigma_1}(x)) = 2^j \frac{d}{dx}(d(x) * l_{\beta\sigma_1}).$$

$$\text{with } \beta = \frac{\sqrt{\sigma_0^2 + (2^j\sigma_1)^2}}{\sigma_1}.$$

The wavelet transform $W_{2^j} f(x)$ can be rewritten:

$$W_{2^j} f(x) = \frac{2^j}{\beta} W_\beta d(x).$$

One can easily show that we always have

$$\beta = \begin{cases} 2^j & \text{if } 2^j\sigma_1 \ll \sigma_0 \\ \frac{\sigma_0}{\sigma_1} & \text{if } \sigma_0 \ll 2^j\sigma_1 \end{cases}$$

we can therefore distinguish two domains:

- If $2^j\sigma_1 \gg \sigma_0$ then $\beta \approx 2^j$ so

$$W_{2^j} f(x) = W_\beta d(x).$$

In this range of scale, the dyadic wavelet transform is not sensitive to the smoothing of the edge. It behaves as if there were a strict singularity in x_0. In his work, S. Mallat found a way to characterize the local regularity of such an edge. Shortly, if the singularity of $d(x)$ is Lipschitz α in x_0 (if there exists a polynomial $P_n(x)$ of order n such that for all x in a neighborhood of x_0, we have $\mid f(x) - P_n(x) \mid = O(\mid x - x_0 \mid^\alpha)$) the amplitude of the extrema $W_{2^j} d(x_0)$ is $O(2^{j\alpha})$. Actually the experiments with real images overshadow this characterization. I notice that this rule is not robust when dealing with more than one edge per scan line. In comparison with the second case the extrema amplitude $W_{2^j} d(x_0)$ does not change as much through scale space. That is why I group these kind of edges in the category of "normal edges". As we are going to see they represent most of the edges in real scenes (see Fig 14).

- If $2^j\sigma_1 \ll \sigma_0$ then $\beta \approx \frac{\sigma_0}{\sigma_1}$ so

$$W_{2^j} f(x) = \frac{2^j}{\frac{\sigma_0}{\sigma_1}} W_{\frac{\sigma_0}{\sigma_1}} d(x)$$

In this range of scale, the dyadic wavelet transform $W_{2^j} f(x)$ increases like 2^j. However there is always a j_0 from which $2^j \sigma_1 \not\ll \sigma_0$. In order to have an approximation of the edge width (that is σ_0), we just have to watch when the increase stops (see Fig 15).

In Fig 14 and Fig 15 the effect of the discretization and the effect of the sampling are not forgotten. These are the simulations of what the actual algorithm provides.The results confirm the mathematical simulation despite the strong assumption that the wavelets are first derivative of gaussians. Actually these behaviors can be observed provided that the wavelet is the first derivative of a smoothing function.

Fig 16 corresponds to a third case: the ridge behavior. As you can see two effects characterize the ridge. The extrema go apart from each other when going up the scale. On the contrary the amplitude of the extrema decreases drastically.This last effect is all the more important as the ridge is narrow. Besides the quantification and sampling errors do not help to model this behavior.

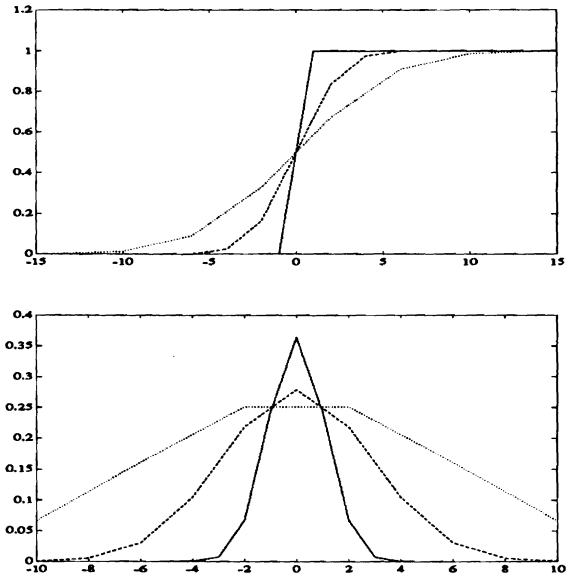

Figure 14: Top graph: step-edge and its successive smoothed versions. Bottom graph: successive wavelet transforms of the step-edge ($W_{2^0} f$, $W_{2^1} f$, $W_{2^2} f$).

plain line: scale 2^0.
thick dashed line:scale 2^1.
thin dashed line:scale 2^2.

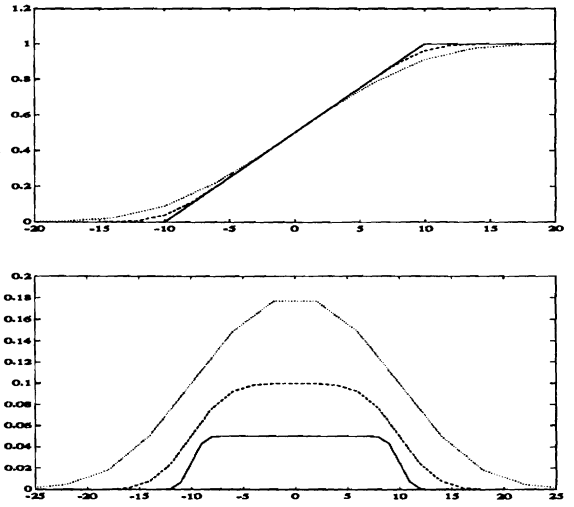

Figure 15: Top graph: wide-edge and its successive smoothed versions. Bottom graph: successive wavelet transforms of the wide-edge ($W_{2^0}f$, $W_{2^1}f$, $W_{2^2}f$)

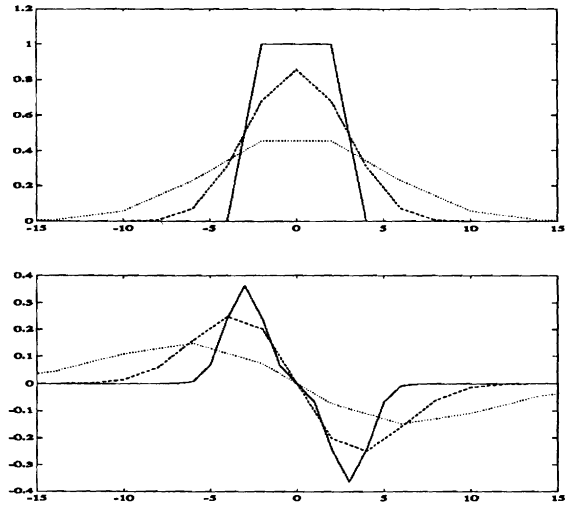

Figure 16: Top graph: ridge and its successive smoothed versions. Bottom graph: successive wavelet transforms of the ridge ($W_{2^0}f$, $W_{2^1}f$, $W_{2^2}f$)

plain line: scale 2^0.
thick dashed line:scale 2^1.
thin dashed line:scale 2^2.

4.1.2 The edge behavior in real images

The fact is that in real scan lines we cannot isolate an edge and watch its behavior. We must take into account the notion of "competitive" edges. As Fig 17 illustrates, the small stair disappears at scale 2^2. Compared to the bigger stair it is a detail and as such it cannot appear in the same amount of scale. However reading the evolution of the extrema for both stairs is not obvious any more. The information is shared by these two. This is a typical example of edge merging.

Besides, when I track the extrema from a coarse scale scan line to the next finer scan line I must search in a certain neighborhood around the assumed position of the edge. The merging problem and the ridge behavior are the main reasons for these delocalizations. Subsequently it brings another difficulty. When from one edge two new edges come out at the next finer scale, we must choose the one that will be connected and therefore that will correspond to the coarse edge and the one that will be considered as a new edge which answer in scale space just vanished. In our example Fig 18 , the choice will be easy because the two edges do not provide extrema with the same amplitude. Nevertheless tricky situations can occur. That is why the algorithm can take sometimes some arbitrary decisions that lead to wrong attributions.

Fig 18 shows what merging through scale space means. We can see that the use of dyadic wavelets provides us with a set of discrete scales. We literally work with a set of distant slices in the scale space. It prevents us from following exactly the delocalization of the edges with a continuous coarse to fine tracking. So linking one edge at scale 2^j to another one at scale 2^{j-1} can sometimes lead to wrong attributions or connections.

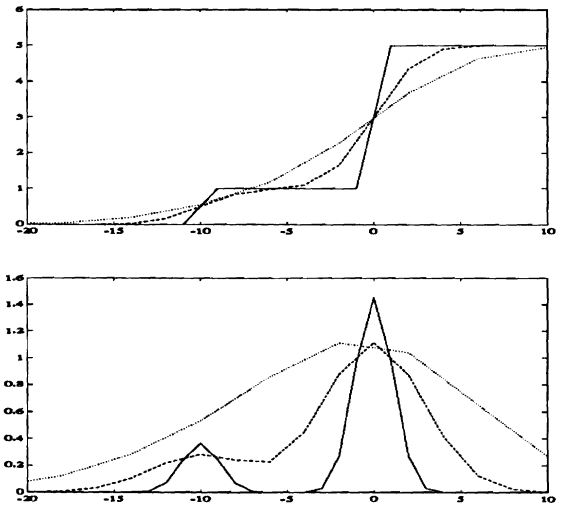

Figure 17: Top graph: two edges and their successive smoothed versions. Bottom graph: successive wavelet transforms of the two edges ($W_{2^0}f$, $W_{2^1}f$, $W_{2^2}f$)

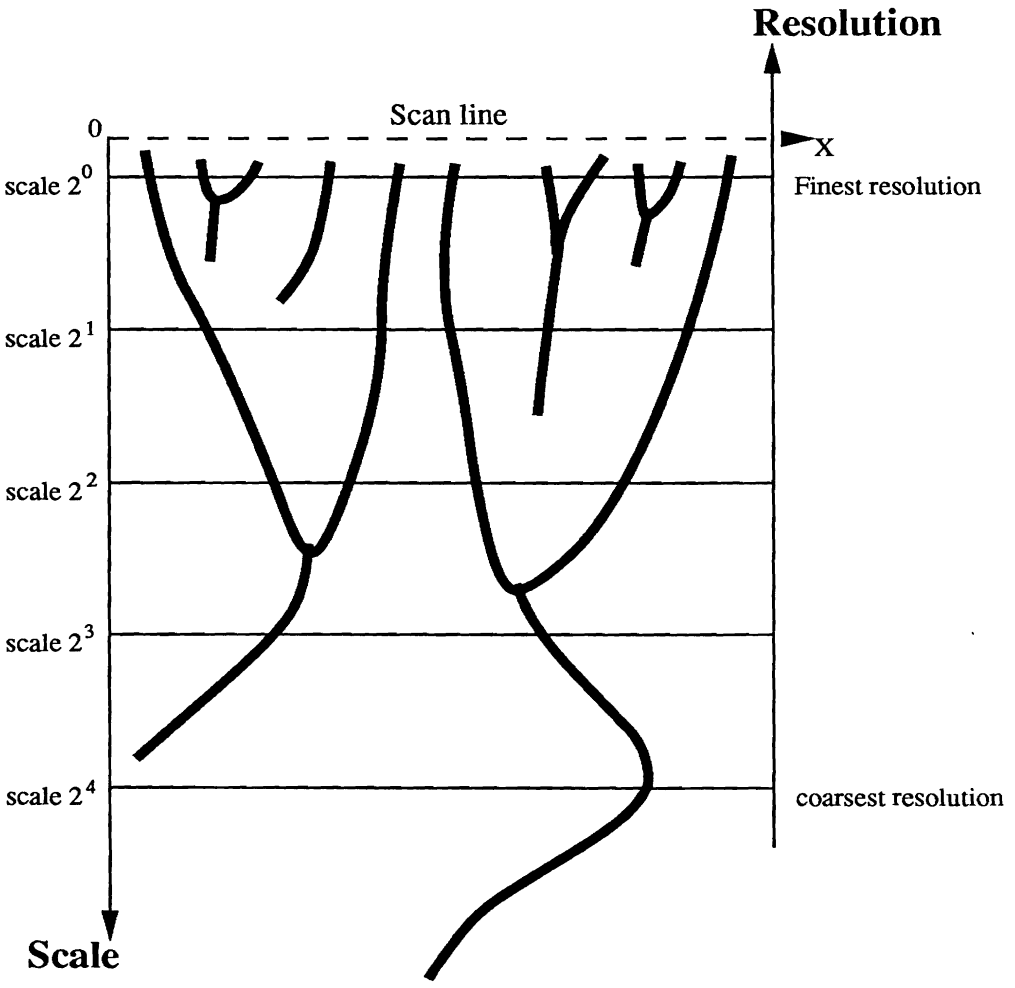

Figure 18: Edge merging through scale space.

All these reasons make me build a robust tracking based on few simple rules. I ignore all the rare possibilities of merging and of strange behavior on purpose. It would increase the degree of arbitrariness anyway. The final good answer is indeed often readable only on the real scene with our own vision system. There are still a lot to do to determine how and where to intervene in the tracking process in order to perform a feed-back system dealing with error calculation. Nevertheless I will show later how we can work on the interpretation of the output of this process to remove some errors.

So I extract 3 different edge behaviors, as the mathematical simulation tells me to do:

- The first one is the "normal" behavior. The extrema value does not change drastically through scale space. The edge is certainly narrow (at least compared with the standard deviation of the finest filter). The step-edge belongs to this category.

- The second one is the behavior of wide edges. From one scale to the other the extrema value doubles. The edge is wide, maybe a very slow transition drowned in noise. The scale from which this behavior stops, gives an approximation of the edge width. The wide-edge in Fig 16 belongs to this category.

- The third one is the typical behavior of ridges. From one scale to the other the extrema value decreases a lot. The narrower the ridge is the more important the decrease is. (see Fig 16)

If we have a look back at Fig 9 to Fig 13 we can actually see these different behaviors. The first maximum from the left is due to a normal edge. The first minimum from the left is due to a wide edge. The third and fourth extrema are due to a ridge.

4.2 The tracking process

This process is divided in three stages: the Edge Tracking and Linking through scale space, the Edge Signature Computation and the Symbolic Merge. (see Fig 19)

4.2.1 Stage 1: The Edge Tracking and Linking through scale space

The coarse-to-fine tracking procedure uses the rules I just mentioned. In order to deal with the shift in position I define a search-window. To link one extremum to the one at the next finer scale I check all the extrema in this search-window and connect one of them with the coarse one if it fits one of the 3 behaviors I extracted. This window is two-dimensional in order to handle the sampling effect among the lines. (see Fig 19) This process is performed twice, one time on the decomposed images $(W_{ix})_i$ and the second time on the decomposed images $(W_{iy})_i$. Actually in order to simplify the linking process I extract all the extrema of these decomposed images and store them in a structure (lists of pointers). Then the tracking process manipulates these lists of pointers.(see Fig 21)

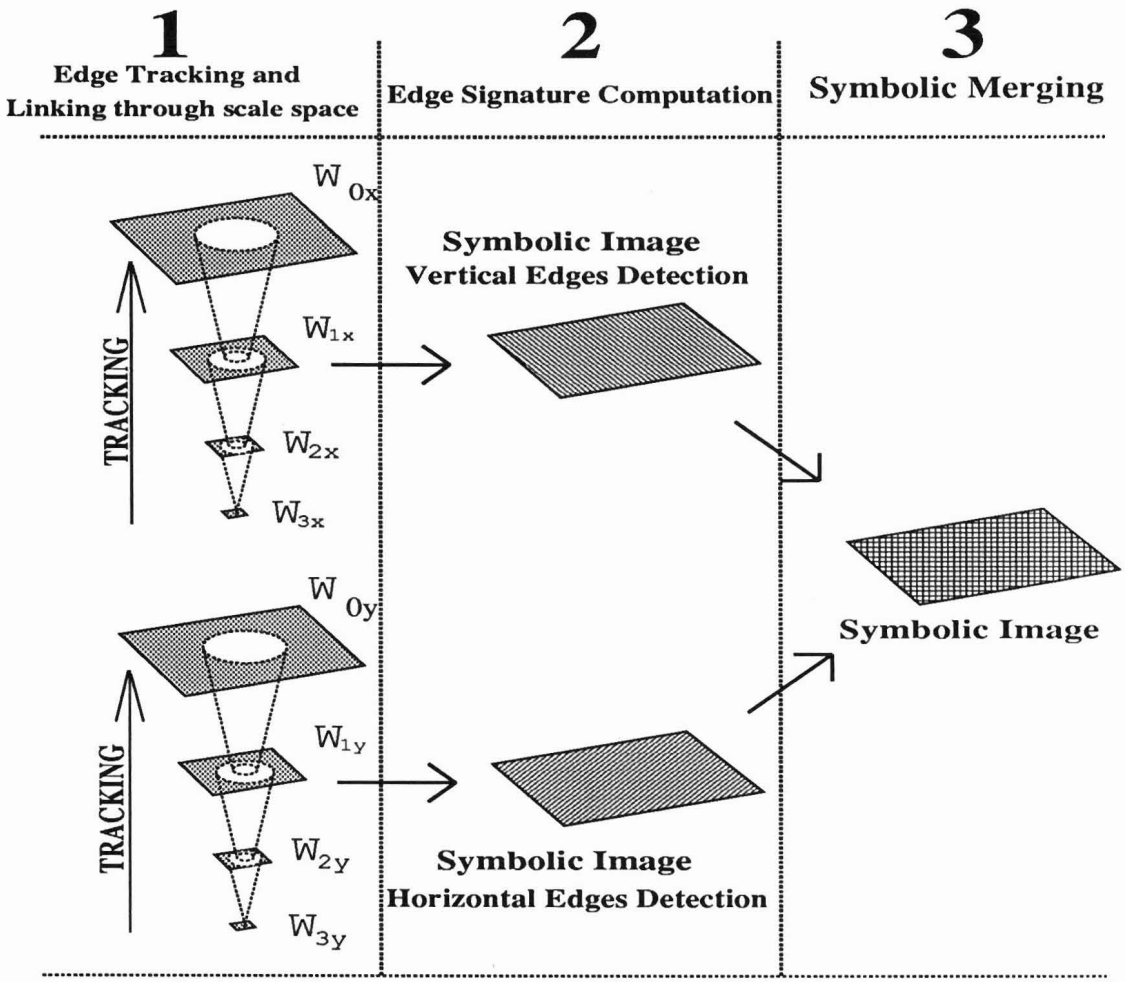

Figure 19: The 3 stages of the edge detecting process.

4.2.2 Stage 2: The Edge Signature Computation

Once the tracking is done, I can take all the extrema detected at the finest scale 0 and stored in the structure relevant to the W_{0x} and W_{0y} , count how deep they are connected through scale space with coarser extrema elements and subsequently assign to them the result of this reckoning.

In order to store this piece of information I create two symbolic images, one for each direction of detection (see Fig 19). Each pixel is assigned the fact that it is an edge-pixel or not. If it is not an edge-pixel I assign a null to it. If it is an edge-pixel, the degree of importance and the extremum value at the finest resolution are stored as shown in Fig 22. Besides while looking how deep the extremum is connected through scale space I can read the succesive extremum values. If there is an increase, I can say that my edge-pixel is due to a wide edge. This piece of information is coded too. There is some space for other pieces of information in the code. We will see later how it has been used and could be used in further developments.

4.2.3 Stage 3: The Symbolic Merge

As the result of stage 2, we have two symbolic images. The image S1 that comes from the $(W_{ix})_i$ displays information on the edges that are more or less vertical. The image S2 that comes from the $(W_{iy})_i$ displays information on the edges that are more or less horizontal. In order to get a single symbolic image we must merge these two. If a pixel is an edge-pixel in S1 (respectively S2) and not in S2 (respectively S1) we just take the code from S1 (respectively S2). If a pixel is declared edge-pixel in S1 and in S2 too we decide which code is to be taken according to the rule described in Fig 4 page 11. This method removes much of the redundancy between S1 and S2. I will show the example of circular objects for which the arcs of a circle in S1 and S2 merge very well. However this method has a tendancy to give what we call an over-estimation in the contour detection. (see Fig 20 where this effect is exaggerated on purpose)

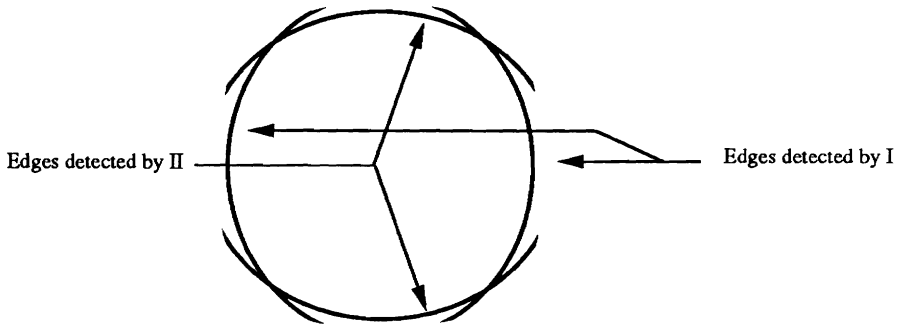

Figure 20: The over-estimation effect after symbolic merging.

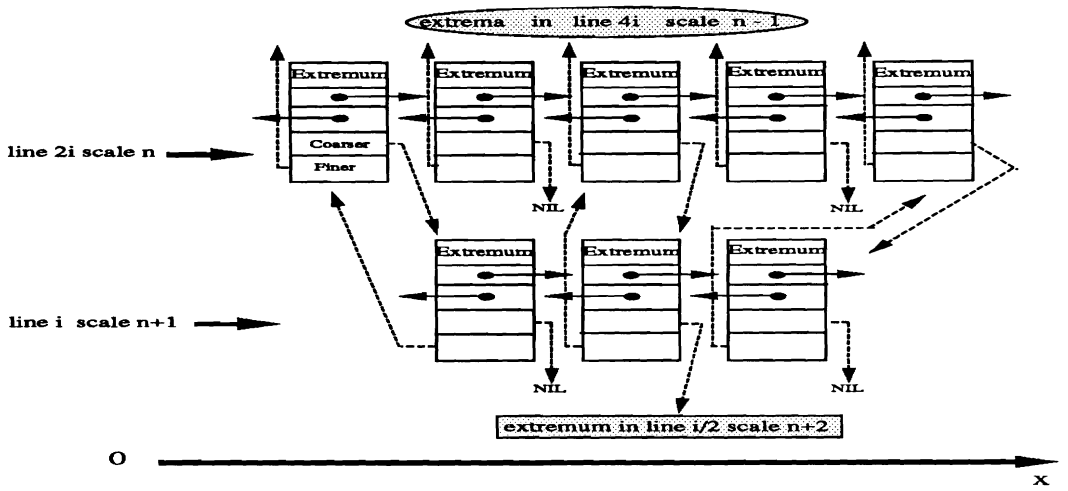

Figure 21: Intermediate structure on which the tracking and the linking are performed.

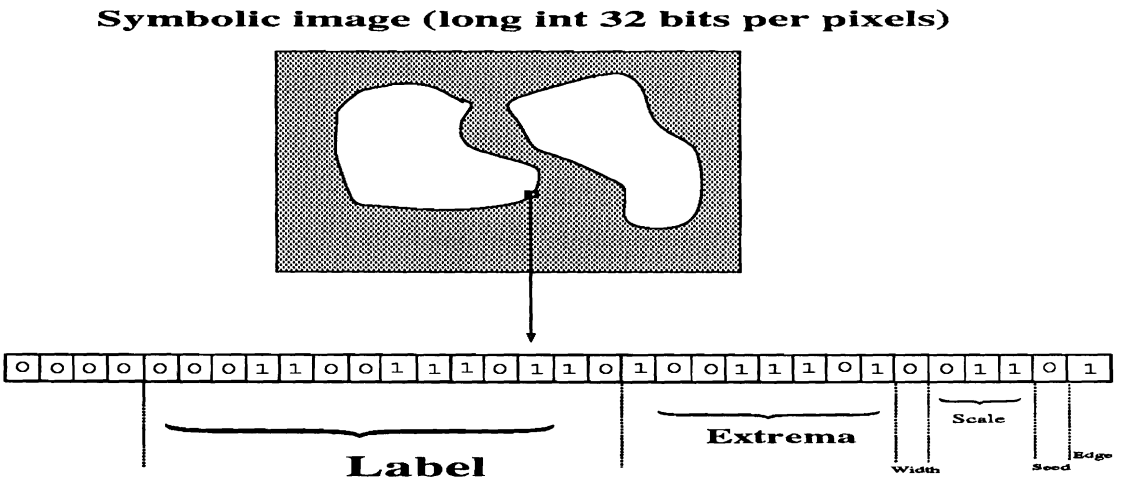

Figure 22: How the edge signature is coded.

Edge bit : 0 this is not an edge-pixel, 1 this is an edge-pixel.
Seed bit : 0 this is not an seed-pixel, 1 this is an seed-pixel.
Scale bits : number between 0 to 7 of the coarsest scale where the edge still exists.
Width bit : 0 normal edge, 1 wide edge.
Extrema bits : extremum value between 0 and 255 at the finest resolution.
Label bits : used to link the edge-pixels or the seed-pixels in the spatial domain in order to get labelled lines and contours.

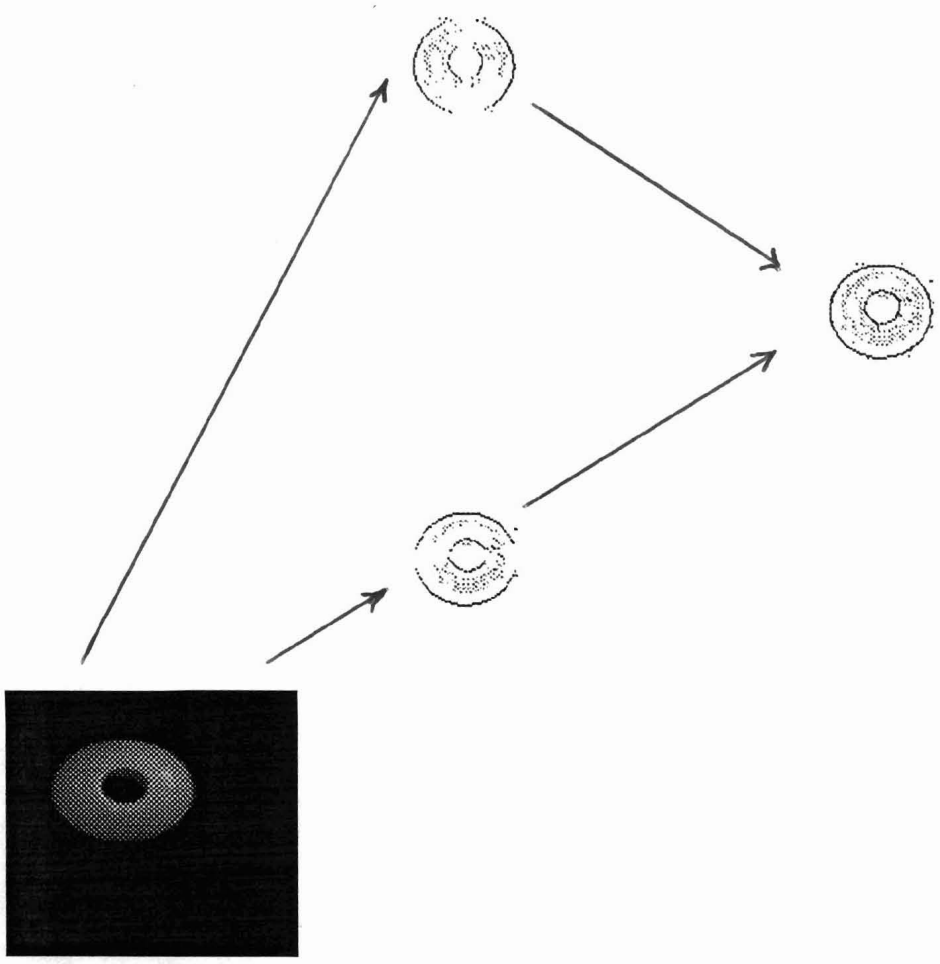

Figure 23:

Left: Image on which the detection is to be performed.
Middle: The two symbolic images at stage II.
Right: Single symbolic image after merging (stage III).

5 Interpretation and Edge Classification

The deal is now to interpret the symbolic image. The edge signature tells us if an edge is very contrasted or not (extremum bits), if an edge belongs to a coarse feature or to small details (scale bits) and if an edge is sharp or very wide (width bit). But from now on we want to determine the physical properties of the edge, in other words the physical causes of the edge.

5.1 Physical causes of an edge

Here are the main causes:

- Material discontinuity (color discontinuity or texture change).
- Color discontinuity due to highlights.
- Shadow discontinuity.
- Shading.
- Orientation discontinuity (singularity in the surface of an object).
- Depth discontinuity (the border of a cylinder for example).

First I took grey scale images as inputs. The idea was that we could find a lot of clues to classify the edges from brightness images. We got a good discrimination between large objects in the scene and details inside these objects. The texture was detected as high frequency edge components whereas its contour was extracted as coarse features.(see Fig 28) However a lot of ambiguities still remained.

The width parameter should have given us a way to discriminate between the edges coming from sharp material discontinuities or orientation discontinuities and the edges coming from depth discontinuities or shadows. It turned out that this parameter does not work all the time. The width of a shadow-edge depends on the illumination conditions and the distance between the camera and the scene. The scale information cannot help to distinguish highlights from real objects every time. A highlight can be very small and yet sometimes can be nearly as large as the surface of the object.

Consequently it became necessary to use color information. That is why this edge detector is applied on 3-plane color images.

5.2 The Color Images

It is difficult to apply a good segmentation on simple RGB images. We need a color space that have meaningful vectors of representation. We need color constancy. Sang W Lee [7] performed a color space transformation that leads to a space where the Z-axis is the brightness axis and where the orthogonal plane to this Z-axis is the hue plane. (see Fig 24). This transformation is based on the spectral reflexion of illuminants (or light sources) on a white panel.

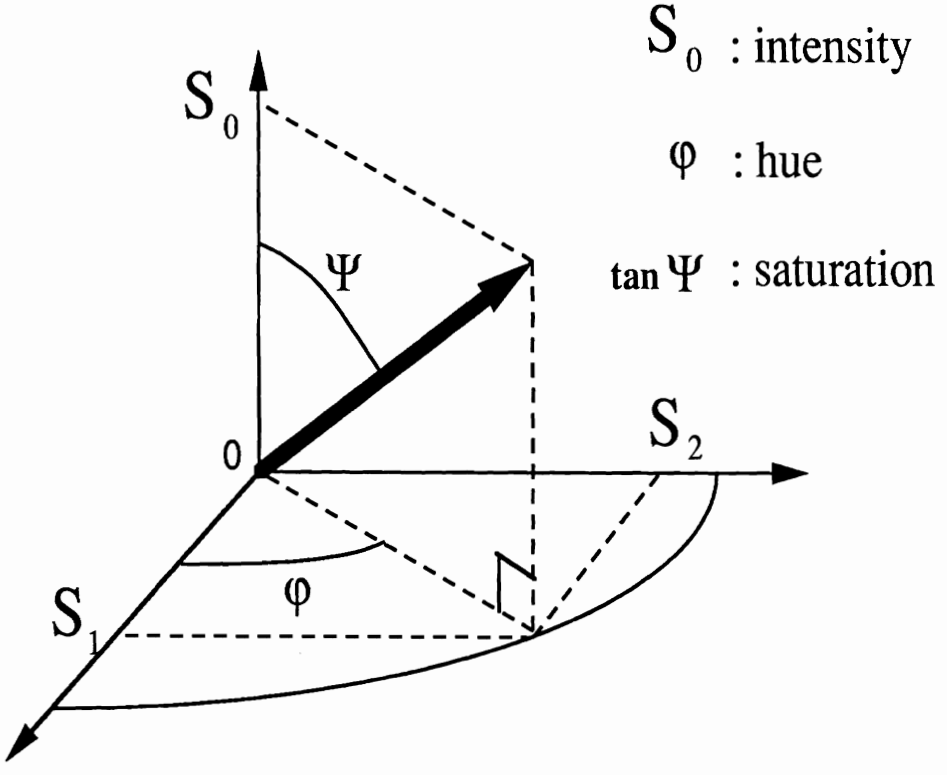

Figure 24: Color Space that gives IHS 3-plane images.

As the result of these color processings we get IHS 3-plane images (I for intensity, H for hue, S for saturation). The edge detector will be applied on these 3 planes.

$$I = S_0$$

$$H = \arctan \frac{S_2}{S_1}$$

$$S = \frac{\sqrt{S_1^2 + S_2^2}}{S_0}$$

Let us describe how we can detect highlight, shadow and shading in color space when we assume that the color constancy is perfect.

Let us denote the body color of an object by its co-ordinates in color space S_0, S_1, S_2.

Highlight $S_0, S_1, S_2 \Longrightarrow S_0 + S_H, S_1, S_2$
$I, H, S \Longrightarrow I + S_H, H, S\frac{S_0}{S_H + S_0}$

Shading $S_0, S_1, S_2 \Longrightarrow \lambda(x)S_0, \lambda(x)S_1, \lambda(x)S_2$
$I, H, S \Longrightarrow \lambda(x)I, H, S$

Shadow $S_0, S_1, S_2 \Longrightarrow \lambda S_0, \lambda S_1, \lambda S_2$
$I, H, S \Longrightarrow \lambda I, H, S$

Therefore some edges will not appear in all three images. A highlight spot will be seen only in the I image and in the S image. Shading and shadow will be seen only in the I image. (see examples fig 25 and 26).

We have now many clues that should allow us to classify our edges according to their physical properties and to their degree of importance. Yet, as we are going to see, a lot of ambiguity are still not removed.

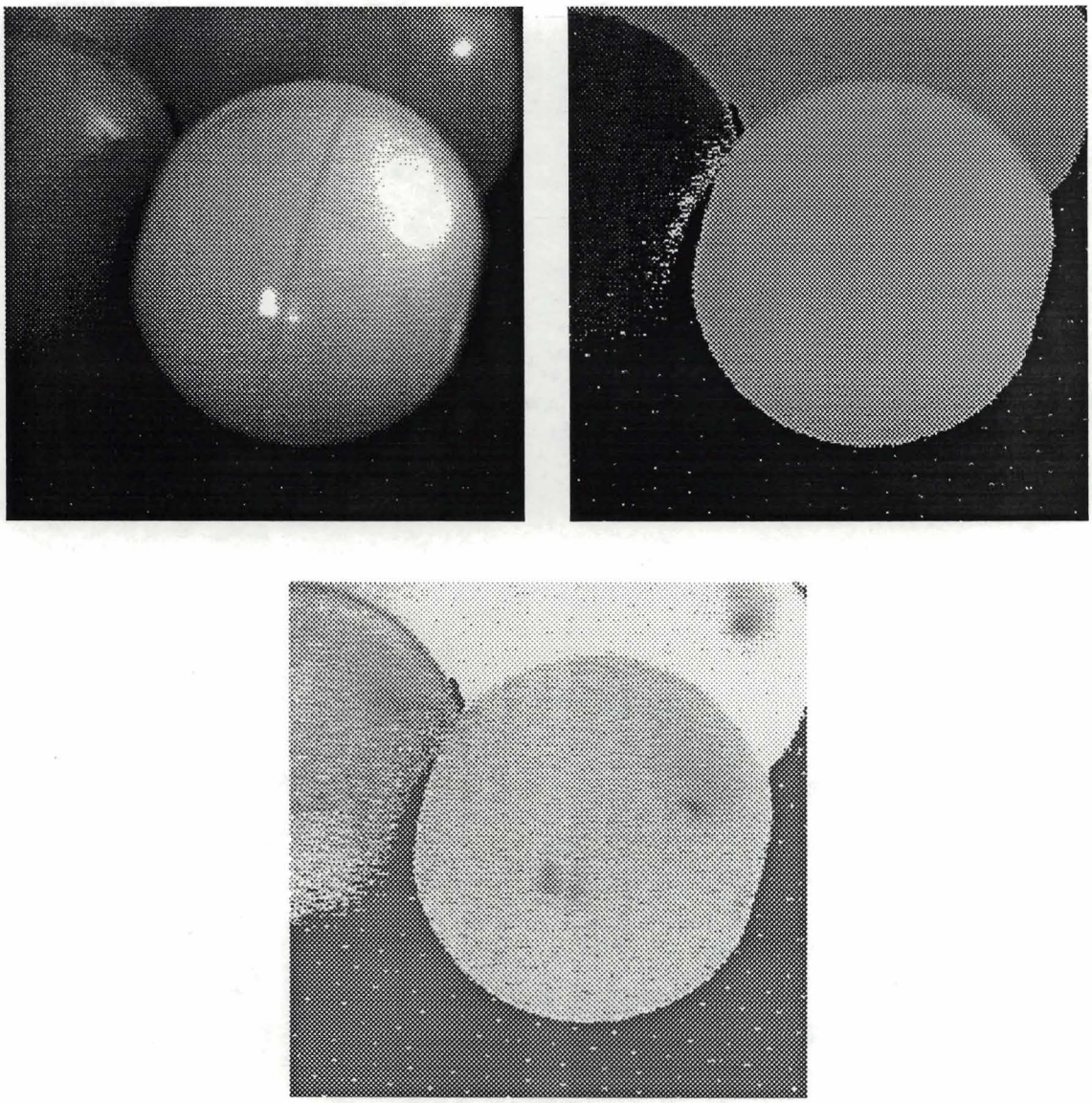

Figure 25: Top left: Intensity image. Top right: Hue image. Bottom: Saturation image.

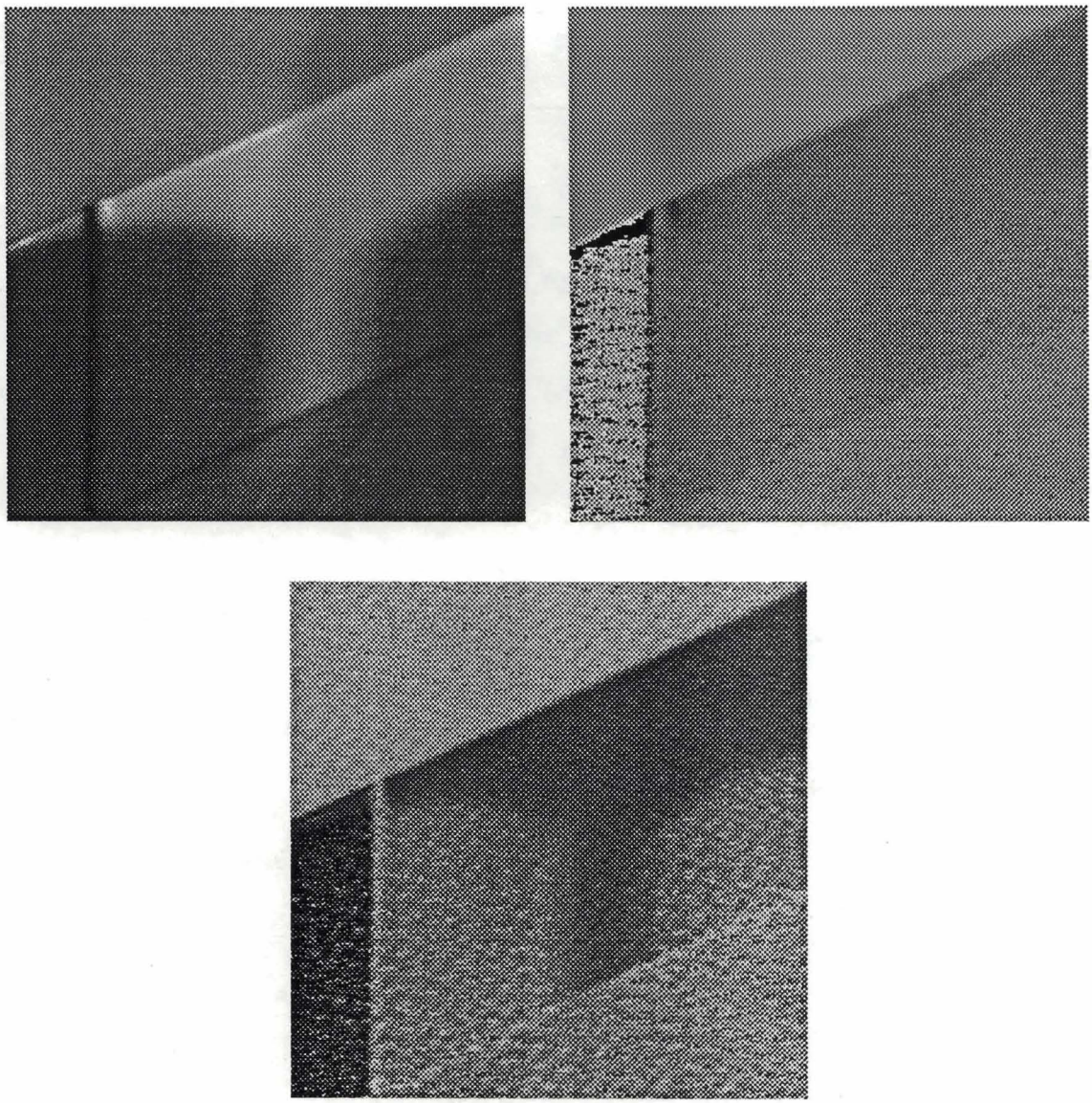

Figure 26: Top left: Intensity image. Top right: Hue image. Bottom: Saturation image.

6 Results and Comments

6.1 Interpretation of the outputs

The detector has been applied on many images taken with a CCD camera (Sony XC77). The environment was as described in the first figure of this paper: several objects lay in an operating table, illuminated with several several light sources. These objects can have different body colors, texture and smoothed surfaces.

In all the following examples I display exactly what the screen output is. The top image is the real image. The bottom left image displays the scale information extracted from the symbolic image. It means that the 3 scale bits are only shown. The brighter the pixel the higher the degree of importance. As I used 4 scales most of the time, I got 4 different grey levels from black to white. The bottom right image displays the extrema bits. It is the real value of the extremum detected at the finest scale. It is scaled between 0 and 255 and it is a signed value. It gives information about the contrast range of the edges. The brighter or darker the pixel is, the bigger the discontinuity relevant to this edge-pixel is.

Here are the advantages and positive results we got:

- We can see in Fig 27 that the corners are very well detected. There is no blurring effect.
- We can see in Fig 29 that the over-estimation effect does not provide jagged contours.
- The scale information makes this edge detection definitely more meaningful in all the examples but especially in Fig 28. The contour of periodic patterns and grids are extracted as coarser features, whereas the very contrasted edges inside the texture are considered as details and high frequency components. The context in the scene has been therefore extracted.
- We can see in Fig 30 that a simple thresholding process applied on the contrast information could fail if the arbitrary threshold was too high. The big ball contour is considered as a coarse feature as a whole (see bottom left) but the contrast range varies a lot along the contour and becomes very low along the top part of the ball. The need to decompose an image through scale space turns out here to be just necessary.

Now let us see what is still wrong and not perfect:

- The over-estimation effect explains why the bottom part of the ball contour in Fig 30 is not neat. When the edge is not a straight line or a nice curve the vertical detector and the horizontal detector do not provide edges that merge nicely in the final symbolic image. It creates these stairs where the over-estimation effect is very important. (see Fig 29 too)

- The color information is not as reliable as we would expect every time. First the color constancy is perfect only if we have one kind of illuminant. In the case where there are several colored illuminants the highlights for example will not behave as described in page 29. Therefore they can still leave some traces in the hue plane.

- In Fig 31 we can observe that the shadow did not behave as I describe in page 29. The body hue of the bricks is not much affected by the shadow but the saturation increased drastically in the shadowed area. The explanation is that the bricks are glossy and thus provide a certain amount of highlight. This highlight is removed by the shadow and subsequently the saturation increases.

- In the hue image (Fig 30) we can observe a inter-reflexion effect between the ball on the left and the ball in the middle. The bright one is actually yellow, and the other one is bright blue. The yellow ball creates a shadow on the blue one and the hue in the shade turns a little to green. Besides, this greenish blue is the hue relevant to 0^o in the hue circle. That is why this region has some hue-pixels that correspond to small angles $\approx 0^o$ and some others that correspond to big angles $\approx 360^o$. 0^o pixels are black, 360^o are white. Finally this phenomenon and the edge merging effect induced the distorsion of the large yellow ball contour.

- We noticed that the digitized images coming from the CCD camera were very noisy.(see Fig 29 and 30) For the moment, there is no implemented pre-process that increases the SNR. However this work should be done. Actually we deal with two different noise: a gaussian uniform noise and a special noise that comes from the camera (see the periodic spots in Fig 30) A good modeling and removal of these two noises could improve the results we have.

- A busy scene like Fig 32 shows lots of example of edge merging. We can see competitive edges that are very close to each other and that obviously have merged in scale space. Consequently they have been attributed different scale bits. Yet we would have given them the same degree of importance very easily. In Fig 29 the small highlight on the top right of the image is a coarse feature, whereas the wide highlight on the big yellow ball has a contour with different degrees of importance. The reason is that the small highlight is isolated in a dark area and as such provides edges in all the scales. On the other side the large one is included in a bigger object (the yellow ball) and is close to the ball contour. The fact is that the tracking process uses simple rules applied on signals. Our comprehension of the world uses what seems to be upper-level mechanisms. In that extent, we are still far from performing a perfect scale classification as the human vision does.

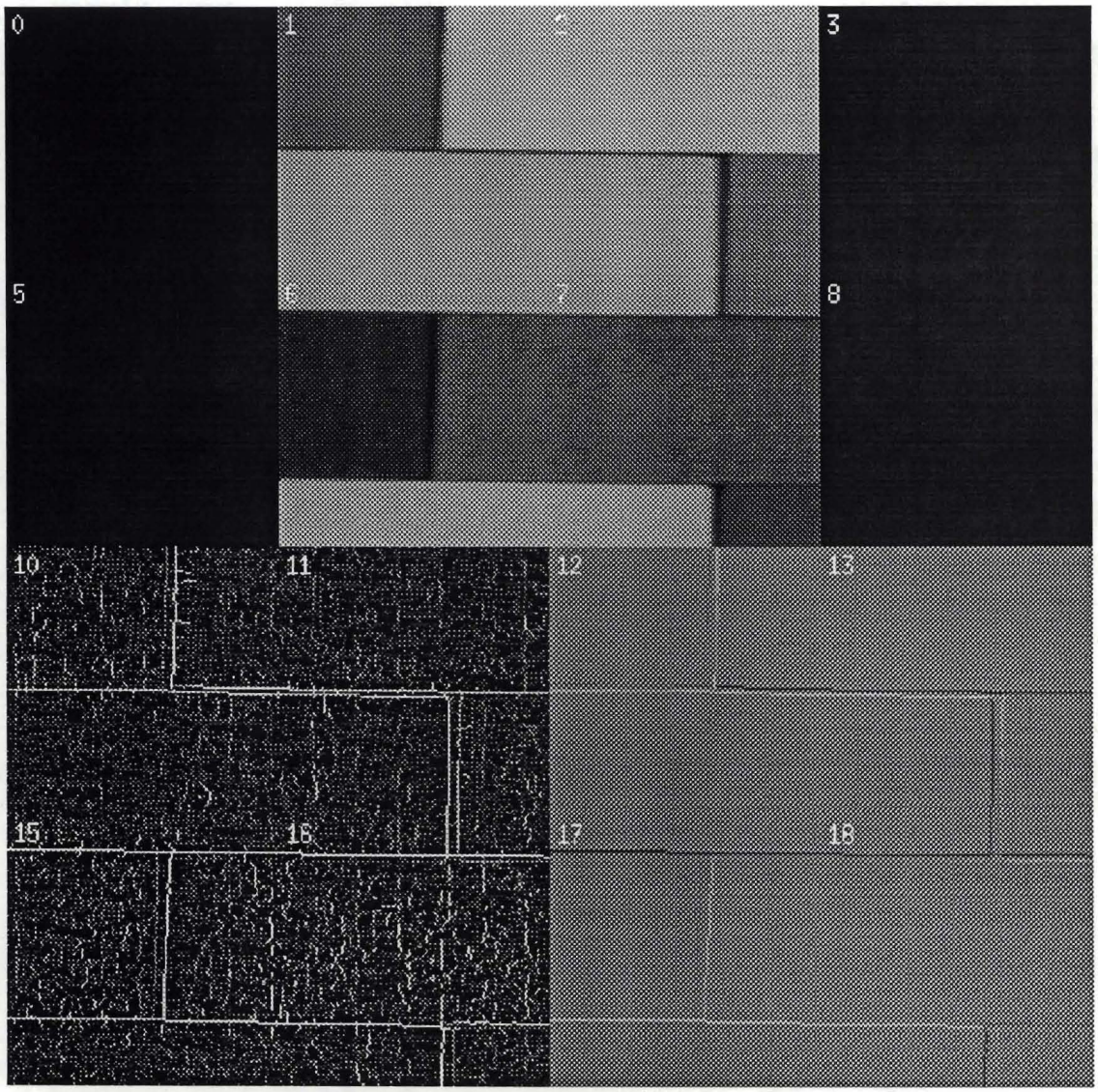

Figure 27:

Top: brightness image of the wall.
Bottom left: symbolic image, SCALE information displayed.
Bottom right: symbolic image, CONTRAST information displayed.

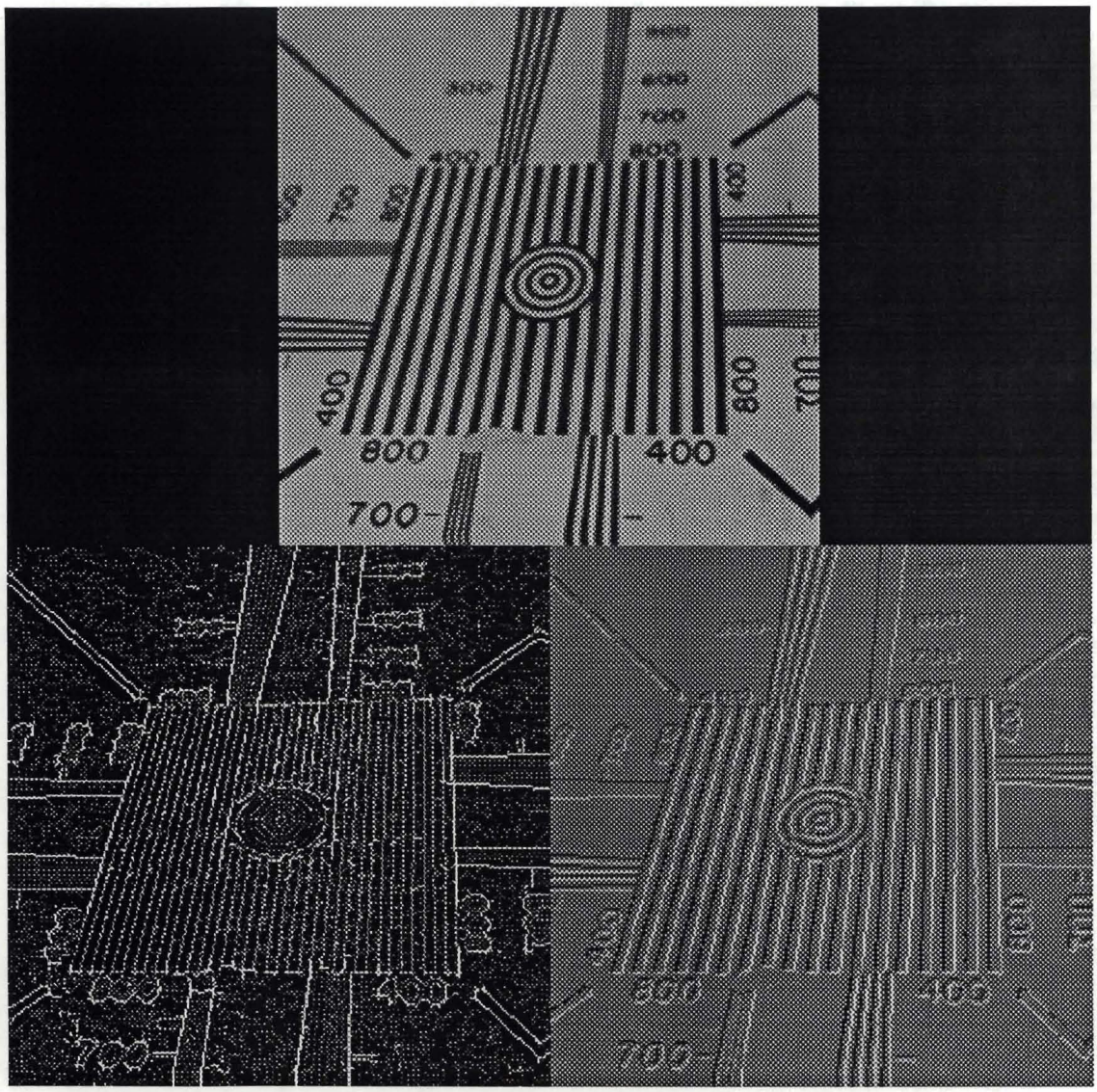

Figure 28:

Top: brightness image of patterns.
Bottom left: symbolic image, SCALE information displayed.
Bottom right: symbolic image, CONTRAST information displayed.

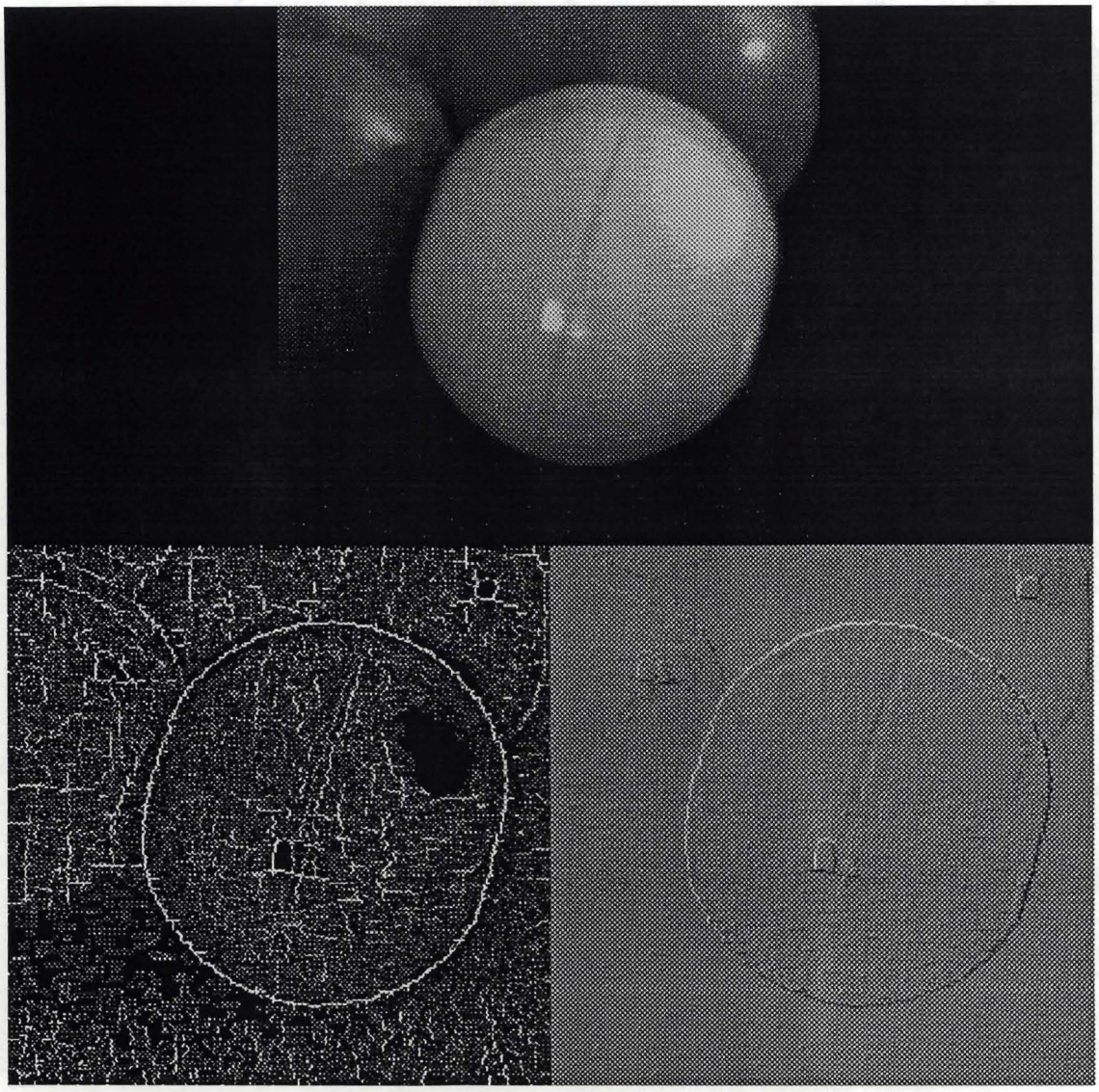

Figure 29:

Top: brightness image of balls.
Bottom left: symbolic image, SCALE information displayed.
Bottom right: symbolic image, CONTRAST information displayed.

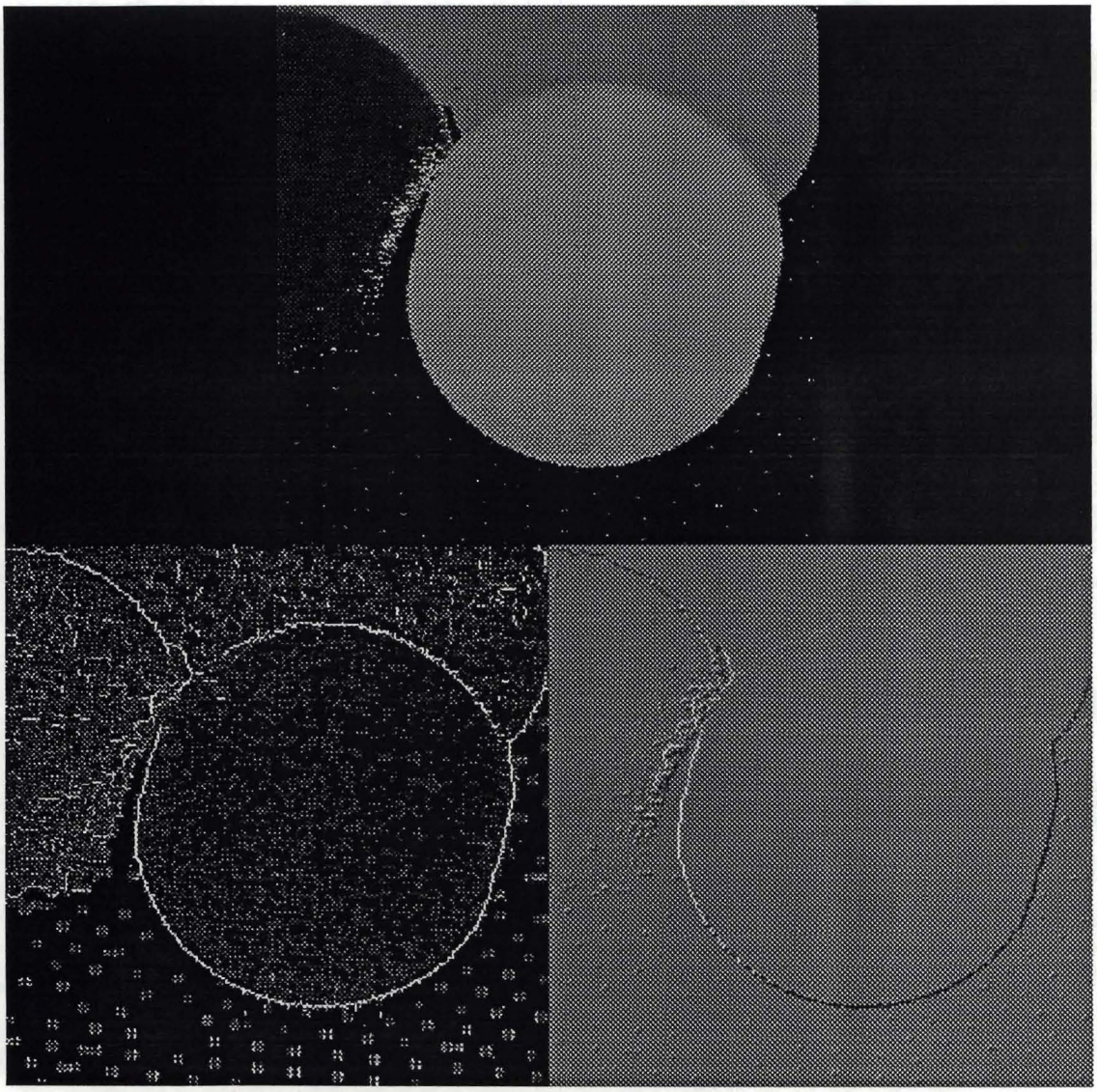

Figure 30:

Top: hue image of balls.
Bottom left: symbolic image, SCALE information displayed.
Bottom right: symbolic image, CONTRAST information displayed.

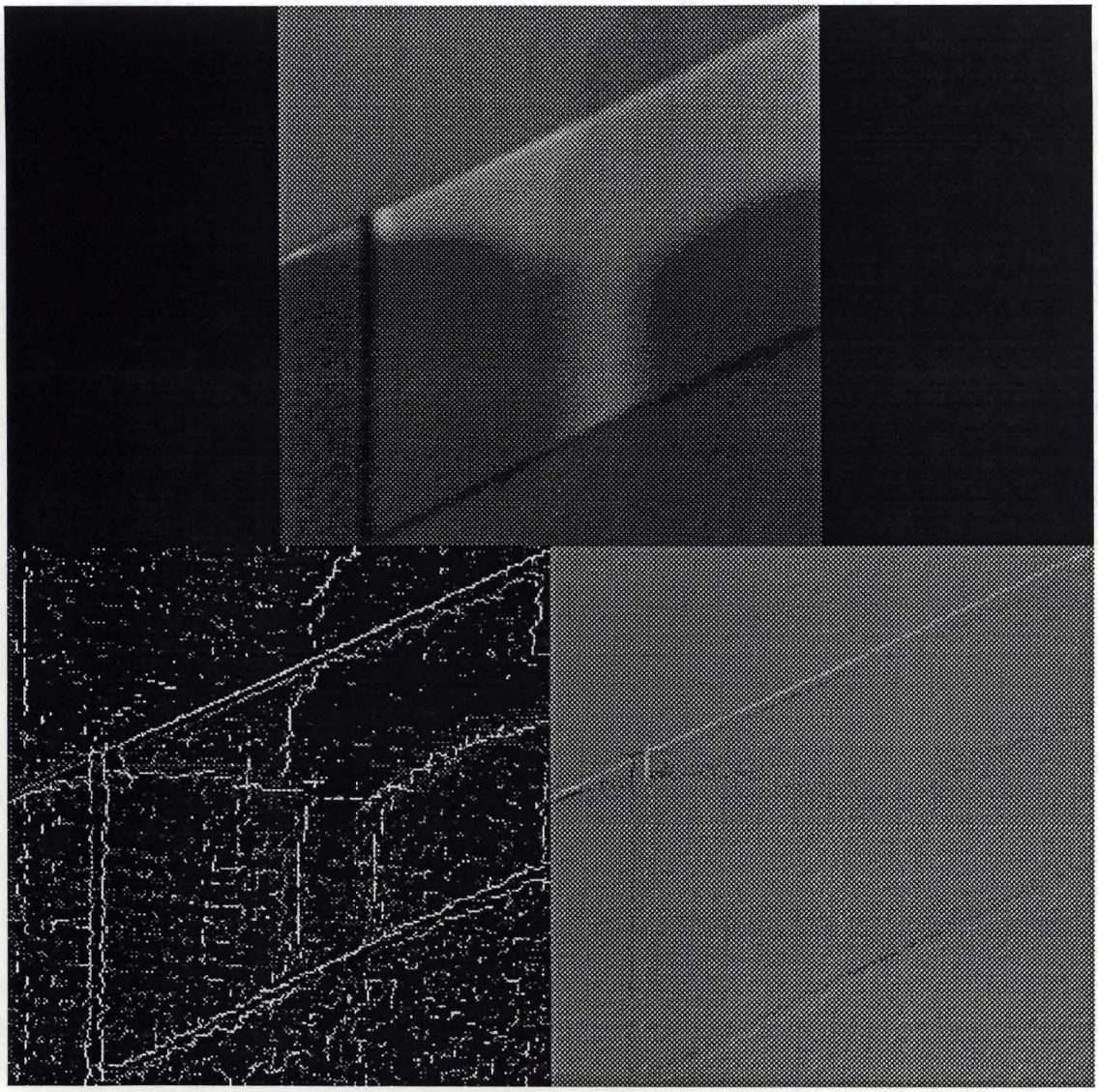

Figure 31:

Top: brightness image of shadows on a wall.
Bottom left: symbolic image, SCALE information displayed.
Bottom right: symbolic image, CONTRAST information displayed.

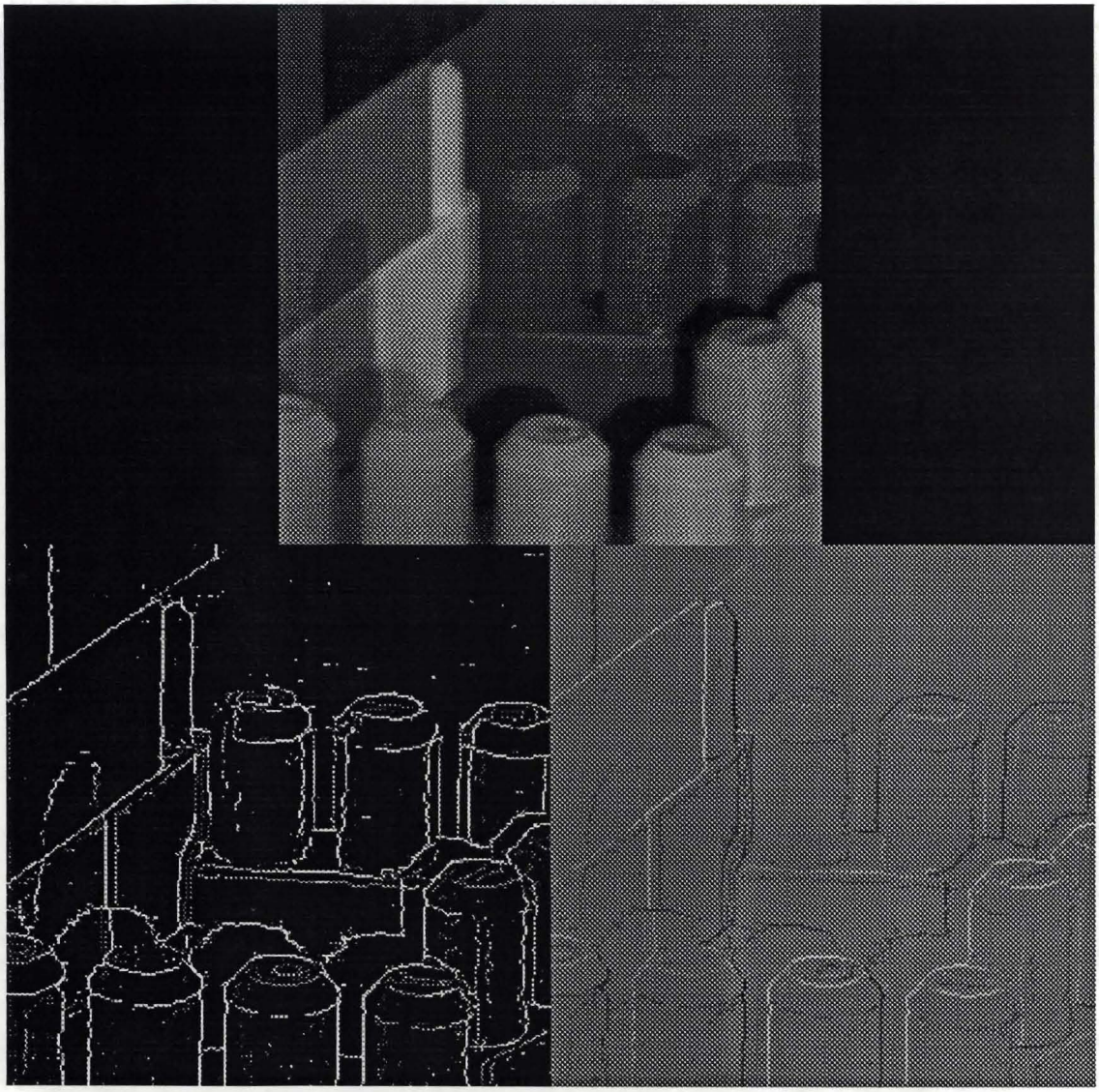

Figure 32:

Top: brightness image of a busy scene.
Bottom left: symbolic image, SCALE information displayed.
Bottom right: symbolic image, CONTRAST information displayed.

6.2 Comments and Conclusion

The amount of information that the symbolic image provides is definitely richer than the output of any other kind of edge detector. From now on we can see where the important and coarse features are. In the other hand the edges that come from texture or small closed contours inside bigger ones are classified as details. The color information help us to attribute physical properties to the contours. It does not prevent errors and ambiguities to exist. However most of them are explained by physical phenomena and not only by intrinsic properties of the signals.

This multiscale process uses a finite number of scales that are supposed to work with a very large amount of images (see the environment described in Fig 1). But as previously mentioned, the edge-merging can lead to wrong connections in scale space. Besides there is some arbitrariness when I decide not to give the same degree of importance to two edges that merge in scale space. In the other hand there is no way to tune the size of our filters to adjust the detection. The context is no a-priori knowledge of the scene. Consequently we must try to deal with this decomposition and with this tracking. It becomes necessary to work now on the interpretation of the symbolic image.

I tried to link the edge-pixels spatially by labelling them (see Fig 22). It means that the edge-pixels that belong to the same contour had the same label. But the criterion was only the degree of importance, in order to extract object contours according to the actual importance of the object. It turned out to be not good enough. Because of all the problems previously mentioned we cannot guarantee a perfect determination of the degree of importance we would like to attribute. Besides some edges get their importance from the only fact that they provide a very high local contrast. I am convinced that we must take the contrast range information into account and combine it with the degree of importance. Subsequently this combination must be used as the criterion to extract and label closed contours. Moreover it seems to subjectively corroborate how the human system responds to visual stimuli.

Aleš Leonardis and Gareth Funka-Lea are using this work to elaborate a more general perceptual learning process. They work on the extraction of texture and smooth surfaces from the initial image and from the wavelet decomposition. Aleš Leonardis found a nice way to modelize smooth surfaces with polynomial interpolation. His process is iterative and grows on the surface to be extracted. Therefore he needs a seed to start the process. These seed-pixels must be localized near the contours previously detected. I started to work on these seed-pixels in the symbolic image (see Fig 22). However this work is in its way and not finished.

The ultimate goal is to use the robot arm on which the camera is fixed to do real experiments. Once a shot is taken the whole system must recognize the ambiguous zones and decide to take new pictures from different angles in order to increase its perceptual understanding of the visual field. 3-D information and stereo vision are to be added too. The challenge is then to put all those tools together and overcome their tendency to provide errors.

References

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

[2] Stephane Mallat. *Multifrequency Channel Decompositions of Images and Wavelet Models.* IEEE Trans. on acoustics Speech and Signal Process. vol 37 No 12, December 1989.

[3] A. Grossmann and J. Morlet. *Decomposition of Hardy functions into square integrable wavelets of constant shape.* SIAM J. Math. vol 15 pp 723-736, 1984.

[4] P. J. Burt and E. H. Adelson. *The Laplacian Pyramid as a compact image code.* IEEE Trans. Commun. vol COM-31 pp 532-540, April 1983.

[5] A. Witkin. *Scale space filtering.* Proc. Int. Joint Conf. Artificial Intell., 1983.

[6] Nicolas Treil. *Image Wavelet Decomposition and Applications.* Technical Report MS-CIS-89-22 GRASP LAB 177, University of Pennsylvania, Department of Computer and Information Science, School of Engineering and Applied Science, 1989.

[7] R. Bajcsy, S.W. Lee, A. Leonardis.*Color Image Segmentation with Detection of Highlights and Inter-reflections.* GRASP Laboratory Technical Report MS-CIS-89-39, University of Pennsylvania, 1989.