Geodesic Active Regions for Texture Segmentation

Nikolaos Paragios, Rachid Deriche

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Geodesic Active Regions for Texture Segmentation

Nikos PARAGIOS and Rachid DERICHE

N° 3440

Juin 1998

THÈME 3

ISSN 0249-6399

Geodesic Active Regions for Texture Segmentation

Nikos PARAGIOS and Rachid DERICHE

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Robotvis

Rapport de recherche n° 3440 — Juin 1998 — 38 pages

Abstract: This paper proposes a framework for segmenting different textured areas over synthetic or real textured frames by curves propagation. We assume that the system has the ability to be taught over different texture prototypes. For each prototype a global statistical model is generated, as a set of probability density functions attributes from a multi-valued frame analysis, where different filter responses are used to create this multi-valued frame. Then, each prototype is represented by a reliable statistical model. Given an input frame composed of different texture types, the same bank of filters is applied. Over the generated multi-valued frame, we define an energy as a special form of a geodesic active contour model, a **Geodesic Active Region Model**, where we integrate boundary finding and region based segmentation approaches. This energy is minimized using a steepest gradient descend method, where *smoothing*, *edge-based*, and *region statistics* forces, move the curve toward the minimum of the designed objective function. Using the level set formulation scheme, complex curves can be detected, while topological changes for the evolving curves are naturally managed. In order to deal with the problem of noise influence, as well as to reduce the required computational cost, a multi-grid approach has also been considered. Finally, two different methods are used for the level set implementation, the *Narrow Band* and the *Hermes Algorithm*. Very promising experimental results are provided using synthetic and real textured frames.

Key-words: Texture Segmentation, Filters Bank, Statistical Modeling, Geodesic Active Contours, Geodesic Active Regions,Partial Differential Equations, Level Set

This work was funded in part under the VIRGO research network (EC Contract No ERBFMRX-CT96-0049) of the TMR Programme.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 — Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Régions Actives Géodésiques pour la Segmentation de Textures

Résumé : Dans ce rapport, nous présentons une méthode de segmentation d'images texturées, en faisant évoluer une courbe initiale qui converge, tout en pouvant changer de topologie, vers les frontières des différentes parties texturées présentes dans l'image. La méthode repose sur plusieurs parties dont la première consiste en une phase d'apprentissage préalable qui permet d'associer à chaque texture donnée un vecteur d'attributs issus d'une analyse statistique des densités de probabilité d'un ensemble de sous-images. Celles-ci proviennent de l'application à la texture concernée d'un banc de filtres bien adapté pour cette tâche de modélisation..Une énergie, qui intègre des informations sur la texture de la région et sur sa frontière, est ensuite proposée afin de formaliser la tâche de segmentation en une approche variationnelle. L'équation d'Euler-Lagrange, déduite de la minimisation de cette énergie, est alors utilisée afin de déformer une courbe initiale, considérée comme un contour actif géodésique qui va converger vers les différentes frontières des régions texturées présentes dans l'image, d'où le nom de *Régions Actives Géodésiques* associé à cette approche. La résolution de l'EDP par la méthode des courbes de niveau d'Osher et Sethian permet ensuite de mettre en oeuvre de manière efficace le processus d'évolution des contours tout en gérant automatiquement d'éventuels problèmes de changement de topologie durant la phase d'évolution.Une approche multi-résolution et les versions rapides, connues sous le nom de *NBA* et *Hermes* sont aussi utilisées pour mettre en oeuvre la méthode. Divers résultats expérimentaux sur des données synthétiques et réelles illustrent les remarquables capacités de cette nouvelle méthode.

Mots-clés : Segmentation de textures, Modélization statistique, Contours actifs Géodésiques, EDP, Courbes de niveau, Minimisation d'énergie.

Contents

1 Introduction

Image segmentation as well the edge/boundary detection are critical problems of early vision and they have been widely studied. Despite the progress which has been done in this area the proposed algorithms suffer in robustness and generality over large frame datasets. In this paper, we are interesting for a special application of image segmentation, the segmentation of texture frames. The ability of understanding and characterizing texture is an essential process and has great practical value in image processing applications, since different objects can be easily segmented, characterized and categorized based on texture information.

A common initial step in texture segmentation is texture analysis and modeling. In computer vision, the goal is to create a general statistic model capable of describing a wide variety of textures prototypes in a common framework, usually consistent with the understanding human texture perception. In this area, two main approaches have been considered:

- Filtering theory, suggests the decomposition of retinal image into a set of different sub-bands, which are the responses of the convolutions of the input image with a bank of linear and non-linear filters. This bank is usually composed of Gabor Filters [14] and Wavelets pyramids [23, 35]. These methods seems to have impressive performance in texture segmentation [5, 13, 16].

- Statistical modeling, supposes that the texture prototypes are probability distributions of random fields [11, 25]. The problem of texture analysis is formulated as a well-defined statistical problem, and a small number of parameters are involved in the representation. The main drawback of these models, is that they have limited forms, hence suffer from the lack of expressive power.

Similarly, concerning the segmentation process, we find two different categories of approaches. One is region based, which relies on the homogeneity of spatially localized features, whereas the other one is based on the methods of boundary finding relying on the gradient features at a subset of the spatial positions of an image (near an object boundary).

- The region-based approaches can be decomposed in two sub-categories. The region growing and merging techniques [32, 33] and the global optimization approaches, based on minimizing energy functions using Bayesian criteria [18, 24]. The main advantage of region growing methods, is that they generate, adapt and test the statistics inside the region, however they generate small holes and irregular boundaries. On the other hand using energy-based global formulation, we are less affected from the presence of noise, but is usually very difficult to minimize them, and the computational cost is also a quite significant drawback.

- Similarly, the boundary finding is divided in two different categories. One is local filtering techniques [19], such as edge detectors [6, 12], whereas the other one are Snakes and Balloons [8, 10, 17] methods. Filtering approaches use only local information and cannot ensure continuous edge-detection. Snake/Balloon models are based on information along the boundaries, and usually require a good initialization [17] to yield correct convergence. Opposite to these Snake/Balloon models, *geodesic active contour* models have been recently proposed, [8, 20, 22] whose initialization step is not crucial. The use of boundary finding techniques provides some important advantages. Shape variations can be easily handled, the model is less sensitive in changes in the grey scale distributions over the image since it relies on changes in the grey level, rather than their actual values. Finally by the use of these techniques, edges are better localized. In spite of the advantages offered by boundary finding techniques, their use for texture segmentation seems to be problematic, since they are based on edges-features that are completely unreliable for texture frames.

- Finally, there is some recent work seeking to integrate region growing and edge detection [9, 21, 31, 36, 37, 38]. The difficulty lies in the fact that even though the two methods yield complementary information, they involve conflicting and incommensurate objectives, as region based segmentation attempts to capitalize on homogeneity properties whereas boundary finding techniques use the non-homogeneity of the same data as a guide.
  In [37], a statistical framework for image and texture segmentation is proposed, which combines the geometrical features of snake models and the statistical methods for region growing. Although in this approach snake models have been involved, the "boundary detection" information is not utilized. Additionally the algorithm has two separate steps, which are not coupled, the segmentation step, and the region growing step. Finally, the number of initial regions as well as their initialization, seems to be a very crucial step, strongly related with the efficiency of the proposed approach.

Our overall goal is to develop a framework which **combines the existing approaches** in the domain **of texture analysis** (*Filtering and Statistical modeling*), as well as in the domain **of texture segmentation** (*Region-Based and Boundary Finding*). The first step of our approach consists of texture analysis and modeling. This is achieved by fusing filtering theory and statistical analysis. This step is performed off-line. Our goal is to generate a global statistical model for each texture prototype. As a first step, we select from a general filter bank a set of linear and non-linear filters, to capture the features of the texture. Then the marginal distribution of each filter response is used for specifying the statistic character of the prototype. This distribution is approximated as a mixture synthesis of Gaussian distributions. The set of these distributions defines a global multi-vector statistical model for each texture prototype.

The second step, consists of creating a global segmentation framework, where region based and boundary finding techniques are cooperating in a coupled common model. This would lead to a system where the two modules would operate simultaneously. The combination of these two modules presents a set of quite important advantages. We try to integrate boundary finding and region based segmentation rather than edge detection and region growing, by defining a new **Geodesic Active Region Model** where *smoothing*, *"edge-based"*, and *statistics region* forces move the region boundary. The main difference of our model compared with the classic geodesic active contour model [8, 20] is that the interface evolves using information not only among it, but also information which come from the regions inside and outside of it. Thus, the contour is propagating by means of velocity that contains three terms, one which is related to the regularity of the curve, a second which shrinks or expands it towards the boundary, and a third which supports the region "homogeneity" of the internal and the external region, defined by the boundary. This model is motivated as a combination of a curve evolution approach and an energy minimization one. The changes of topology can be easily obtained using a level-set approach [27], thereby several texture regions can be detected simultaneously. In order to deal with noise influence and to achieve a faster algorithm, the front propagation methods are combined with a multi-grid approach. Finally, for the front propagation problem, two different schemes are used, the *Narrow Band* [2] and the *Hermes* [28].

The main properties of our approach are:

- A global statistical model is proposed for texture description which integrates the filtering theory with the statistical analysis.
- A coupled energy model is proposed (objective function) which integrates, the boundaries detection with the region segmentation.
- A Geodesic Active Region model is proposed, which connects the minimization of the objective function with the surfaces propagation.
- The model is parameter-free, changes of topology can be easily treated and the initialization step is not important.

The remainder of this paper is organized as follows. Section 2 describes the statistical model for texture representation, based on filtering methods and statistical analysis. This section is divided in two parts: First classical filtering methods for texture segmentation are revised. Then, the global statistical model for texture description is proposed. In Section 3, we present the main result of the paper the **Geodesic Active Region** model: the connection between curves propagation, geodesic active contours and texture segmentation. This section is divided in four parts: First the objective function is defined as a special form of a Geodesic Active Contour. Second, the minimization of this function is demonstrated,

while third the multi-grid model is integrated. Finally, the problem is solved using the level-set formulation, briefly introduced in the the fourth part of this Section. The level-set front propagation algorithms are shortly introduced in Section 4, while experimental results with the proposed approach, followed by conclusions and discussion appear in Section 5.

2 Texture Description Model

2.1 Filtering Methods

In many different applications like texture segmentation, target detection, etc., the use of linear or non-linear filter operators has been applied for feature extraction with quite satisfactory results. The main difficulty of this method is the selection of these filters, especially in the case of a very general application. Concerning the texture characterization problem, it is well known that the bank of filters which deal successfully with this problem is composed of intensity filters, isotropic and anisotropic filters, Gabor filters and their spectrum analyzers.

In this bank of filters, the Gaussian function serves a quite important role, since it acts like a low-pass frequency analyzer. To fix the notation we define a symmetric center-surround two dimensional Gaussian function as

$$g(x,y|\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

Using this notation the bank of used filters is:

- The intensity filter, $\delta(x,y) = I(x,y)$: [**fig.** 1(a)]
- An isotropic center-surround filter, which in our case is the Laplacian of Gaussian filter (LOG filter),

  $$F(x,y|\sigma) = S \cdot \left(1 - \frac{r^2}{2\sigma^2}\right) e^{-\frac{r^2}{2\sigma^2}} \ , \quad r^2 = x^2 + y^2 \tag{2}$$

  where S is a scale factor. Concerning the variance of these filters, we have $\sigma \in \{1,2,3,4,5\}$. A response of an isotropic filter of this form is shown in [**fig.** 1(b)]. Additionally, this category contains the (x,y) anisotropic directional derivatives filters.
- Gabor bandpass filters,

  $$G(x,y|\sigma,\phi,\psi) = g(x,y|\sigma) e^{-j2\pi(\phi x + \psi y)} \tag{3}$$

  with $\phi,\psi \in \left\{0, \frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}, \pi, 2\pi\right\}$.

(a) (b) (c) (d) (e)

Figure 1: **(a)** Intensity Filter Response, **(b)** The Laplacian of Gaussian Isotropic Filter Response, **(c)** Spectrum Analyzer of Gabor $\left(\frac{\pi}{6}, 0\right)$ filter, **(d)** Spectrum Analyzer of Gabor $\left(\frac{\pi}{3}, 0\right)$ filter, **(d)** Spectrum Analyzer of Gabor $\left(\frac{\pi}{3}, \frac{\pi}{6}\right)$ filter

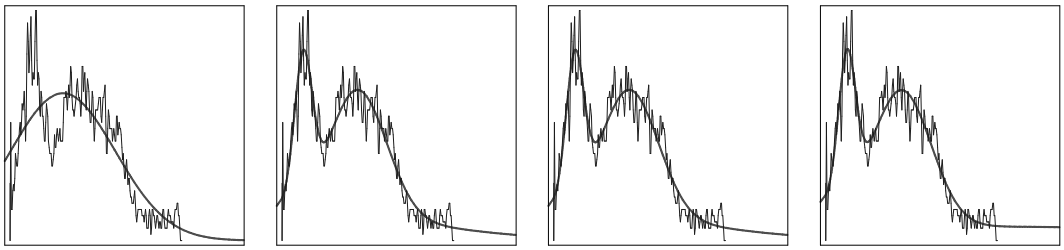
- The spectrum analyzer, whose response is the power of the Gabor pairs [**fig.** 1(c, d, e)]
$$S(x, y) = ||(G * I)(x, y)||^2 \tag{4}$$
smoothed by a Gaussian function.

2.2 Statistical Analysis

Let $T = \{t_1, t_2, .., t_{\mathtt{T_N}}\}$ be the set of input texture prototypes, and let $F = \{f_1, f_2.., f_{\mathtt{F_N}}\}$ be the bank of the selected filters, where $\mathtt{T_N}$ (*resp.* $\mathtt{F_N}$) is the number of texture prototypes (*resp.* the number of selected filters).

The first step of our approach is to teach the system over the different prototypes, which is equivalent of creating a global statistical model for each texture prototype, based on the available data. The set of available data concerning the texture prototypes, is composed of their input frames I_i, $\{i \in [1..., \mathtt{T_N}]\}$ and the responses I_{ij}, $\{j \in [1, ..., \mathtt{F_N}]\}$ of the selected filters over these frames. We assume that each filter response can be modelized statistically, using low-level statistics. In other words, we propose to generate a probability density function $p_{ij}(x|\Theta_{ij})$ of the filter response I_{ij} (where i corresponds to the texture prototype t_i and j corresponds to the filter f_j). We suppose that this probability density function is homogeneous, *i.e* independent of the pixel location, and it can be decomposed into many different components, where each component is under Gaussian law, that is: $p(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

Let P_{ij}^k be the *a priory* probability of the component k. The observed filter response values I_{ij}, are assumed to be obtained by selecting a component k with probability P_{ij}^k, and then selecting a value x according to the probability law $p_{ij}^k(x|\mu_{ij}^k, \sigma_{ij}^k)$. Thus the probability



(a) (b) (c) (d)

Figure 2: Mixture Analysis for [**fig.** 1.c] *(Spectrum Analyzer of* $(\frac{\pi}{6}, 0)$ *Gabor filter)*
Solid Line: **Samples**, Dashed Line: **Probability Density Functions**.
(a) One Component, *Mean Approximation Error: 7.3861e-05, Maximal: 0.0039875*
(b) Two Components, *Mean Approximation Error: 4.3575e-05, Maximal: 0.0034018*
(c) Three Components, *Mean Approximation Error: 4.184e-05, Maximal: 0.0028997*
(d) Four Components, *Mean Approximation Error: 4.1353e-05, Maximal: 0.003001*

density function is given by

$$p_{ij}(x|\Theta_{ij}) = \sum_{k=1}^{C_N} P_{ij}^k p_{ij}^k(x|\mu_{ij}^k, \sigma_{ij}^k) \tag{5}$$

where C_N is the number of mixture components, and Θ_{ij} is the vector of the unknown mixture synthesis parameters: $\Theta_{ij} = \{(P_{ij}^k, \mu_{ij}^k, \sigma_{ij}^k) : k \in [1, ..., C_N]\}$. Under this hypothesis there are two key problems: the number of different components C_N, and the estimation of the unknown parameters Θ_{ij} of these components. Concerning the number of components, experimentally in most of the cases it has been found to be equal to two, but there are some cases where at least three components must be assumed. This case appears very often for textures prototypes which are not quite homogeneous. The determination of the components number is based on the mean approximation error between the given samples and the mixture approximation, in other words we increment the number of components until the mean approximation error drops down from a given threshold. Concerning the example of [**fig.** (2)], the improvement of the approximation between the use of two and three components is not significant, thus we approximate this filter response with two components. The estimation of the unknown parameters Θ_{ij} is obtained using a well known non-linear minimization method, the Levenberg-Marquardt algorithm [26].

This operation is applied over the set of different filters responses. This permits us to create a vector of probability density functions $\mathbf{p}_i = (p_{i1}, p_{i2}, ..., p_{iF_N})$, for the texture prototype t_i. The same operation is applied for every texture prototype, and as an output we determine a multi-valued statistical representation of each prototype (see [**fig.** (3)].

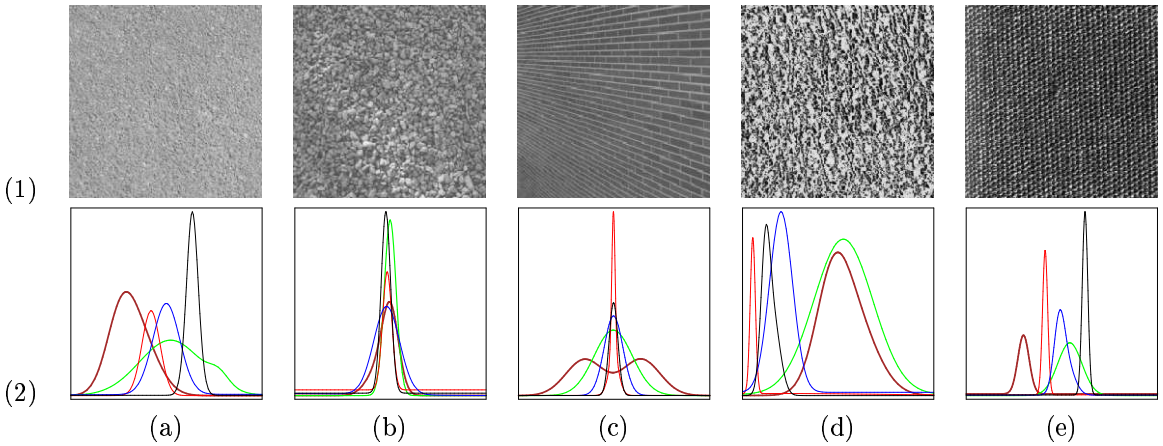


Figure 3: **(1)** Texture Prototypes, **(2)** Statistical Analysis: **(2,a)** Analysis of the Intensity Filter Response, **(2,b)** Analysis of the Laplacian Filter Response, **(2,c)** Analysis for the Real Part of Gabor Filter $\left(\frac{\pi}{3}, \frac{\pi}{2}\right)$ Response, **(2,d)** Analysis for the Spectrum Analyzer of Gabor Filter $\left(\frac{\pi}{3}, \frac{\pi}{2}\right)$, **(2,e)** Analysis for the Spectrum Analyzer of Gabor Filter $(2\pi, 2\pi)$

3 Geodesic Active Regions

The problem of texture segmentation consists of creating a partition over a given frame region R, into different subregions with *homogeneous properties* and *characteristics*, hopefully corresponding to different texture types. We suppose that there is a "dominant" texture type (t_{R_0}), and let R_0 be the subregion corresponding to this texture type. The interpretation of the "dominant" texture type differs from the classical ones, since in our case it corresponds to the background texture type, in other words the texture type which covers the borders of input frame $\mathbf{I}$. We suppose that a set of different texture types has been randomly placed onto the "dominant" texture type. The goal of our work is to create a partition of R into $\mathtt{R_N}$ piecewise *homogeneous* regions $\{R_i\}, i = 1, 2, .., \mathtt{R_N}$, **i.e** $R = \cup_{i=0}^{\mathtt{R_N}} R_i$, $R_i \cap R_j = \emptyset$, if $i \neq j$, correspond to different texture types. Let ∂R_i be the boundary of region R_i corresponds to texture type t_{R_i}, and let $\partial R = \partial R_0 = \cup_{i=1}^{\mathtt{R_N}} \partial R_i$ be the segmentation boundaries of the entire frame (the orientations of the boundaries are not examined).

3.1 Setting the Energy

The ideas of curves propagation have been generously studied and successfully applied over a reach variety of problems in computer vision. Based on work developed in [8, 20, 22, 29, 28, 30], we will try to reformulate the problem of texture segmentation within the framework

of curve evolution theory, by proposing a new model called **Geodesic Active Regions**. Let $E(\partial R)$ be the objective function to be minimized. The classic geodesic active contour model consists of minimizing

$$E(\partial R) = \sum_{i=1}^{R_N} \beta \underbrace{\int_0^1 |\partial \dot{\mathrm{R}}_i(p_i)|^2 dp_i}_{term_1} + \gamma \underbrace{\int_0^1 g^2(|\nabla \mathbf{I}(\partial R_i(p_i))|) dp_i}_{term_2} \tag{6}$$

where $\partial R_i(p_i) : [0,1] \rightarrow \mathbf{R^2}$ is a parameterization of the region boundary R_i in a planar form, and $\{\beta, \gamma\}$ are real positive constants. The first energy component accounts for the expected spatial properties (*i.e. smoothness*) of the contour while the second energy component stands for the *attraction* energy term of the curve towards the objects contour. Finally, g is a monotonically decreasing function such that $g(r) \rightsquigarrow 0$ as $r \rightsquigarrow \infty$ and $g(0) = 1$.

Following the work of [3, 7], it can be proved that the minimization of (6) leads to a geodesic curve with a new metric,

$$E(\partial R) = \zeta \sum_{i=1}^{R_N} \int_0^1 g(|\nabla \mathbf{I}(\partial R_i(p_i))|) |\partial \dot{\mathrm{R}}_i(p_i)| dp_i \tag{7}$$

where ζ is a positive constant.

It is well known that in the case of texture segmentation, the edges don't always correspond to region boundaries, thus if one wants to take into account the texture information, the function $g(.)$ must be replaced. Let $h : \mathbf{R^2} \rightarrow \mathbf{R}$ be an alternative selection of function $g(.)$, with values close to *zero* over the regions boundaries, and close to *one* otherwise.

3.1.1 Setting the Energy Using "texture-boundary" Measurements

The definition of function $h(.)$ for the case of texture frames is a quite interesting step. The role of function $h(.)$ is to incorporate the available texture edge-boundary information. It is quite obvious that in the case of texture images the gradient features don't rely always to real boundary edges. In order to obtain these features we try to design a local operator which neither selects the "best" observed frame, neither uses the entire set of the observed frames, and gives a measure of the likelihood of a pixel lying on the boundary between the dominant texture region and another texture region. We propose two different methods, one which makes use of a single frame and a second method which makes use of several differently parametric images or images from different modalities.

Let s be a pixel in the image grid. At each pixel s, a small neighborhood is defined [**fig.** 4]. Now, given a possible boundary point s and a direction θ, the neighborhood is divided into parts N_{s_l} and N_{s_r}. Concerning the different orientations θ, four different possible directions are assumed, $\theta = \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$.

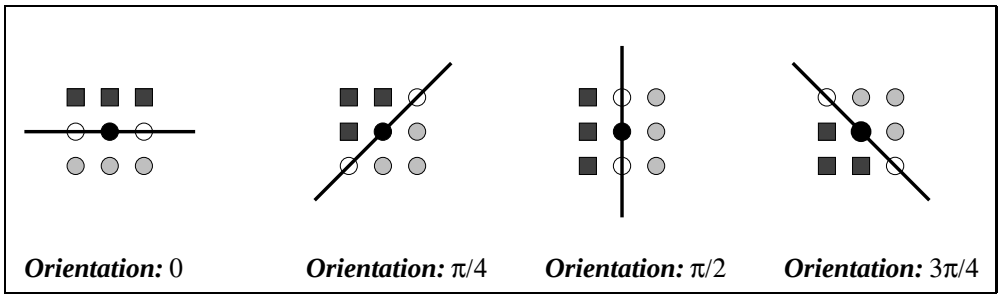


Figure 4: Local Operator to derive Image Information

- **Obtaining texture edge-boundary Measurement using a Single Frame**

We define the normalized probability that s lies on the boundary between the dominant texture region R_0 and the texture region R_r as:

$$p(s,\theta) = \frac{p(N_{s_L} \in R_0) \cdot p(N_{s_R} \in R_r) + p(N_{s_L} \in R_r) \cdot p(N_{s_R} \in R_0)}{p(N_{s_L} \in R_0) \cdot p(N_{s_R} \in R_0) + p(N_{s_L} \in R_r) \cdot p(N_{s_R} \in R_r)} \tag{8}$$

The problem which arises, is how we are going to define the probability of a given set of pixels belonging to a region, and what type of data we are going to use to define this probability. Although different parameterized frames are available, we are going to use, here, a single frame to estimate these probability values. We select the frame (filter response) which "best" discriminates the dominant texture prototype from the other prototypes. Thus the probability density functions corresponds to the statistical analysis of this filter responses over the different texture prototypes. This distribution is selected by maximizing:

$$p^*_{t_{R_0}\mathtt{j}} = \underset{\mathtt{j} \in [1,\mathtt{F_N}]}{\mathbf{argmax}} \left\{ \frac{1}{\sigma(p_{t_{R_0}\mathtt{j}})} \sum_{i=1}^{\mathtt{T_N}} \sigma(p_{i\mathtt{j}}) \left[\mu(p_{t_{R_0}\mathtt{j}}) - \mu(p_{i\mathtt{j}})\right]^2 \right\} \tag{9}$$

The interpretation of the above objective function is obvious. We are looking for the filter response which best segments the dominant texture, that provides small variance ($\sigma(p_{t_{R_0}j})$ for probability density function corresponding to the dominant texture case. Complementary, this filter response best discriminates the dominant texture prototype from the others, that the sum of distances between the mean value corresponding to t_{R_0} and the means values corresponding to the other texture types is maximized. This step can be easily performed off-line.

- **Obtaining texture edge-boundary Measurement using Multiple Frames**

Let $\mathbf{x}_i(s_l)$ (*resp.* $\mathbf{x}_i(s_r)$) be the mean neighborhood value of N_{s_l} (*resp.* N_{s_r}) for the observed frame $\mathbf{I}_i$: the response of filter f_i. Since several differently parametric frames are available, we can estimate the vector $\mathbf{x}(s_l) = [x_1(s_l), ..., x_{\mathtt{F_N}}(s_l)]$ (*resp.* $\mathbf{x}(s_r)$). Based on the texture statistical model (explained in the previous Section), we can estimate for a given neighborhood region (with observation data set $\mathbf{x}$) the probability vector corresponding to each texture prototype, thus we generate the matrix

$$P(\mathbf{x}) = \begin{bmatrix} \mathbf{p}_1(\mathbf{x}) \\ \vdots \\ \mathbf{p}_{\mathtt{T_N}}(\mathbf{x}) \end{bmatrix} = \underbrace{\begin{bmatrix} p_{11}(x_1) & \dots & p_{1\mathtt{F_N}}(x_{\mathtt{F_N}}) \\ \vdots & \vdots & \vdots \\ p_{\mathtt{T_N}1}(x_1) & \dots & p_{\mathtt{T_N}\mathtt{F_N}}(x_{\mathtt{F_N}}) \end{bmatrix}}_{\mathtt{F_N}\times\mathtt{T_N}} \tag{10}$$

Now given a point s, a partition over the neighborhood (N_{s_l}, N_{s_r}), and its corresponding data vectors $(\mathbf{x}(s_l), \mathbf{x}(s_r))$, we define the probability that s lies on the boundary between the dominant texture region R_0 and the texture region R_r using a correlation criterion, as the dissimilarity between the two corresponding probability vectors :

$$p(s,\theta) = \frac{||\mathbf{p}_{t_{R_0}}(\mathbf{x}(s_l)) - \mathbf{p}_{t_{R_r}}(\mathbf{x}(s_r))||^2}{||\mathbf{p}_{t_{R_0}}(\mathbf{x}(s_l))||^2 + ||\mathbf{p}_{t_{R_r}}(\mathbf{x}(s_r))||^2} + \frac{||\mathbf{p}_{t_{R_r}}(\mathbf{x}(s_l)) - \mathbf{p}_{t_{R_0}}(\mathbf{x}(s_r))||^2}{||\mathbf{p}_{t_{R_0}}(\mathbf{x}(s_l))||^2 + ||\mathbf{p}_{t_{R_r}}(\mathbf{x}(s_r))||^2} \tag{11}$$

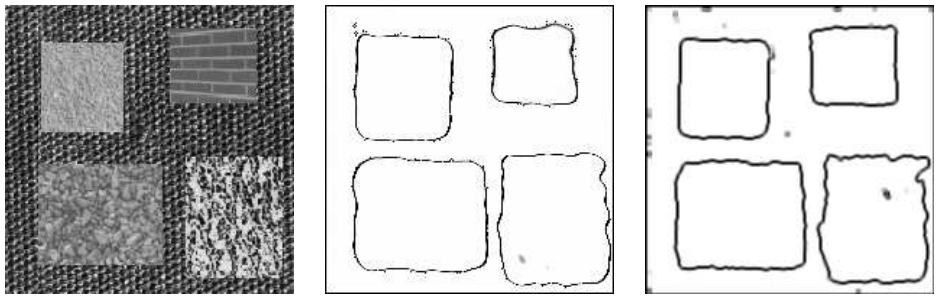
where

$$\frac{||\mathbf{p}_m(\mathbf{x}) - \mathbf{p}_n(\mathbf{y})||^2}{||\mathbf{p}_m(\mathbf{x})||^2 + ||\mathbf{p}_n(\mathbf{y})||^2} = \frac{\sum_{j=1}^{\mathtt{F_N}}[p_{mj}(x_j) - p_{nj}(x_j)]^2}{\sum_{j=1}^{\mathtt{F_N}} p_{mj}(x_j)^2 + \sum_{j=1}^{\mathtt{F_N}} p_{nj}(y_j)^2} \tag{12}$$

The correlation scores are computed by comparing the corresponding probability vectors between the texture prototypes t_{R_0} and t_{R_r}. Supposing that we are in the boundaries of a texture region, then the correlation between the probability vectors of t_{R_0} and t_{R_r} is very bad, thus the probability that s lies on the boundary goes to one.

Since the probability that s lies on the boundary between the dominant texture region R_0 and the texture region R_r is defined for both cases, the next problem is to define the texture prototype t_r^* corresponding to the region R_r, as well as the orientation θ^*. The best way to select t_r^* as well as the orientation θ^* is to generate the matrix

$$P_{\mathsf{EDGE}}(t,\theta) = \begin{bmatrix} p(0,t_1) & p(\frac{\pi}{4},t_1) & p(\frac{\pi}{2},t_1) & p(\frac{3\pi}{4},t_1) \\ \vdots & \vdots & \vdots & \vdots \\ p(0,t_{t_{R_0}-1}) & p(\frac{\pi}{4},t_{t_{R_0}-1}) & p(\frac{\pi}{2},t_{t_{R_0}-1}) & p(\frac{3\pi}{4},t_{t_{R_0}-1}) \\ p(0,t_{t_{R_0}+1}) & p(\frac{\pi}{4},t_{t_{R_0}+1}) & p(\frac{\pi}{2},t_{t_{R_0}+1}) & p(\frac{3\pi}{4},t_{t_{R_0}+1}) \\ \vdots & \vdots & \vdots & \vdots \\ p(0,t_{\mathtt{T_N}}) & p(\frac{\pi}{4},t_{\mathtt{T_N}}) & p(\frac{\pi}{2},t_{\mathtt{T_N}}) & p(\frac{3\pi}{4},t_{\mathtt{T_N}}) \end{bmatrix} \tag{13}$$



(a) (b) (c)

Figure 5: (a) Input frame (b) Texture edge-boundary features extraction using a single frame, (c) Texture edge-boundary features extraction using multiple frames

and t_r^* , θ^* correspond to the biggest element in matrix P. Thus for each point $s = (x, y)$ the *"edge" features* [**fig.** 5] are captured by the function $h(.)$:

$$h(s) = e^{-p[\theta^*(s), t_r^*(s)]} \tag{14}$$

Motivated by work proposed in [7] we can rewrite the geodesic active contour equation as:

$$E(\partial R) = \zeta \sum_{i=1}^{\mathrm{R_M}} \int_0^1 h(\partial R_i(p_i)) |\partial \dot{\mathrm{R}}_i(p_i)| dp_i \tag{15}$$

where the texture edge-based features have been incorporated to the model.

3.1.2 Setting the Energy Using Region-based Measurements

The main goal of region-based segmentation methods is to classify a particular texture frame into a number of regions. Thus for each pixel in the image, we need somehow to decide or estimate which class it belongs to. There is a variety of approaches to do region based segmentation, but most of them are finally turn out to be minimization of objective-cost functions. Usually these objectives-costs functions consist of two different terms. One, which express the expected spatial properties (homogeneity of the segmentation map), and a second which express the adequacy between the segmentation map and the observed data. An important advantage in our case comes from the fact that we have an idea concerning the elements which have to be segmented, based on the statistical analysis over the different texture prototypes. This means that we can define the adequacy term between the observed data and the segmentation maps

Guided by the way of defining these objective functions for Markov Random Fields and avoiding the term which express the hypothesis of homogeneity, for our texture segmentation problem we define the objective function as:

$$E(R) = \alpha \sum_{i=0}^{\mathrm{R_N}} \int\int_{R_i} \sum_{j=1}^{\mathrm{F_N}} w_j \mathbf{log}\left[p_{t_{R_i}j}(\mathbf{I}_j(x,y))\right] dxdy \tag{16}$$

where $\mathbf{I}_j$ is the response of filter f_j over the input frame, α is a negative constant, and w_j are positive weight constants. The above equation has a simple interpretation. Supposing that, for a given point (x,y) the proper decision has been taken. This means that the probability values $p_{t_{R_i}j}(\mathbf{I}_j(x,y))$ for each filter response support this decision. These probability values have to be closed to *one*, thus the function $\mathbf{log}\left[p_{t_{R_i}j}(\mathbf{I}_j(x,y))\right]$ gives a very small negative value, which is multiplied by α, gives a small positive value. On the other hand if a wrong decision has been taken the corresponding probability values are closed to *zero*, and the function $\mathbf{log}\left[p_{t_{R_i}j}(\mathbf{I}_j(x,y))\right]$ gives very big negative values, which multiplied by α, charge the objective function excessively.

3.1.3 Geodesic Active Regions: The Energy Integration

Motivated by the excellent work proposed in [9, 36, 37], and following our previous work on tracking using geodesic active contours [29, 28, 30] and tracking using classical snakes and region information [4], we incorporate the two different segmentation models, by defining an objective function as an improved geodesic active contour model:

$$E(\partial R) = \underbrace{\zeta \sum_{i=1}^{\mathrm{R_N}} \int_0^1 h(\partial R_i(p_i))|\dot{\partial \mathrm{R}}_i(p_i)|dp_i}_{Boundary\ \ Finding\ \ Term} + \underbrace{\alpha \sum_{i=0}^{\mathrm{R_N}} \int\int_{R_i} \sum_{j=1}^{\mathrm{F_N}} w_j \mathbf{log}\left[p_{t_{R_i}j}(\mathbf{I}_j(x,y))\right] dxdy}_{Segmentation Term} \tag{17}$$

The above equation has a simple interpretation. The region segmentation is obtained by minimizing two kind of "energy terms". The first one (*Boundary Finding*) gives a minimal-length smoothed curve over the region boundaries, while the second one (*Segmentation*) minimizes the objective function inside this region, by supposing *homogeneity*. Concerning the boundary of the dominant texture (∂R_0), it doesn't appear in the objective function because it corresponds to the union of the other Regions Boundaries (the orientation is different). The optimization problem can be reformulated as

$$E(\partial R) = \alpha \int\int_{R_0} \sum_{j=1}^{\mathrm{F_N}} w_j \mathbf{log}(p_{t_{R_0}j}(\mathbf{I}_j(x,y))dxdy$$
$$+ \sum_{i=1}^{\mathrm{R_N}} \left\{ \zeta \int_0^1 h(\partial R_i(p_i))|\dot{\partial \mathrm{R}}_i(p_i)|dp_i + \alpha \int\int_{R_i} \left\{ \sum_{j=1}^{\mathrm{F_N}} w_j \mathbf{log}(p_{t_{R_i}j}(\mathbf{I}_j(x,y))dxdy \right\} \right\} \tag{18}$$

Although the objective function (18) is well-defined, there are still some unknown variables. We want to minimize this function over the region boundaries, and as an output we would like to obtain also the region segmentation. In order to do this, as it is quite clear from the definition of the objective function, we need a correspondence between the regions R_i and texture prototypes t_{R_i} something which we don't have a priory. This problem is going to be confronted later (**Subsection 3.2**).

3.2 Minimizing the Energy

Finally, the objective function is minimized using a steepest gradient descend method. Let $\vec{u} = (x, y)$ be a point of the initial curve. We compute the Euler-Lagrange equations [1] , and according to them, in order to deform each point $\vec{u}$ of the initial curve towards the local minima of the objective function [8],we should use the following equation:

$$\frac{d\vec{u}}{dt} = \alpha \left\{ \sum_{j=1}^{F_N} w_j \mathbf{log}(p_j(\mathbf{I}_j(\vec{u}))) \right\} \vec{\mathcal{N}}_{\partial R_0}(\vec{u}) +$$

[1] The following nice development can be found in [36, 37]. The problem of taking functional derivatives of integrals along contours and integrals over regions can be confronted using Green's theorem, which is a special case of Stokes theorem.Consider

$$E[\partial R] = \int \int_R f(x, y) dx dy$$

where $\partial R = (x_{(s)}, y_{(s)})$ is the boundary of the contour of region R, with $0 \leq s \leq l$ is the arc-length. Greens theorem states that: for a planar region R, $(P(x, y), Q(x, y))$ is any vector field with continuous first order derivatives, then

$$\int \int_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial x} \right) dx dy = \int_{\partial R} P dx + Q dy = \int_0^l (P\dot{\mathrm{x}} + Q\dot{\mathrm{y}}) ds$$

where $\dot{\mathrm{x}}$,$\dot{\mathrm{y}}$ denotes the differentiation with respect to s. Let $Q(x, y) = \frac{1}{2} \int_0^x f(t, x) dt$ and $P(x, y) = -\frac{1}{2} \int_0^y f(x, t) dt$. In that case $\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial x} = f(x, y)$, thus

$$E[\partial R] = \int \int_R f(x, y) dx dy = \int_0^l L(x, \dot{\mathrm{x}}, y, \dot{\mathrm{y}}) ds$$

where $L(x, \dot{\mathrm{x}}, y, \dot{\mathrm{y}}) = Q(x, y)\dot{\mathrm{x}} + P(x, y)\dot{\mathrm{y}}$. By Euler-Lagrange equation, we get the gradient of $E[\partial R]$ with respect to any point $(x_{(s)}, y_{(s)}) \in \partial R$ we have:

$$\frac{\delta E}{\delta x} = \frac{\partial E}{\partial x} - \frac{d}{ds} \frac{\partial L}{\partial \dot{\mathrm{x}}} = \ldots = f(x, y)\dot{\mathrm{y}} \quad , \quad \frac{\delta E}{\delta y} = \frac{\partial E}{\partial y} - \frac{d}{ds} \frac{\partial L}{\partial \dot{\mathrm{y}}} = \ldots = -f(x, y)\dot{\mathrm{x}}$$

and since $\vec{\mathcal{N}}_{(x,y)} = (\dot{\mathrm{y}}, -\dot{\mathrm{x}})$ is the normal along the contour, by setting $\vec{u} = (x, y)$ we have

$$\frac{\delta E}{\delta u} = f(x, y) \vec{\mathcal{N}}_{(x,y)}$$

$$\zeta\left(h(\vec{u})\mathcal{K}_{\partial R_r}(\vec{u})+\nabla h(\vec{u})\cdot\vec{\mathcal{N}}_{\partial R_r}(\vec{u})\right)\vec{\mathcal{N}}_{\partial R_r}(\vec{u})+\alpha\left\{\sum_{j=1}^{\mathrm{F_N}}w_j\mathbf{log}(p_j(\mathbf{I}_j)(\vec{u}))\right\}\vec{\mathcal{N}}_{\partial R_r}(\vec{u}) \quad (19)$$

where r is the index of the region in which the point $\vec{u}$ belongs, $\mathcal{K}_{\partial R_r}$ is the Euclidean curvature with respect to the curve ∂R_r and $\mathcal{N}_{\partial R_r}$ is the unit inward normal to ∂R_r. Since the curves ∂R_0 and ∂R_r have inverse inwards vectors at each common point $\vec{u}$, we have $\vec{\mathcal{N}}=\vec{\mathcal{N}}_{R_r}=-\vec{\mathcal{N}}_{R_0}$. Taking this into account and replacing $\mathcal{K}_{R_r}$ by $\mathcal{K}$, the motion equation for $\vec{u}$ can be rewritten as:

$$\frac{d\vec{u}}{dt}=\left(\underbrace{\zeta\left[h(\vec{u})\mathcal{K}(\vec{u})+\nabla h(\vec{u})\cdot\vec{\mathcal{N}}(\vec{u}))\right]}_{Boundary\ \ Finding\ \ Forces}+\underbrace{\alpha\left[\sum_{j=1}^{\mathrm{F_N}}w_j\mathbf{log}\frac{p_{t_{R_r}j}(\mathbf{I}_j(\vec{u}))}{p_{t_{R_0}j}(\mathbf{I}_j(\vec{u}))}\right]}_{Statistics\ \ Region\ \ Forces}\right)\vec{\mathcal{N}}(\vec{u})) \quad (20)$$

The obtained motion equation has a quite obvious interpretation. There are two kind of "forces" acting on the contour, both in the direction of the inward normal. The first term, is a contour force which contains information regarding the boundaries of the different region areas. This term takes into account the boundary image characteristics, and at the same time is a *smoothing* force, since its value depends on the curvature term. The second term is a statistic-region force. We will try to interpret this term. Supposing that $\vec{u}$ is a pixel of the dominant texture font. In such a case the input data $\mathbf{I}$ as well as the responses of the filter bank $\mathbf{I}_j$, must support the hypothesis R_0, thus $p_{t_{R_0}j}(\mathbf{I}_j(\vec{u}))>p_{t_ij}(\mathbf{I}_j(\vec{u})),\ \forall\ t_i\neq t_{R_0}$. The influence of the statistics region forces, now is quite evident. In the case where the observed multi-value data at $\vec{u}$ fits better to the dominant texture $\alpha\mathbf{log}\left(\frac{p_{t_{R_r}j}(\mathbf{I}_j(\vec{u}))}{p_{t_{R_0}j}(\mathbf{I}_j(\vec{u}))}\right)>0,\ \forall\ t_{R_i}$ we are going to compresses the region, otherwise we are going to expand the region. This is a very nice property, since the contour can evolve either inwards, either outwards at the same time for different pixel sites. This is possible due to the fact that the contour propagation is guided by a speed function which is either positive, either negative, and is estimated according to the observed data. Opposite to this, on the classic geodesic active contour model we don't meet this property, since we have negative speed values only because of the curvature effect. As a consequence, the contour initialization doesn't pose significant problems.

Finally, the problem of correspondence regions and texture prototypes in not solved yet. Giving a second look on the motion equation (20), we can see that the term of textures-regions correspondence appears only in the statistics force, which is estimated locally. Taking this into account, the textures-regions correspondence is obtained by maximizing the statistic force. In other words for each point of the given boundaries we compare the dominant texture hypothesis, with the texture hypothesis which fits better with the observed data on this point.

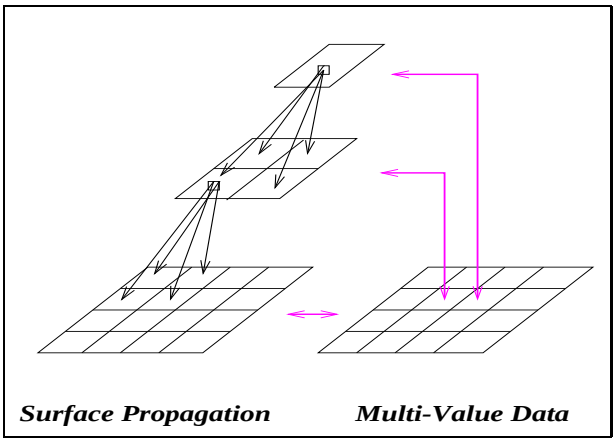


Figure 6: A Multi-Grid Approach for Geodesic Active Regions

3.3 Multi-Grid Geodesic Active Regions

However, analyzing the motion equation (20), we can see some hidden drawbacks. It is quite obvious that the statistic forces at each boundary point $\vec{u}$ will depend on the probability distributions of the different texture prototypes. This force seems to be plausible, since the vector of values in the ideal case fits better with the texture prototype corresponding to it, but this hypothesis may not be always valid. First, the texture analysis can give similar distributions for two different texture prototypes. Additionally, it is quite difficult to categorize a pixel based on its local data, since the *homogeneity* of the texture prototypes is usually defined over small block or regions. Second the presence of noise, could generate similar problems. This is not allowed in our case, since as an obvious result the model will converge to many different contours and regions corresponding to noise and not to real texture prototypes.

To cope with these problems, we can use a circular window approach, as proposed in [37]. In such a case, a circular window $W(\vec{u})$ of m pixels is defined around each point, which corresponds the the neighbor set of this point. The effect of this is to replace the probability density values $p_{t_{R_r}j}(\mathbf{I}_j(\vec{u}))$ by the joint probability $\prod_{v \in W(\vec{u})} p_{t_{R_r}j}(\mathbf{I}_j(\vec{v}))$, which could also be adopted for our case. As we are going to explain later, a quite significant drawback of the methods for contour propagation, is the excessive and sometimes the unbearable computational cost. Taking into account this fact, together with the problems analyzed in the previous paragraph, we adopt the idea of a "circular" window by proposing a multi-grid approach, which at the same time decreases the required computational cost.

The main idea is to solve the Geodesic Active Region problem in many different spaces, which are subsets of the original one. Each contour point corresponds to a set of pixels in the

original space. A quite sophisticated approach consists in defining a consistent *multi-grid* contour propagation model by using contours which are constrained to be piecewise constant over smaller and smaller pixel subsets [15]. The objective function which is considered at each level is then automatically derived from the original finest scale energy function, as well as the partial differential equation which deforms the initial contour. Also, full observation space is used at each contour level and there is no necessity for constructing a multi-resolution pyramid of the data [**fig.** 6]. Correspondingly for the level L, we obtain a generalized objective function:

$$E(\partial R^L) = \zeta \sum_{i=1}^{\mathrm{R_N}} \int_0^1 h_L(\partial R_i^L(p_i)) |\dot{\partial \mathrm{R}}_i^L(p_i)| dp_i +$$

$$\alpha \sum_{i=0}^{\mathrm{R_N}} \int\int_{R_i^L} \left\{ \sum_{j=1}^{\mathrm{F_N}} w_j \frac{1}{|W^L(x,y)|} \int\int_{(u,v)\in W^L(x,y)} \mathbf{log}\left[p_{t_{\partial R_0^L} j}(\mathbf{I}_j(u,v)) \right] dudv \right\} dxdy \quad (21)$$

where ∂R_i^L is the boundary of region R_i^L (corresponds to level L), $W^L(x,y)$ the window in full data space corresponding to the point (x,y) of data space at level L, and

$$h_L(\partial R_i^L(p_i)) = \frac{1}{\left|W^L(\partial R_i^L(p_i))\right|} \sqrt{\int\int_{(x,y)\in W^L(\partial R_i^L(p_i))} h^2(u,v) dudv}$$

The multi-resolution approach solves our problems. At low resolution levels a block of data points is used to move the boundary points, thus the chance to be representative in terms of the texture prototype are quite bigger. On the other hand, at these low resolution levels the size of the window is quite big and the boundary is not located precisely. This is not a problem since moving from the low resolution to the high resolution levels, the window size gets smaller and smaller, and the statistics forces moving the boundary points are more accurate. Additionally at low resolution level, we have obtained a segmentation where the noise influence has been removed, and since this result is used to initialize the operation at the next level we don't have the problems mentioned above.

3.4 Level Set Implementation

The motion equation (20) could be implemented using a Lagrangian approach, where we produce equations of motion for the position vector $\partial R(p,t)$, and then updating these position using difference approximation scheme. However, there are several problems with this approach. The main problem is that the evolving model is not capable to deal with topological changes of the moving front. Additionally, this method cannot be easily extended to three dimensions.

This could be avoided by introducing the work of Osher and Sethian [27]. The central idea is to represent the moving front $\partial R(t)$ as the zero-level set $\{\Phi = 0\}$ of a function Φ. This representation of $\partial R(t)$ is implicit, parameter-free and intrinsic. Additionally, it is topology-free since different topologies of the zero level-set do not imply different topologies of Φ. It easy to show, that if the moving front evolves according to

$$\frac{d}{dt}\partial R(p,t) = \mathcal{F}(p)\vec{\mathcal{N}}$$

for a given function $\mathcal{F}$, then the embedding function Φ deforms according to

$$\frac{d}{dt}\Phi(p,t) = \mathcal{F}(p)\,|\nabla\Phi(p,t)|$$

For this level-set representation, it is proved that the solution is independent of the embedding function Φ, and in our case is initialized as a signed distance function. Based on (18) and embedding (20) in Φ we obtain that minimizing the Geodesic Active Region Function is equivalent to searching a steady-state solution of the following equation:

$$\frac{d\Phi}{dt} = \left(\zeta\left[h\mathcal{K} + \nabla h \cdot \nabla\Phi\right] + \alpha\left[\sum_{j=1}^{\mathrm{F_N}} w_j \mathbf{log}\frac{p_{t_{R_r}j}(\mathbf{I}_j)}{p_{t_{R_0}j}(\mathbf{I}_j)}\right]\right)|\nabla\Phi| \tag{22}$$

where the $\partial R(p,t)$ is represented by a level-set of Φ and the value of $\mathcal{K}$ is estimated on Φ, $\mathcal{K} = \mathbf{div}(\nabla\Phi/|\nabla\Phi|)$.

4 Front Propagation Algorithms

A direct implementation approach of equation (22) involves the re-estimation of the characteristic image of all the level set pixels (not simply the zero level set corresponding to the front itself). This front evolution method is computationally very expensive, due to many useless operations that are performed during the front propagation (especially in pixels which are out of interest). In order to overcome this drawback two different methods have been proposed: (i) the "Narrow Band" method that works with a small percentage of pixels (those which are around to the latest estimation of the contour) [2], (ii) the "Hermes" method, a fast approach suitable to a large variety of applications [28].

4.1 Narrow Band Approach

The key idea is to deal only with pixels which are close to the latest estimation of the zero level-set contour in both directions (inwards and outwards). This is known as Narrow Band Approach [2], and proposes to modify the level-set method so that it only affects the points

close to the current propagating contour. This band is created dynamically based on the actual propagating contour, by including points that lie less than some given distance away from the actual contour points (*band size*). The problem is that the contour position changes dynamically (from iteration to iteration), as well as the narrow band pixels. The estimation of the contour position from iteration to iteration increases dynamically the cost (in terms of complexity), thus the contour position is re-estimated only in cases where the contour is very close to the borders of the band. The selection of the band size affects significantly the efficacy of this algorithm. A significant cost reduction is achieved through this approach (compared to the classic method), but the cost remains considerable.

4.2 Hermes Algorithm

Hermes algorithm was originally proposed in [28], and tries to combines two well-known level-set algorithms, the Narrow Band and the Fast Marching [34]. Despite the fact that Fast Marching is a very fast algorithm, it cannot be used for our case since it demands a curve which evolves using an only positive or negative speed function during its evolution. In our case this is not-valid, because of the region-statistics forces. As it is explained, these forces shrink the curve if it is located on the dominant texture region, otherwise they expand it.

The main characteristic of *Hermes* algorithm is that it proposes a fast way to deform the initial curve towards the global minimum of the objective function. In our case the equation which deforms the initial curve (22) can be rewritten in a more general form as:

$$\mathbf{\Phi}^{t+1}_{(x,y)} = \mathbf{\Phi}^{t}_{(x,y)} + \mathcal{V}(x, y, \Phi)dt \tag{23}$$

where $\mathcal{V}(x, y, \Phi)$ is the speed function, depending on geometric features (*curvature*) and image features (*"edge-based" forces and "region-based" forces*). Since the speed $\mathcal{V}(x, y, \mathbf{\Phi})$ is basically estimated according to image characteristics, there are some points for which the front evolves faster compared to the others. The key idea on which Hermes approach is based, is to evolve the contour according to the speed values of its points. The algorithm at each step selects the point with the biggest speed from a set of actual contour points, and deforms the level-set frame locally.

First we initialize the contour and we set both the contour points and their neighbors as *active*. We select from the set of active points the one with the biggest speed and we iteratively modify the level-set frame for this point as well as for its neighborhood using (23). This operation is applied for a certain number of iterations. Since there are modified level-set frame values, there are some affected active points (in terms of *curvature*), as well as some affected neighborhood points which are not *active*, that we add to the set of *active*. If the level-set value of the selected point changes sign, which means that the contour deforms locally, we remove this point from *Active* ones. Periodically we find the contour position

in order to avoid the creation of a large set of *Active* pixels and we reinitialize the level-set frame using a distance function, since this frame is partially modified. Concerning the neighborhood definition we use a 5×5 centralized window, while the number of the locally applied iterations varies between 10 and 50. Both parameters affect drastically the CPU computational cost.

The key issue for an efficient version of the Hermes algorithm lies on a fast way of locating the grid point among the Active points with the biggest speed. For this reason, a variation of a heap-sort algorithm is used. Initially all the narrow points are sorted in a heap-sort (so that the smallest member can be easily located). When a point is removed from the heap-sort, the values of its neighbors are recomputed, and the results are bubbled upwards until they reach their correct locations. Moreover, whenever we want to add a point to the heap-sort, we put it at the end and we process it in the same way.

5 Experimental Results, Conclusions and Discussion

In this paper, we presented some new ideas concerning the integration of boundary-finding techniques and region-based approaches for texture segmentation.

5.1 Implementation Issues

Although we proposed a complete model, there are some technical aspects which affects its efficiency. These aspects consist of: the filter bank selection, the number of multi-grid levels, the contour initialization, and the Level-Set algorithm selection. We will try to give an idea of how we deal with these aspects.

- **Filters Bank Determination**
  This problem involves the definition of the Filter Bank, which includes three aspects: the filters number, their selection, and their size. We can say that the first two problems are strongly related. It is quite obvious that a "good" selection of filters decreases their required number to discriminate the texture prototypes. Experimentally, we could say that in most of the cases, we use five or six filters, but these filters are manually selected to be optimal, *i.e.* to extract features which are clearly discriminated between the different prototypes. The filter size holds an important role on the features extraction process. The texture homogeneity cannot be captured using a small window, since this homogeneity appears in the forms of repeated "patterns". On the other hand a big filter size creates problems between the boundaries of different texture regions and makes their localization quite problematic in terms of accuracy. Experimentally the filter size is selected to be 10×10 or 15×15.

- **Multi-grid Levels**
  The selection of multi-grid levels is strongly related to the size of the input frames, in terms of the pyramid compensation cost. Since this approach is not based on the construction of a frame pyramid, we can easily go to very low resolution levels. This is possible due to the use of the full data observation space in each level. Experimentally, in most of the cases, this number is selected to be either three or four.

- **Contour Initialization**
  The contour initialization is low-risk step. Supposing that the dominant texture is known, then we can initialize the contour at the borders of the input frame, and the proposed model will deform it towards the global minimum of the objective function [**fig.** (7,8,10,12)]. Additionally the model, deforms successfully contours which have been initialized randomly, towards the global minimum of the objective function [**fig.** (9,11,13)]. This is achieved thanks to the *region-statistics forces* which deform the contour outwards to regions different from the dominant one. This "random" initialization supposes that a part of each texture prototype appears in the region inside the initial contour.

- **Level-Set Propagation Algorithm**
  The selection of the Level-Set implementation algorithm is not a crucial step. Concerning the Narrow Band approach, it converges smoothly towards the optimal solution, but it is time consuming. On the other hand, Hermes algorithm deforms faster the initial contour towards the optimal solution, but it doesn't ensure a smooth propagation since the Level-Set frame at each step is updated locally. As a consequence the Level-Set frame refinement must be performed quite often.

5.2 Experimental Results

Real-word texture frame, as well synthetic texture frames have been used to test and validate the proposed approach.

Concerning the synthetic case some texture prototypes have been selected from a database of texture images. As a first step the system is taught on these prototypes by applying the bank of preselected filters, and analyzing their responses. The output of this operation is the creation of a global statistical description model for each prototype. Then a synthetic frame is created where regions of the selected texture prototypes appear randomly. This is considered as the input frame, on which the same bank of preselected filters is applied. Then using the different filter responses, as well as the texture description models, the Geodesic Active Region models is activated, and deforms the initial curve to the optimal solution of the texture segmentation problem.

The first experimental result [**fig.** (7)] involves a texture synthesis frame composed of two quite different prototypes (in terms of intensity distributions). As a consequence, the number of required filters is small, and the segmentation process is easily performed. On the contrary, the second example [**fig.** (8,9)] involves a texture synthesis frame with two quite similar prototypes (in terms of intensity distributions). Additionally, the non-dominant texture prototype is not quite homogeneous (in terms of a repeated pattern). For this case, two different contour initializations have been used: the "proper" (at the borders of the input frame) [**fig.** (8)], and a random one which contains a part of the dominant texture as well as a part of the the non-dominant texture [**fig.** (9)]. As it is shown for this example, the contour shrinks if it is located in the dominant texture region; otherwise the contour is expanded. Finally, the last experimental result [**fig.** (10,11)] involves a texture synthesis frame with five different texture regions, where two different contour initializations are shown. The large number of different texture regions requires the selection of a representative filter bank. In this example, the famous level-set property of changing the topology is demonstrated, where the initial curve breaks into multiple curves corresponding to the different texture regions.

On the contrary for the real case, the inverse operation is followed since the input is the real-texture synthesis frame [**fig.** (12,13,14,15,16)]. Small patterns are selected to represent the different texture prototypes appearing to this frame, and the system is taught with these patterns. Then the same process is followed as in the case of synthetic texture frames. Concerning the firt "real-world" example that consists two demonstrations (zebra and chita photos) [**fig.** (12,13,14)], we select from a 256×256 textured frame three different window patterns 64×64 (resp. 96×96) [**fig.** 12.1] (resp. [**fig.** 14.1]) that are the different texture prototypes, and based on these patterns we activate the Geodesic Active Region Model which segments quite well the different texture regions. The independence of the model from the contour initialization is clearly presented using two completely different contour initializations. The second "real-world" example related to medicine [**fig.** (15,16)], is a microscopic medical breast image, which exhibits an inflammatory carcinoma with metastasis. Three different texture patterns have been selected [**fig.** 15.1], and nine different filters have been applied, which give as output the texture description models. The power of the Geodesic Active Region model is demonstrated using two different contour initializations. The first one is at the borders of the frame [**fig.** (15)], while the second contains only non-dominant texture regions [**fig.** (16)].

5.3 Discussion and Conclusion

Summarizing, we have considered a Contour Propagation approach for texture segmentation. The main contribution of our approach is the Geodesic Active Region Model, a contour propagation model for texture segmentation, which **incorporates the existing approaches** in the domain **of texture analysis**, as well as in the domain **of texture**

segmentation. Firstly, a global texture description model is generated for each texture prototype. This model is obtained by fusing filtering theory and statistical analysis, where various filters responses are modelized statistically using mixtures synthesis of Gaussian distributions. The second step, consists of creating a global segmentation framework, where region based and boundary finding techniques are cooperating in a coupled common model. The contour propagation is obtained by integrating two different modules, region based module and boundary finding module in a common model. This leads to a system where the two modules operate simultaneously, where the contour propagation is guided by *smoothing*, *"edge-based"*, and *statistics region* forces. The main advantage of our model is that the contour evolves using information not only among it, but also information which come from the regions inside and outside of it. The changes of topology can be easily obtained using a Level-Set approach, thereby several texture regions can be detected simultaneously. The noise presence is confronted multi-grid approach, which simultaneously decreases the computational cost.

Various experimental results (in MPEG format), including the ones shown in this article, can be found at:

http://www.inria.fr/robotvis/personnel/nparagio/demos.html

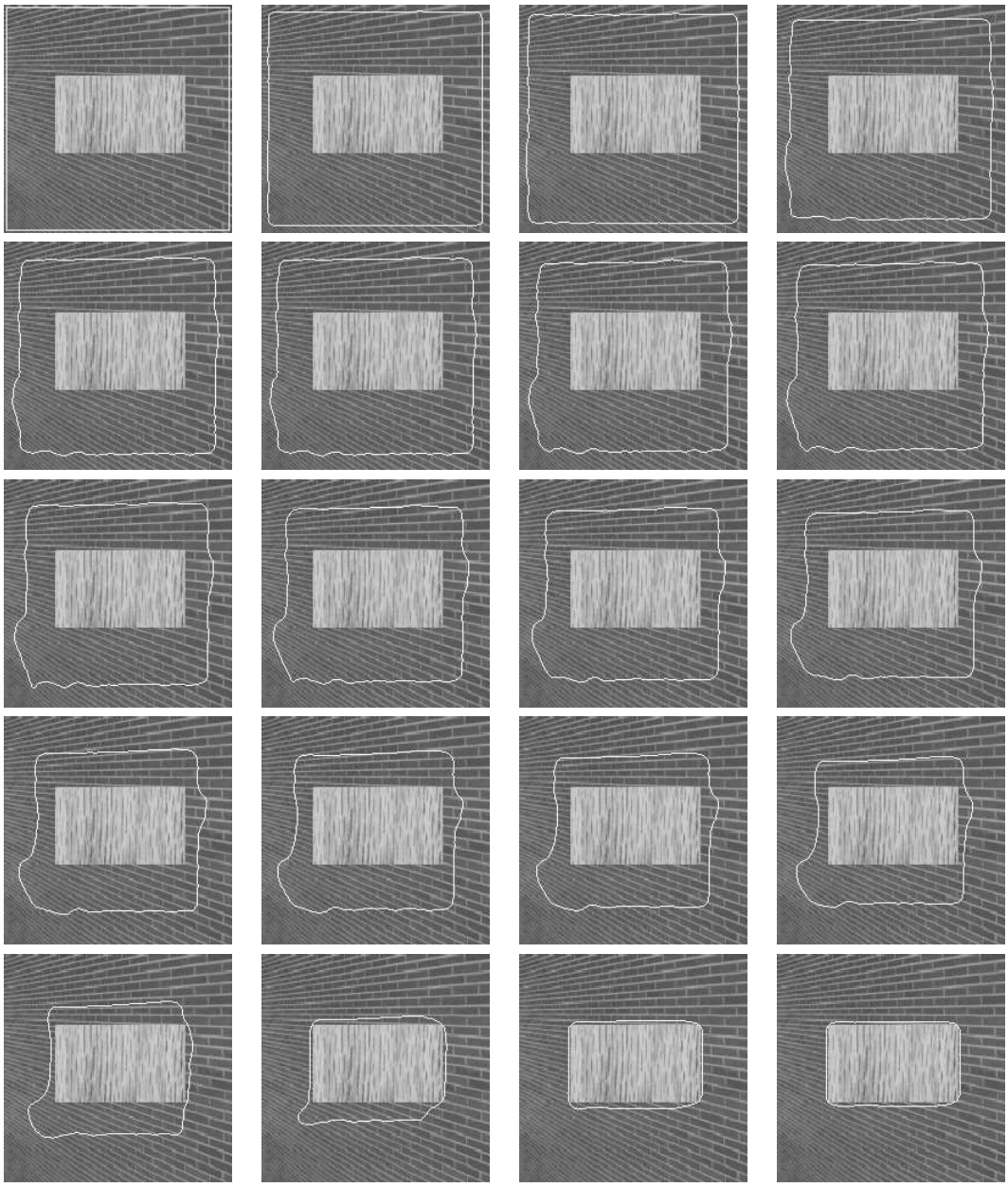


Figure 7: Curve Propagation (left to right, top to down), Texture Prototypes:*2*, Number of applied filters:*3*: *Intensity, Laplacian of Gaussian, Spectrum Analyzer of Gabor* $(2\pi, 2\pi)$

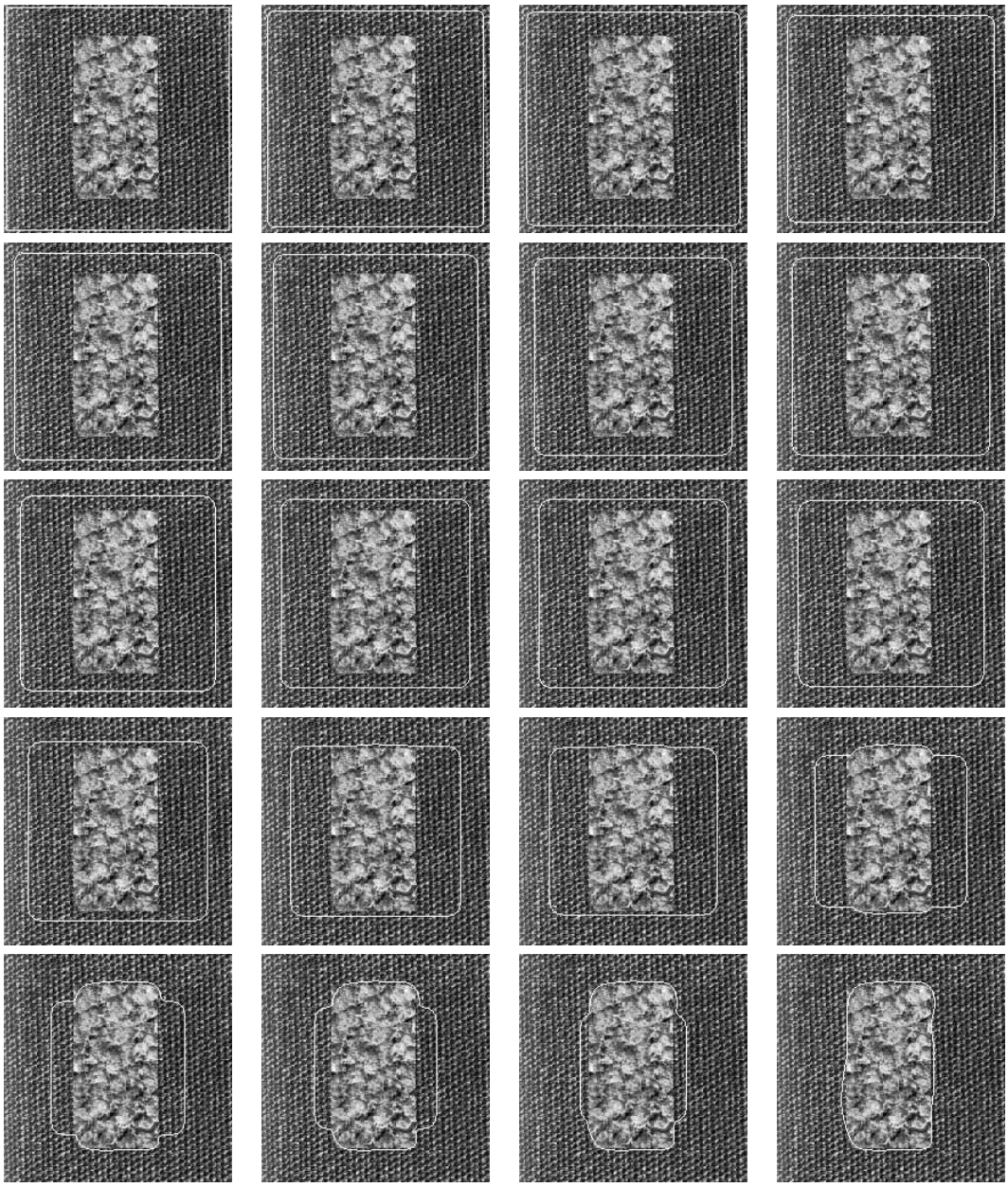


Figure 8: Curve Propagation (left to right, top to down), Texture Prototypes: *2*, Number of applied filters:*4*, *Intensity, Laplacian of Gaussian, Spectrum Analyzer of Gabor* $(2\pi, 2\pi)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, 0)$

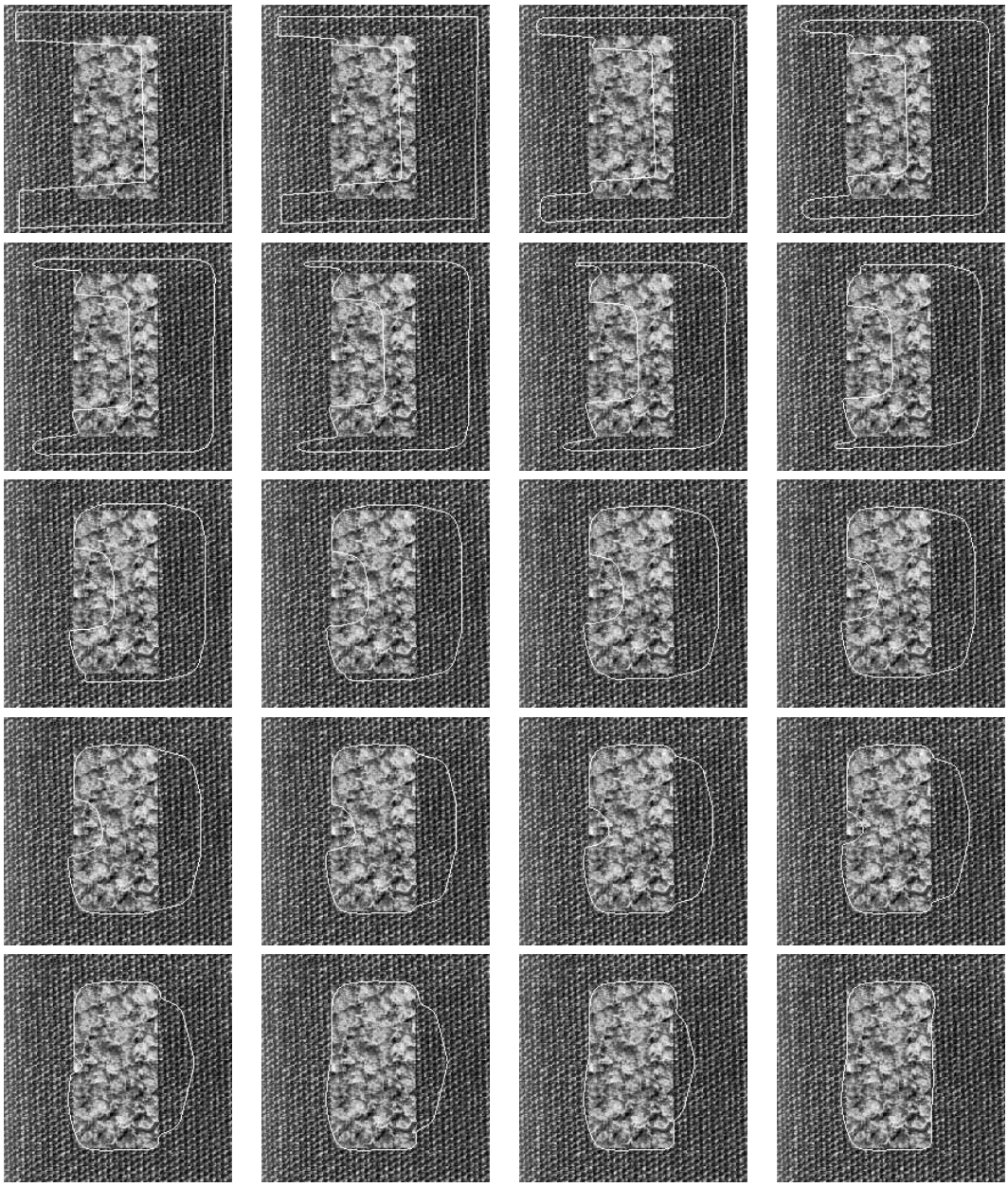


Figure 9: Curve Propagation (left to right, top to down), Texture Prototypes: *2*, Number of applied filters: *3*: *Intensity, Spectrum Analyzer of Gabor* $(2\pi, 2\pi)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, 0)$

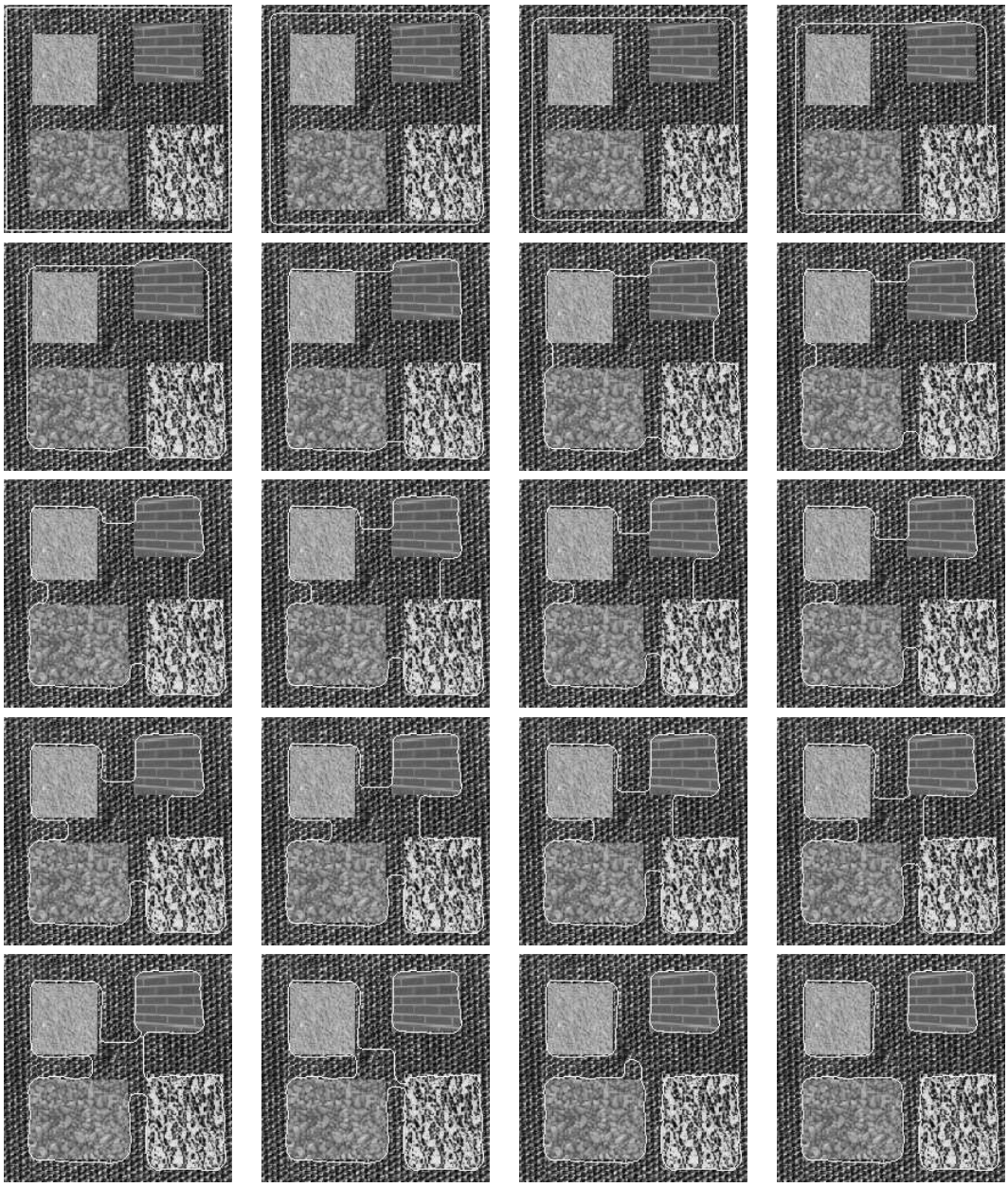


Figure 10: Curve Propagation (left to right, top to down), Texture Prototypes:*5*, Number of applied filters:*5*: *Intensity, Laplacian of Gaussian, Spectrum Analyzer of Gabor* $(2\pi, 2\pi)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, 0)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{2}, \frac{\pi}{3})$

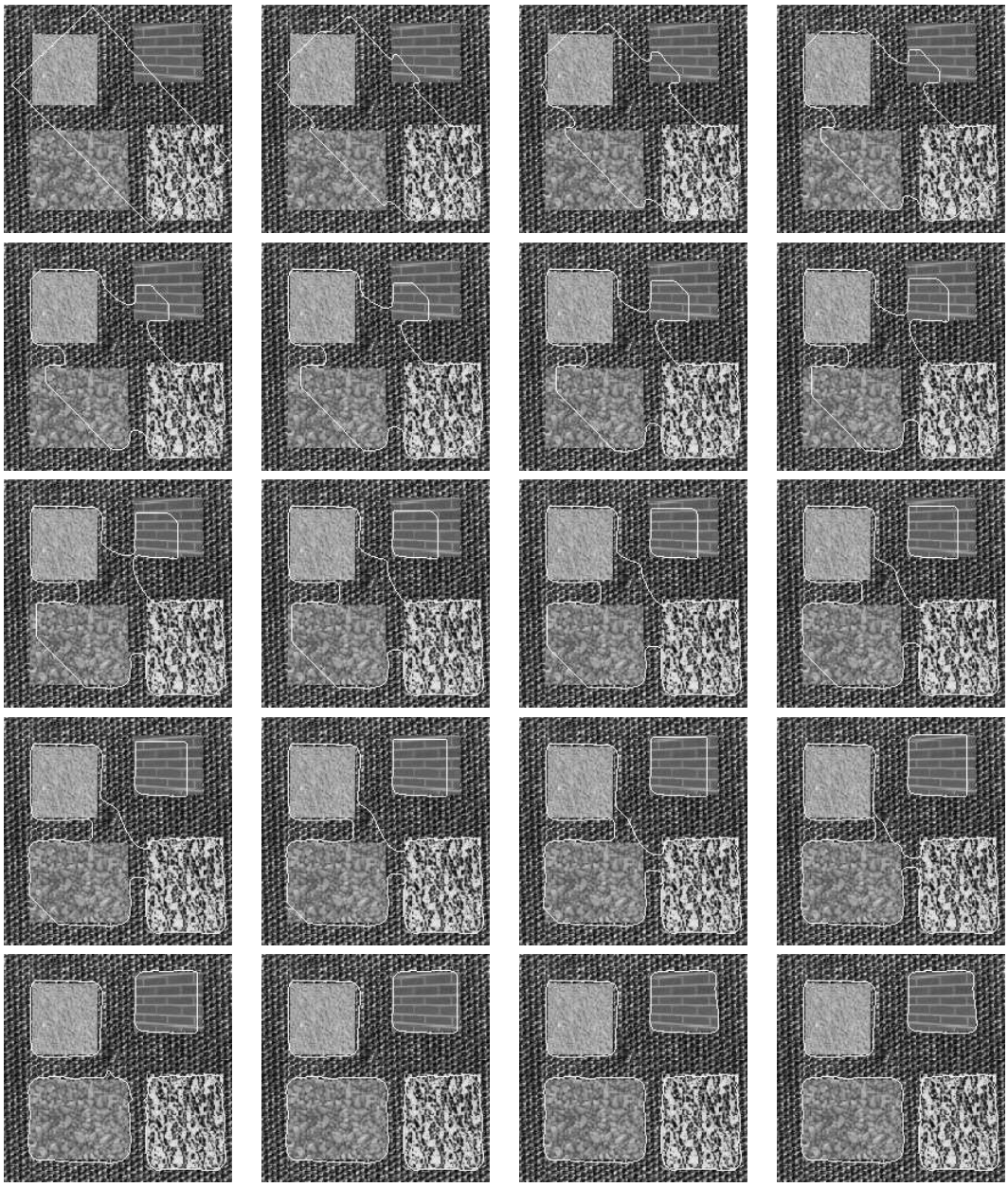


Figure 11: Curve Propagation (left to right, top to down), Texture Prototypes:*5*, Number of applied filters:*5*: *Intensity, Laplacian of Gaussian, Spectrum Analyzer of Gabor* $(2\pi, 2\pi)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, 0)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{2}, \frac{\pi}{3})$

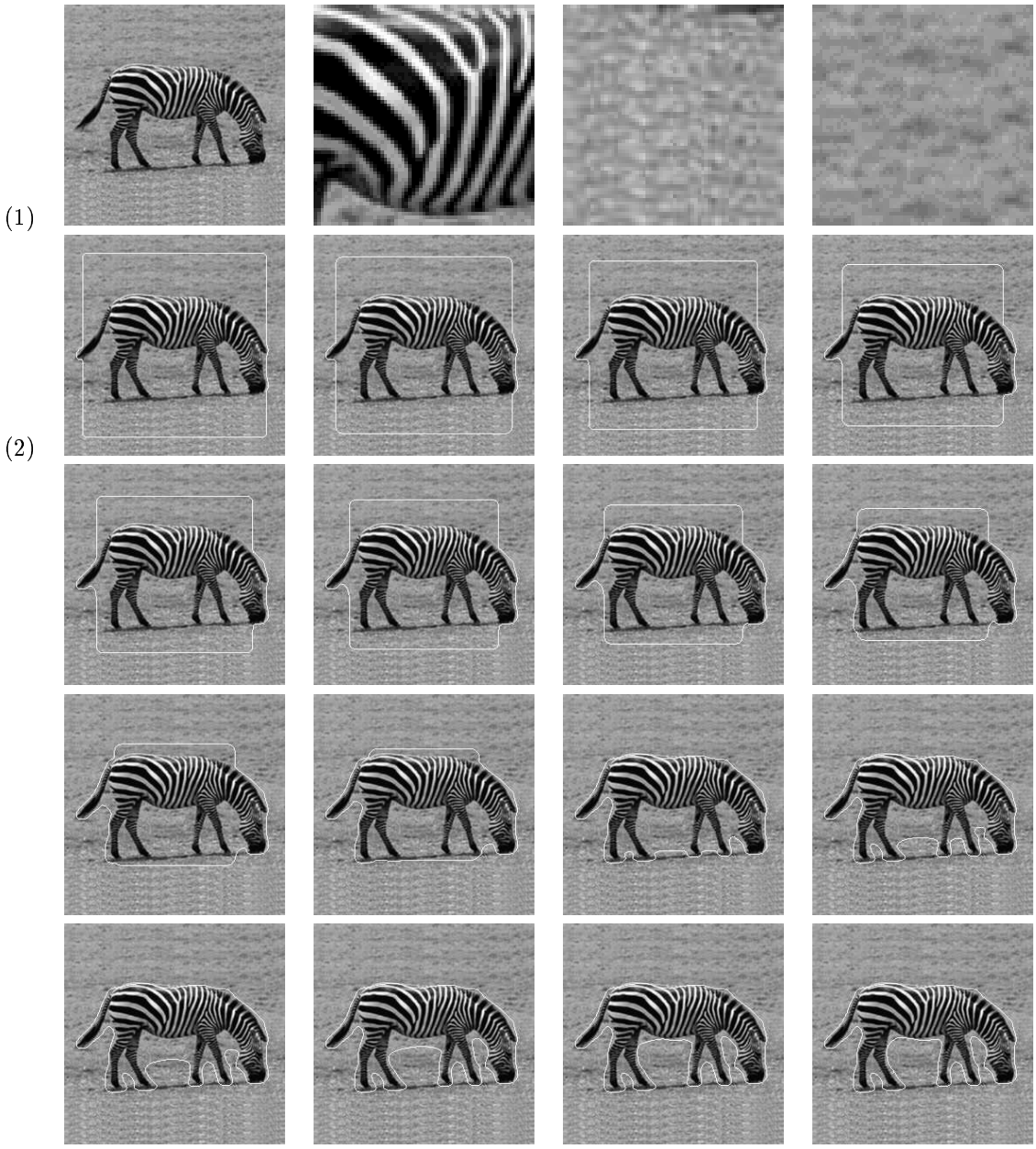


Figure 12: (1) Texture prototypes, (2) Curve Propagation (left to right, top to down), Number of applied filters:*6*: *Intensity, Isotropic Directional Derivatives, Spectrum Analyzer of Gabor* $(2\pi, 2\pi)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, 0)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{3}, 0)$*, Spectrum Analyzer of Gabor* $(0, \frac{\pi}{3})$

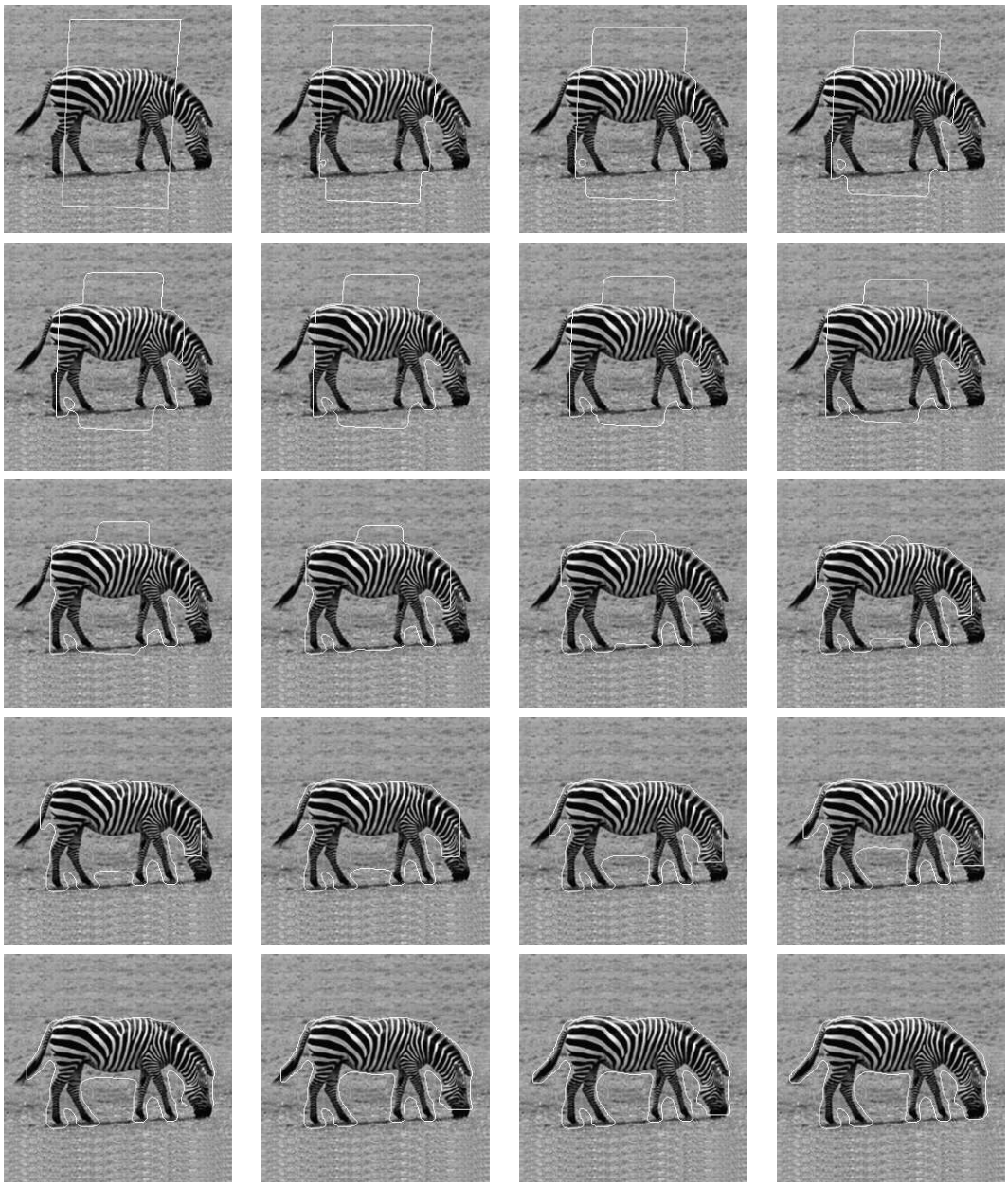


Figure 13: Curve Propagation (left to right, top to down), Texture Prototypes:*3*, Number of applied filters:*6*: *Intensity, Isotropic Directional Derivatives, Spectrum Analyzer of Gabor* $(2\pi, 2\pi)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, 0)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{3}, 0)$*, Spectrum Analyzer of Gabor* $(0, \frac{\pi}{3})$

(1)

(2)

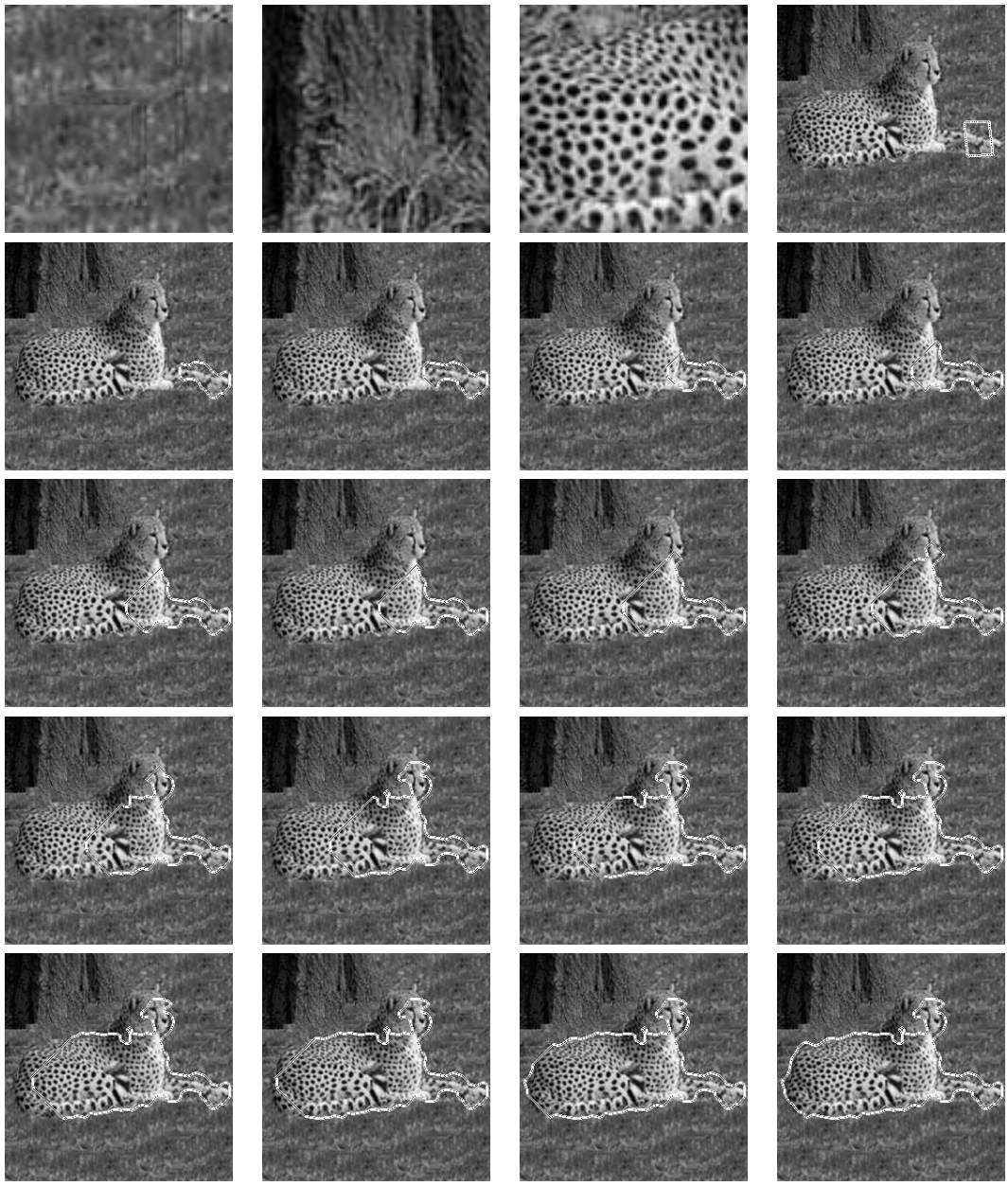


Figure 14: (1) Texture prototypes, (2) Curve Propagation (left to right, top to down), Number of applied filters:*8*: *Intensity, Spectrum Analyzer of Gabor* $(2\pi, 2\pi)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, 0)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{3}, 0)$*, Spectrum Analyzer of Gabor* $(0, \frac{\pi}{3})$ *Spectrum Analyzer of Gabor* $(0, \frac{\pi}{6})$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, \frac{\pi}{3})$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{3}, \frac{\pi}{6})$

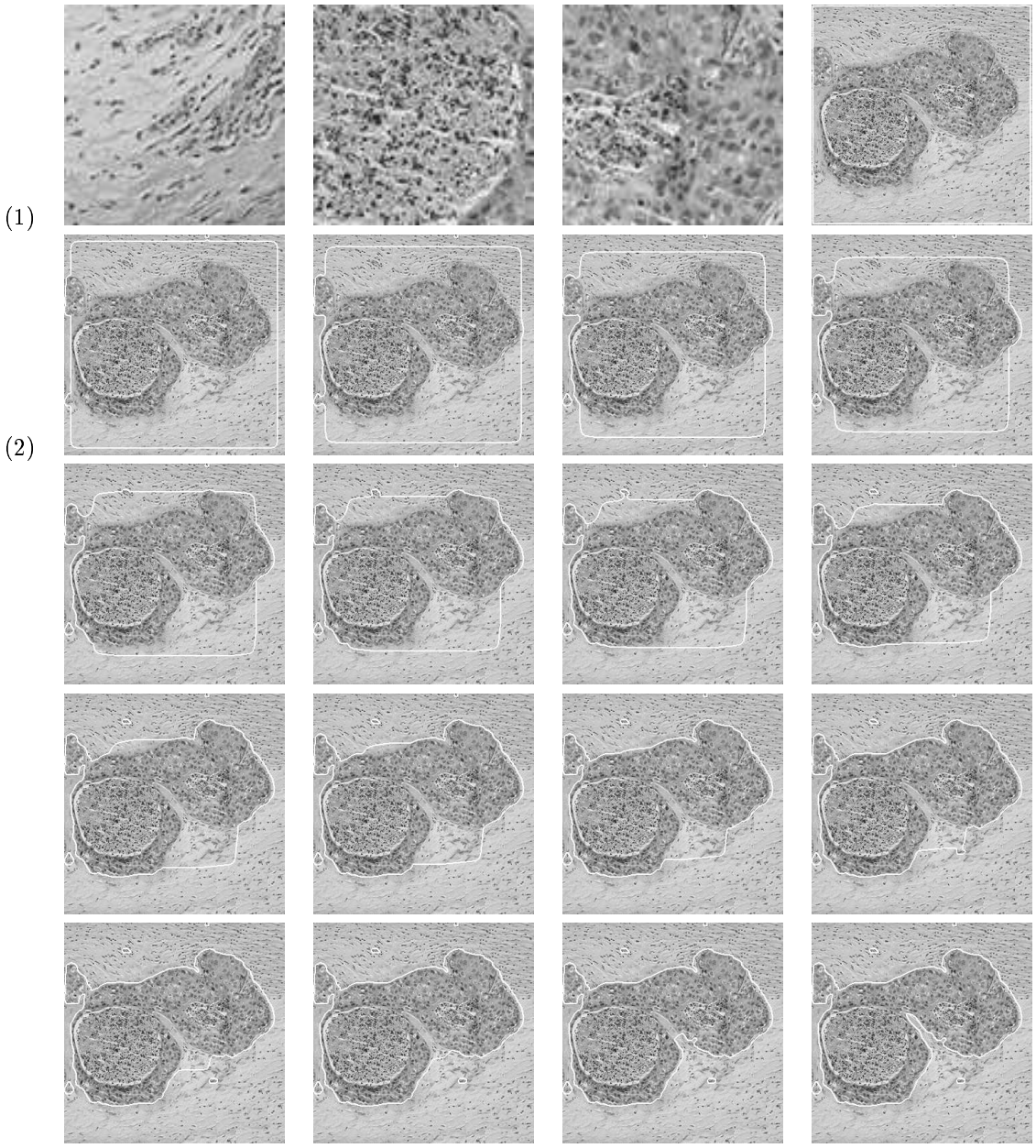


Figure 15: (1) Texture prototypes: 3, (2) Curve Propagation (left to right, top to down), Number of applied filters:9: *Intensity, Laplacian of Gaussian, Spectrum Analyzer of Gabor* $(2\pi, 2\pi)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, 0)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{3}, 0)$*, Spectrum Analyzer of Gabor* $(0, \frac{\pi}{6})$*, Spectrum Analyzer of Gabor* $(0, \frac{\pi}{3})$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, \frac{\pi}{6})$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{3}, \frac{\pi}{3})$

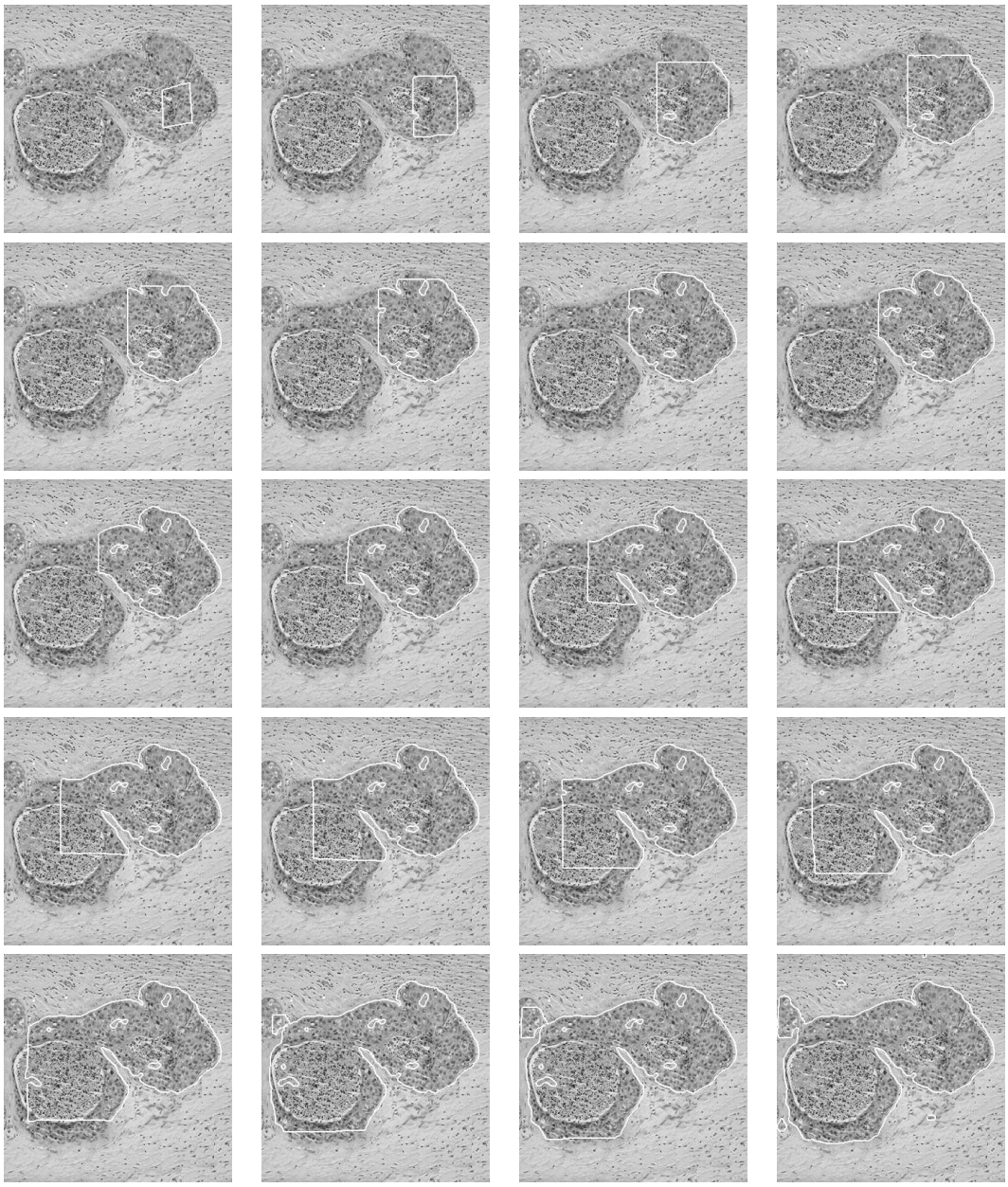


Figure 16: Curve Propagation (left to right, top to down), Texture Prototypes:*3*, Number of applied filters:*9*: *Intensity, Laplacian of Gaussian, Spectrum Analyzer of Gabor* $(2\pi, 2\pi)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, 0)$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{3}, 0)$*, Spectrum Analyzer of Gabor* $(0, \frac{\pi}{6})$*, Spectrum Analyzer of Gabor* $(0, \frac{\pi}{3})$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{6}, \frac{\pi}{6})$*, Spectrum Analyzer of Gabor* $(\frac{\pi}{3}, \frac{\pi}{3})$

References

[1] Boston, MA, June 1995. IEEE Computer Society Press.

[2] D. Adalsteinsson and J. A. Sethian. A Fast Level Set Method for Propagating Interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.

[3] Gilles Aubert and Laure Blanc-Feraud. An elementary proof of the equivalence between 2d and 3d classical snakes and geodesic active contours. Technical Report 3340, INRIA, January 1998.

[4] Bénédicte Bascle and Rachid Deriche. Region tracking through image sequences. In *Proceedings of the 5th International Conference on Computer Vision* [1], pages 302–307. Appeared also as an Inria Research Report RR-2439 (Dec. 1994).

[5] A. Bovik, C. Analysis of multichannel narrow-band filters for image texture segmentation. *IEEE Trans. Signal Processing*, 39(9):2025–2043, 1991.

[6] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):769–798, November 1986.

[7] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of the 5th International Conference on Computer Vision* [1], pages 694–699.

[8] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *The International Journal of Computer Vision*, 22(1):61–79, 1997.

[9] A. Chakraborty, H. Staib, and J. Duncan. Deformable Boundary Finding in Medical Images by Integrating Gradient and Region Information. *IEEE trans.Medical Imaging*, 15(6):859–870, 1996.

[10] L.D. Cohen. On active contour models and balloons. *CVGIP: Image Undertanding*, 53:211–218, 1991.

[11] G. Cross and A. Jain. Markov random field texture models. *IEEE, PAMI*, 5:25–39, 1983.

[12] R. Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *The International Journal of Computer Vision*, 1(2):167–187, May 1987.

[13] D. Dunn and W. Higgins. Optimal gabor filters for texture segmentation. *IEEE Trans. Image Processing*, 4(7):947–964, 1995.

[14] D. Gabor. Theory of communications. *IEE Proceedings*, 93(26), 1946.

[15] F. Heitz, P. Perez, and P. Bouthemy. Multiscale minimization of global energy functions in some visual recovery problems. *GVGIP: Image Unterstanding*, 59:125–134, 1994.

[16] A. Jain and R. Farrokhsia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24:1167–1186, 1991.

[17] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision*, pages 259–268, London, June 1987.

[18] C. Kervrann and F. Heitz. A Markov Random-Field Model-Based Approach to Unsupervised Texture Segmentation Using Local and Global Spatial Statistics. *Image Processing*, 4(6):856–862, June 1995.

[19] A. Khotanzad and J.Y. Chen. Unsupervised Segmentation of Textured Images by Edge Detection in Multidimensional Features. *IEEE, PAMI*, 11(4):414–420, April 1989.

[20] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proc. Fifth International Conference on Computer Vision* [1].

[21] A. Leonardis, A. Cupta, and R. Bajcsy. Segmentation of range images as the search for geometric parametric models. *The International Journal of Computer Vision*, 14(3):253–270, 1995.

[22] R. Malladi, J. A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *PAMI*, 17(2):158–175, February 1995.

[23] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$. *Trans. Amer. Math. Soc.*, 315:69–87, 1989.

[24] B.S. Manjunath and R. Chellappa. Unsupervised Texture Segmentation Using Markov Random Field Models. *IEEE, PAMI*, 13(5):478–482, May 1991.

[25] J. Mao and A. Jain. Texture classification and segmentation using multiresolution simultaneous autoregresive models. *Pattern Recognition*, 25:173–188, 1992.

[26] J.J. More. The levenberg-marquardt algorithm, implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, Lecture Notes in Mathematics 630. Springer-Verlag, 1977.

[27] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed : algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[28] N. Paragios and R. Deriche. A PDE-based Level Set Approach for Detection and Tracking of Moving Objects. Technical Report 3173, INRIA, France, May 1997. http://www.inria.fr/rapports/sophia/RR-3173.html.

[29] N. Paragios and R. Deriche. A PDE-based Level Set Approach for Detection and Tracking of Moving Objects. In *Proceedings of the 6th International Conference on Computer Vision*, Bombay,India, January 1998. IEEE Computer Society Press.

[30] Nikolaos Paragios and Rachid Deriche. Detecting multiple moving targets using deformable contours. In *International Conference on Image Processing*, volume II of III, pages 183–186, Santa-Barbara,California, October 1997.

[31] A. Pentland. Automatic Extraction of Deformable Part Models. *The International Journal of Computer Vision*, pages 107–126, 1990.

[32] H.M. Raafat and A.K.C. Wong. Texture Information-Directed Region Growing Algorithm for Image Segmentation and Region Classification. *CVGIP*, 43(1):1–21, July 1988.

[33] T.R. Reed, H. Wechsler, and M. Werman. Texture Segmentation Using a Diffusion Region growing technique. *Pattern Recognition*, 23(9):953–960, September 1990.

[34] J. A. Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. In *Proc. Nat. Ac. Science*, volume 93, pages 1591–1694, 1996.

[35] P. Simoncelli, W. Freeman, H. Adelson, and Heeger J. Shiftable multiscale transforms. *IEEE trans. on Information Theory*, 38:587–607, 1992.

[36] S.C. Zhu, T.S. Lee, and A.L. Yuille. Region Competition: Unifying Snakes, Region Growing Energy/Bayes/MDL for Multi-band Image Segmentation. In *proc. of 5th International Confereence on Computer Vision*, Boston, USA, June 1995.

[37] Song Chun Zhu. *Statistical and Computational Theories for Image Segmentation, Texture Modeling and Object Recognition.* PhD thesis, Harvad University, January 1996.

[38] Song Chun Zhu and Alan Yuille. Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, September 1996.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399