

Copyright © 1997, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**IMAGE SEGMENTATION AND EDGE DETECTION
VIA CONSTRAINED DIFFUSION AND ADAPTIVE
MORPHOLOGY: A CNN APPROACH TO BUBBLE-
DEBRIS IMAGE ENHANCEMENT**

by

Csaba Rekeczky, Abraham Schultz, István Szatmári,
Tamás Roska, and Leon O. Chua

Memorandum No. UCB/ERL M97/96

15 December 1997

COVER PAGE

**IMAGE SEGMENTATION AND EDGE DETECTION
VIA CONSTRAINED DIFFUSION AND ADAPTIVE
MORPHOLOGY: A CNN APPROACH TO BUBBLE-
DEBRIS IMAGE ENHANCEMENT**

by

**Csaba Rekeczky, Abraham Schultz, István Szatmári,
Tamás Roska, and Leon O. Chua**

Memorandum No. UCB/ERL M97/96

15 December 1997

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

IMAGE SEGMENTATION AND EDGE DETECTION VIA CONSTRAINED DIFFUSION AND ADAPTIVE MORPHOLOGY: A CNN APPROACH TO BUBBLE-DEBRIS IMAGE ENHANCEMENT

Csaba Rekeczky⁺, Abraham Schultz^{**}, István Szatmári⁺*,
Tamás Roska⁺* and Leon O. Chua⁺*

⁺Electronics Research Laboratory, College of Engineering,
University of California at Berkeley, Berkeley, CA 94720, USA

^{**}Naval Research Laboratory, Radar Division,
4555 Overlook Ave., S.W., Washington D.C. 20375, USA

^{*}Analogical and Neural Computing Laboratory, Computer and Automation Institute,
Hungarian Academy of Sciences, H-1111 Budapest, Kende u. 13-17, Hungary

E-mail: rcsaba@fred.eecs.berkeley.edu, schultz@radar.nrl.navy.mil, szatmari@sztaki.hu,
roska@sztaki.hu, chua@fred.eecs.berkeley.edu

Abstract: *In this paper, a CNN based locally adaptive scheme is presented for image segmentation and edge detection. It is shown that combining a constrained (linear or nonlinear) diffusion approach with adaptive morphology leads to a robust segmentation algorithm for a large class of image models. These images comprise of simple geometrical objects, each having a homogeneous gray-scale level and they might be overlapping. The background illumination is inhomogeneous, the objects are corrupted by additive Gaussian noise and blurred by low-pass filtering type effects. Typically, this class has a multimodal (in most cases bimodal) image histogram and no special (easily exploitable) characteristics in the frequency domain. The synthesized analogic (analog and logic) CNN algorithm combines a diffusion-type filtering with a locally adaptive strategy based on estimating the first order (mean) and second order (variance) statistics and deals efficiently with degrading effects in the segmentation. Both PDE and non-PDE related diffusion schemes are examined in the CNN framework. We show how in simple cases the proposed algorithm reduces to a fixed-thresholding method and a DoG (difference of Gaussians)-type operator. On the other hand, the superiority and robustness of the proposed computational technique is examined in more complex examples when the parameters of the above image model are significantly altered. The VLSI implementation complexity and some robustness issues are carefully analyzed and discussed in detail. All algorithmic steps are realized using nearest neighbor CNN templates. It is argued that a simplified version of the algorithm can be realized using the already available CNN Universal Chips and the entire solution fits into the frame of the "next generation". A number of tests have been completed within the frame of the so-called "bubble-debris" classification experiments on original and artificial gray-scale images.*

Table of Contents

| | |
|---|-----------|
| 1. INTRODUCTION | 3 |
| 2. PRELIMINARIES IN THE BUBBLE-DEBRIS CLASSIFICATION EXPERIMENTS | 3 |
| 3. THE CNN FRAME OF COMPUTING AND THE IMAGE MODEL | 4 |
| 3.1 CNN CORE CELL AND THE INTERCELL INTERACTIONS | 4 |
| 3.2 CNN TEMPLATES AND THEIR MINIMAL CODING | 5 |
| 3.3 THE IMAGE AND NOISE MODEL AND THE SEGMENTATION APPROACH | 7 |
| 4. CNN BASED DIFFUSION MODELS: GENERAL FORMULATION | 7 |
| 4.1 PDE BASED APPROACH - CONSTRAINED LINEAR AND NONLINEAR DIFFUSION | 8 |
| 4.2 NON-PDE BASED APPROACH - A NOVEL-TYPE NONLINEAR DIFFUSION FORMULATION..... | 11 |
| 5. ANALOGIC ALGORITHM DESIGNED FOR SEGMENTATION AND EDGE DETECTION | 14 |
| 5.1 DIFFUSION BASED PRE-FILTERING STRATEGIES..... | 14 |
| 5.2 LOCAL THRESHOLD ESTIMATION | 16 |
| 5.3 LOCALLY ADAPTIVE SEGMENTATION AND EDGE DETECTION | 16 |
| 6. ANALYSIS OF THE ALGORITHM PERFORMANCE | 17 |
| 7. IMPLEMENTATION IN A CNN UNIVERSAL CHIP ENVIRONMENT | 18 |
| 7.1 STABILITY AND ROBUSTNESS OF LINEAR AND NONLINEAR TEMPLATES..... | 19 |
| 7.2 VLSI IMPLEMENTATION- AND COMPUTATIONAL COMPLEXITY | 19 |
| 8. CONCLUSIONS | 20 |
| 9. ACKNOWLEDGMENT | 20 |
| 10. REFERENCES | 21 |

1. Introduction

Since their invention in 1988 ([1]-[2]), the rapidly growing field of Cellular Neural Networks (CNNs) have found numerous potential applications, especially in image processing problems where real-time signal processing is required. The CNN approach evolved to a widely accepted computational paradigm [3], and recently its dedicated hardware architecture has also been designed, called the CNN Universal Machine (CNN UM, [4]). The CNN UM is the first parallel, stored program analogic and visual array microprocessor that can be fabricated on a single chip ([15]-[19]). The new device is programmed by analogic algorithms (see e.g. [35]-[38], [42]-[45] and [50]-[51]), i.e. using analog operations in sequence combined with local logic at the cell level.

In this paper, we complete and develop the framework first outlined in [39]-[40]¹, and describe a diffusion and adaptive morphology based CNN approach for image segmentation and edge detection. We investigate the problem of noise filtering, enhancement and segmentation in a bubble-debris classification experiment. The paper focuses on the algorithmic aspects of the problem and the strategies to be followed when tuning the global parameters. Gaining deeper insight into linear and nonlinear template design, stability and robustness analysis the reader is referred to other papers (e.g. [1]-[6], [8]-[12], [14], [39]).

2. Preliminaries in the Bubble-debris Classification Experiments

The bubble-debris classification project aims to develop a real-time warning system for solving the conditional based maintenance problem of helicopter (and jet) engines. This classification system should warn the pilot of a serious problem if the number of debris particles in the oil of the engine exceed a predefined threshold level indicating the probability of an engine failure. An off-line experimental system, a bubble-debris classifier, has already been developed and set up in NRL that is the first step toward the goal outlined.

The problem of distinguishing oil debris particles from air bubbles is difficult due to the coarse resolution of the images and the requirements for an extremely low false alarm rate for miss-classified bubbles. The approach followed in NRL in the development of a bubble-debris classifier was to tune the system parameters to achieve the desired false alarm rate and to minimize the percentage of miss-classified debris particles. The system operates as a series of rejection filters with increasing time consuming classification tasks applied to smaller and smaller number of objects. First single bubbles are removed using a relatively simple test involving the variance of a set of radii drawn through the objects center. Then double-bubbles are removed using an arc test. Finally multiple bubble groups are detected using the erosion operator to find the center of the largest

¹ The interested reader is also referred to [41] the revised version of these studies.

bubble in the group. The percentage of matching pixels of the entire boundary of the unknown object to the circumference of the hypothetical circle associated with the largest bubble in the group is then computed. The radius of this circle is the number of times the erosion operator was applied to get the “extinction set” (the set which vanishes with the next application of the erosion operator) and the center mass of the extinction set is taken to be the center of the circle. Matching is done to within a specified tolerance which has been taken to be about 1.5 pixels. If the unknown object is a debris particle the percentage of matching pixels will in most cases be relatively low.

The strategy of using a sequence of tests of increasing computational complexity was determined by the limitations of the standard (digital) single CPU computer. This approach makes the classification algorithm difficult to tune in that the thresholds used for one test can impact on subsequent tests. The CNN offers the possibility of a totally different approach to the bubble-debris classification problem. The features defined in the NRL system cannot be easily implemented in this environment and is necessary to define a new feature set.

The present study deals with CNN based front-end filtering strategies followed by segmentation, sorting and edge detection that in our hope should improve the input for the classification. The classification, based on morphological operations and implementing an autowave metrics in CNN, is developed and analyzed in a companion paper [52]. It should be noted that the highest frame rate of the image acquisition system in bubble-debris experiments can be as high as 2 milliseconds. A complex CNN algorithm can be executed within this time interval since most CNN operations (and reprogramming the hardware) are in the order of a few microseconds. Therefore, these experiments target the original goal: designing a real-time alarm system within the conditional based maintenance project.

3. The CNN Frame of Computing and the Image Model

In this Section we define the CNN frame of computing and define the general image model of the experiments. First, the core cell of the CNN array and the permitted intercell interactions, are presented (Subsection 3.1). Then, the minimal coded form of linear and nonlinear templates is specified, as the general form of the analog instruction set of the CNN Universal Machine (Subsection 3.2). Finally, in Subsection 3.3 the general image model of this study will be described.

3.1 CNN Core Cell and the Intercell Interactions

As the basic framework throughout this paper, let us consider a two dimensional CNN array in which the cell dynamics is described by the following nonlinear ordinary differential equation with linear and nonlinear terms (the extension to three dimensions is straightforward, allowing similar interlayer interactions):

$$\begin{aligned}
C \frac{d}{dt} v_{x_{ij}}(t) &= -R^{-1} v_{x_{ij}}(t) + \sum_{kl \in N_r} A_{ij,kl} v_{y_{kl}}(t) + \sum_{kl \in N_r} B_{ij,kl} v_{u_{kl}}(t) \\
&\quad + \sum_{kl \in N_r} \hat{D}_{ij,kl}(\Delta v(t)) + I_{ij} \\
v_{y_{ij}}(t) &= f(v_{x_{ij}}(t)) = 0.5 (|v_{x_{ij}}(t) + 1| - |v_{x_{ij}}(t) - 1|) \\
\text{where: } \Delta v(t) &= v_{kl}(t) - v_{ij}(t), \quad v = v_u(t) \vee v_x(t) \vee v_y(t) \\
|v_{x_{ij}}(0)| &\leq 1, \quad |v_{u_{ij}}(t)| \leq 1, \quad |I_{ij}| \leq I_{\max} \\
1 \leq i &\leq M, \quad 1 \leq j \leq N
\end{aligned} \tag{\epsilon 1}^2$$

where $v_{x_{ij}}$, $v_{u_{ij}}$, $v_{y_{ij}}$ are the state, input and output voltage of the specified CNN cell, respectively. The subscript ij refers to a grid point associated with a cell on the 2D grid, and $kl \in N_r$ is a grid point in the neighborhood within the radius r of the cell ij . $A_{ij,kl}$ represents the linear feedback, $B_{ij,kl}$ the linear control, while I_{ij} is the cell current which could be space variant. $\hat{D}_{ij,kl}$ is the generalized nonlinear term applied for $\Delta v = v_{kl}(t) - v_{ij}(t)$, the voltage difference of the input, state or output values. The output characteristics f is a sigmoid-type piecewise-linear function. The time constant of a CNN cell is determined by the linear capacitor (C) and the linear resistor (R) and it can be expressed as $\tau = RC$. Without loss of generality $R = 1$ and $C = 1$ will be considered.

3.2 CNN Templates and Their Minimal Coding

A CNN template³ is given with linear and nonlinear terms completed by the cell current. For both linear and nonlinear templates we specify the minimal coded analog instructions that correspond to a switch configuration (see an overall explanation in [4]) representing different functionality of a CNN universal cell. In Table 1 a possible ordering of the VLSI implementation complexity of different CNN hardware is listed and numbered taking into the consideration the current development trends in the analog VLSI technology (a detailed explanation can be found in Section 7). From now on the implementation complexity of a CNN solution (the ‘‘cost’’ of the hardware needed to implement a certain template) will be discussed and analyzed on this basis.

In the current study the general form for all linear templates is as follows:

$$A = \begin{bmatrix} a_2 & a_1 & a_2 \\ a_1 & a_0 & a_1 \\ a_2 & a_1 & a_2 \end{bmatrix}, \quad B = \begin{bmatrix} b_2 & b_1 & b_2 \\ b_1 & b_0 & b_1 \\ b_2 & b_1 & b_2 \end{bmatrix}, \quad I \tag{\epsilon 2}$$

Consequently, the minimal coded analog instruction that completely determines a linear template operation is given in the form of $[a_0 \ a_1 \ a_2 \ b_0 \ b_1 \ b_2 \ I \ B_c]$ where a_i stands for feedback, b_i for control coefficients, I is the cell current and B_c specifies the boundary condition (constant and zero-flux is considered).

² (ϵN) denotes the Nth equation, formula or template throughout the paper.

³ In this study each CNN template is identified by a code name associated by the processing task. See Table 2 and Table 3.

Similarly, the general form for all nonlinear templates (\hat{D} stands for an arbitrary nonlinear operator):

$$A \rightarrow a_0, \hat{D} = \begin{bmatrix} \eta_2 & \eta_1 & \eta_2 \\ \eta_1 & \eta_0 & \eta_1 \\ \eta_2 & \eta_1 & \eta_2 \end{bmatrix} d(\Delta v), d(\Delta v) = \begin{cases} n_i(\Delta v) & \text{if } m = 0 \\ n_i(\Delta v)\Delta v & \text{if } m = 1 \end{cases}, I \quad (\epsilon 3)$$

The nonlinear cell interactions used by the nonlinear templates are defined as the generalized piecewise-linear sigmoid (n_1) or the piecewise-linear radial basis (n_2) functions shown in Fig. 1. Their formulation is as follows ($\Delta\vartheta > 0$, $n_1: Q_1 Q_2 < 0$, $n_2: Q_1 - Q_2 \neq 0$):

$$n_1 = \begin{cases} Q_1 & \text{if } \Delta v > \vartheta + \Delta\vartheta \\ s(\Delta v - \vartheta) & \text{if } \vartheta + Q_2 / s \leq \Delta v \leq \vartheta + \Delta\vartheta, \text{ where: } s = Q_1 / \Delta\vartheta \\ Q_2 & \text{if } \Delta v < \vartheta + Q_2 / s \end{cases} \quad (\epsilon 4.1)$$

$$n_2 = \begin{cases} Q_2 & \text{if } |\Delta v - \vartheta| \geq \Delta\vartheta \\ Q_1 - s|\Delta v - \vartheta| & \text{if } |\Delta v - \vartheta| < \Delta\vartheta, \text{ where: } s = (Q_1 - Q_2) / \Delta\vartheta \end{cases} \quad (\epsilon 4.2)$$

The parameter vector $[\Delta v \ m \ n_i \ Q_1 \ Q_2 \ \vartheta \ \Delta\vartheta \ \eta_0 \ \eta_1 \ \eta_2 \ a_0 \ I]$ is the minimal coded analog program instruction and uniquely defines all nonlinear templates. In operator specification (Δv) 1, 2, 3 and 4 stands for Δv_{yy} , Δv_{uu} , Δv_{xx} and Δv_{ux} of the $\hat{D}_{ij,kl}$ term (these are the difference values that are used in template classes developed in previous studies, e.g. [47], [39], [13] and [14]). m distinguishes two different voltage controlled current sources: (1) $m = 0 \Rightarrow n_i(\Delta v)$ and (2) $m = 1 \Rightarrow n_i(\Delta v)\Delta v$, where n_i is the type of the nonlinearity and can be either n_1 or n_2 (both defined by 4 parameters $[Q_1 \ Q_2 \ \vartheta \ \Delta\vartheta]$). η_i reflects the spatial weighting of the nonlinearities, a_0 is the self-feedback value and I specifies the cell current. For all nonlinear templates zero-flux (ZF) boundary condition is required.

If not stated otherwise, the above setting is valid for all CNNs discussed throughout the paper. Table 2 and Table 3 contain the complete set of CNN templates used throughout the paper and the above minimal coded forms are used to specify the linear and nonlinear template values, respectively.

It is assumed that elementary logical (NOT, AND, OR) and arithmetical (ADD, SUB) operations are implemented and can be used on the cell level between LLM (local logical memory) and LAM (local analog memory) locations, respectively. In addition data transfer can be performed between LAMs and from LAMs to LLMs.

Further notations: U , X , Y , I , and M denotes the input, state, output, bias⁴ and mask⁵ image corresponding to a CNN layer, respectively. Certain operators (e.g. arithmetical) might have two

⁴ The bias (also referred to as the “bias map”) of a CNN layer is a gray-scale image. The bias map can be viewed as the space variant part of the cell current. Using pre-calculated bias maps “linear” spatial adaptivity can be added to the templates in CNN algorithms. If the bias map is not specified it is assumed to be zero.

input values, denoted by U_1 and U_2 . For simplicity, linear and nonlinear terms in a CNN template will be referred to omitting the subscripts (e.g. $\hat{D}_{ij,k} \rightarrow \hat{D}$)

3.3 The Image and Noise Model and the Segmentation Approach

The main properties of the general image model considered in the experiments are as follows:

- (i) *the image consists of simple geometrical objects each having a homogeneous gray-scale level and they might be overlapping*
- (ii) *inhomogeneous background illumination*
- (iii) *additive Gaussian noise (zero mean, unknown variance)*
- (iv) *image blur (low-pass type image degradation)*

In general case properties (i) and (ii) make it difficult to define a globally optimal threshold level to separate the objects from the background (Remark: this threshold does not even exist if the background and the object gray-scale levels overlap that might be the practical situation). Properties (iii) and (iv) clearly emphasize the necessity of noise filtering and enhancement. From the above model it is straightforward that an adaptive method is required to implement a robust segmentation. Since the primary goal is only a two-level segmentation (separating the objects from the background) it can be assumed that there is no need to make use of the global image properties, rather a locally adaptive strategy can meet the main requirements (at a considerably lower level of computational complexity).

Properties (i)-(iv) define a model for the original images used in the bubble-debris classification experiments. With the assumption that the mean gray-scale level of the objects (bubbles and debris particles) and the background is known and it is only slightly varying (using a priori information about (i) and (ii)) it is possible to create an artificial bubble-debris image. These synthesized images show both visually and in their global characteristics (e.g. histogram) a striking similarity to the originals (see Fig. 7 and Fig. 8) and can be used as the “ground truth” reference for quantitative evaluation of the performance of different segmentation algorithms.

4. CNN Based Diffusion Models: General Formulation

In this Section we define the CNN based diffusion models used in the experiments. The additive Gaussian assumption justifies the application of diffusion-type filters in noise reduction and image enhancement. First, the CNN models of the so-called constrained linear and nonlinear diffusion are derived from the PDE formulation (Subsection 4.1). The focus is on nonlinear models

⁵ The mask (also referred to as the “fixed-state map”) of a CNN layer is a binary image specifying whether at a certain location the corresponding CNN cell is in active (in case of +1) or inactive (in case of -1) state in the actual operation. Using the binary mask is one of the most simple ways to incorporate “nonlinear” spatial adaptivity to the CNN algorithms. If the mask is not specified, it is assumed that all CNN cells are in active state.

that are considered to be optimal in the proposed segmentation algorithm. The second part of this section introduces an algorithmic CNN approximation of a novel type (non-PDE related) nonlinear diffusion formulation. Due to its observed advantages for certain image and noise models this approach is also analyzed in detail (Subsection 4.2).

4.1 PDE based approach - constrained linear and nonlinear diffusion

A major breakthrough in the field of edge detection comes from Perona and Malik [24], who proposed anisotropic diffusion for adaptive smoothing to formulate the problem in terms of the nonlinear heat equation. Studying the classical linear theory ([20]-[23]), they recognize that the announced causality criterion does not uniquely force the choice of a Gaussian (a linear filtering strategy) to do the blurring. They modify and complete the criteria defined by Koenderink [22] and assert the following requirements for multi-scale "semantically meaningful" description of images: (i) *causality*, (ii) *immediate localization*: the region boundaries should be sharp at each resolution and no displacement of edges is allowed, (iii) *piecewise smoothing*: intraregion smoothing should occur preferentially over interregion smoothing. Perona and Malik propose the following *anisotropic diffusion* (nonlinear parabolic PDE) formulation to satisfy the above stated criteria:

$$\begin{aligned} \frac{d}{dt} I(\bar{x}, t) &= \text{div}[c(\bar{x}, t) \text{grad}(I(\bar{x}, t))] , & I(\bar{x}, t_0) &= I_0(\bar{x}) \\ c(\bar{x}, t) &= g(|\text{grad}(I(\bar{x}, t))|) \end{aligned} \quad (\varepsilon 5)$$

where $I(\bar{x}, t)$ is the image intensity ($I_0(\bar{x})$ is the original image), the vector \bar{x} represents the spatial coordinates, the time variable t can also be interpreted as the scaling parameter and $c(\bar{x}, t)$ is the thermal conductivity. As we can see the diffusion function is spatially varying: it depends on the absolute gradient intensity at a given location. Proposed functions for $g(\cdot)$ are (that enable to satisfy the *immediate localization* and *piecewise smoothing* criteria without sacrificing the *causality* assumption):

$$\begin{aligned} g_1 &= \exp\left(-|\text{grad}(I(\bar{x}, t))|^2 / K^2\right), \quad K > 0 \\ g_2 &= \left(1 + (|\text{grad}(I(\bar{x}, t))| / K)^{1+\alpha}\right)^{-1}, \quad \alpha > 0, \quad K > 0 \end{aligned} \quad (\varepsilon 6)$$

The causality and stability of the process is ensured by the strong maximum principle of the elliptic partial differential equations. Using adiabatic boundary conditions (i. e. setting the conduction coefficients to zero at the boundary of the image) all minima and maxima will belong to the original image. The edges will not be displaced while intraregion smoothing is preferred over interregion smoothing.

A feasible CNN model that implements the anisotropic diffusion given by (ε5)-(ε6) can be derived using a proper spatial discretization and simplification of the nonlinearity. (ε5) in two dimensions can be written as:

$$\begin{aligned}
\frac{d}{dt} I(\bar{x}, t) &= \text{div}[c(\bar{x}, t) \text{grad}(I(\bar{x}, t))] \\
&= \nabla^T [c(\bar{x}, t) \nabla I(\bar{x}, t)] \\
&= \frac{\partial}{\partial x} [c(\bar{x}, t) \frac{\partial}{\partial x} I(\bar{x}, t)] + \frac{\partial}{\partial y} [c(\bar{x}, t) \frac{\partial}{\partial y} I(\bar{x}, t)]
\end{aligned} \tag{E7}$$

With a finite difference approximation, the spatial discretization yields (Φ is the so-called flow function):

$$\begin{aligned}
\frac{d}{dt} I(\bar{x}, t) &\approx \frac{1}{\Delta x^2} \left[c(x + \frac{\Delta x}{2}, y, t) (I(x + \Delta x, y, t) - I(x, y, t)) \right. \\
&\quad \left. - c(x - \frac{\Delta x}{2}, y, t) (I(x, y, t) - I(x - \Delta x, y, t)) \right] \\
&\quad + \frac{1}{\Delta y^2} \left[c(x, y + \frac{\Delta y}{2}, t) (I(x, y + \Delta y, t) - I(x, y, t)) \right. \\
&\quad \left. - c(x, y - \frac{\Delta y}{2}, t) (I(x, y, t) - I(x, y - \Delta y, t)) \right] \\
&= \Phi_{i,j-1}(t) - \Phi_{i,j+1}(t) + \Phi_{i-1,j}(t) - \Phi_{i+1,j}(t) \\
&= \sum_{kl \in N_1} \Phi_{ij,kl}(\Delta I_{ij,kl})
\end{aligned} \tag{E8}$$

$$\text{where: } \Phi_{ij,kl}(\Delta I_{ij,kl}) = g(\Delta I_{ij,kl}) \Delta I_{ij,kl}, \quad \Delta I_{ij,kl} = I_{kl}(t) - I_{ij}(t)$$

where $I_{ij}(t) = I(i\Delta x, j\Delta y, t)$, and N_1 denotes the nearest neighborhood ($r = 1$) of $I_{ij}(t)$ with indices $\{(i,j-1), (i,j+1), (i-1,j), (i+1,j)\}$. (E8) is associated with the following CNN cell equation operating in the linear region of the piecewise-linear output characteristic (this ensures $v_{xij}(t) = v_{yij}(t)$):

$$\begin{aligned}
C \frac{d}{dt} v_{xij}(t) &= -R^{-1} v_{xij}(t) + a_0 v_{yij}(t) + \sum_{kl \in N_r} \hat{D}_{ij,kl}(\Delta v_{xx}) + I_{ij} \\
v_{yij}(t) &= f(v_{xij}(t)) \quad , \quad \Delta v_{xx} = v_{xkl}(t) - v_{xij}(t)
\end{aligned} \tag{E9}$$

Setting $R = 1$ and $C = 1$ the corresponding nonlinear template of a CNN solution can be formulated in the form of (**ANISOD1**: $X(O)$ - ORIGINAL IMAGE, B_c - ZF^6):

$$a_0 = 1, \quad \hat{D} = \begin{bmatrix} 0 & \Phi & 0 \\ \Phi & 0 & \Phi \\ 0 & \Phi & 0 \end{bmatrix}, \quad \Phi = g \Delta v_{xx}, \quad g = \begin{cases} 1 - |\Delta v_{xx}| / 2K & \text{if } |\Delta v_{xx}| < 2K \\ 0 & \text{otherwise} \end{cases}, \quad I_{ij} = 0 \tag{E10}$$

Observe that although the smoothing is adaptive, thus space variant, the nonlinear template governing the CNN array is space invariant, i.e. identical for each processing cell. To obtain better isotropy the flow can be calculated between diagonally neighboring pixels resulting in an eight-way

⁶ When a CNN template is specified in the text the corresponding symbolic name, the initial conditions and the boundary condition is given in parentheses (see also Section 3, Table 2 and Table 3). ZF stands for zero-flux (adiabatic, "mirror") boundary condition.

connected network (setting $\Delta d = \sqrt{2}$, $\Delta x = \Delta y = 1$). This extension implies completing the CNN template with diagonal terms. The spatial discretization unavoidably introduces some error but in all other respects the PDE and ODE systems are qualitatively identical. In this very specific case it can be proved [39] that whatever the choice of the approximation of the gradient is, the ODEs governing the CNN array still satisfy the strong maximum (minimum) principle, provided that $g(\cdot)$ is positive and bounded ($0 \leq g(\cdot) \leq 1$), and furthermore if a zero-flux boundary condition is satisfied. This makes it possible to simplify the highly nonlinear functions proposed for $g(\cdot)$ and use a characteristic (piecewise-linear radial basis function) that is easy to implement in VLSI (Fig. 1(b)).

Some concerns were raised by other researchers on the uniqueness of the (transient) solution and a “pre-diffusion” (regularization) strategy is proposed to improve the original model (e.g. [28]). In [39] we investigated several approaches (based on studies [24]-[29]). Since in a practical situation the image acquisition (optical system with a limited bandwidth, sampling and quantization) process implicitly contains the required blurring here we will not include it explicitly into the nonlinear diffusion models. From [39] only a particular one will be recalled here, due to its favorable VLSI implementation (stability and robustness) properties and flexible framework to incorporate further spatial adaptivity to this operator. This approach is a generalization of Nordström’s model ([25], variational regularization for global edge detection), named *constrained anisotropic diffusion*:

$$\frac{d}{dt}I(\bar{x},t) - \text{div}[c(\bar{x},t) \text{grad}(I(\bar{x},t))] = \beta(I(\bar{x},t_0)) - I(\bar{x},t), \quad I(\bar{x},t_0) = I_0(\bar{x}) \quad (\epsilon 11)$$

This formulation makes it possible to force the output to remain close to a pre-defined local morphological constraint or any image which is calculated by $\beta(\cdot)$. If $\beta(\cdot)$ is chosen properly this generalization will not violate the causality assumption of the multi-scale image description, but gives a flexible practical framework where different filtering strategies can be efficiently connected to the anisotropic diffusion. For example, one can define $\beta(\cdot)$ as the weighted average in a given neighborhood that can be achieved by convoluting a Gaussian with the original (initial) signal ($\beta(I(\bar{x},t_0)) = G_\sigma * I(\bar{x},t_0)$). A possible CNN template reflecting these ideas is the following (**ANISOD2**: U - ORIGINAL IMAGE, $X(O)$ - ORIGINAL IMAGE, B_c - ZF):

$$a_0 = 0, \quad \hat{D} = \begin{bmatrix} 0 & \Phi & 0 \\ \Phi & 0 & \Phi \\ 0 & \Phi & 0 \end{bmatrix}, \quad \Phi = g \Delta v_{xx}, \quad g = \begin{cases} 1 - |\Delta v_{xx}| / 2K & \text{if } |\Delta v_{xx}| < 2K \\ 0 & \text{otherwise} \end{cases}, \quad (\epsilon 12)$$

$$B = \begin{bmatrix} 0.05 & 0.15 & 0.05 \\ 0.15 & 0.20 & 0.15 \\ 0.05 & 0.15 & 0.05 \end{bmatrix}, \quad I_{ij} = 0$$

where B is defined as a simple low-pass filter template. Note that the contribution added by the B term can also be interpreted as a pre-calculated bias map of the anisotropic template ($I_{ij} = \beta(v_{u_{ij},k_l})$).

Therefore, when realized in two steps the implementation complexity of ANISOD2 is the same as that of ANISOD1 (but it is an advantage when the unity self-feedback is not used).

If the diffusion coefficients are not spatially varying and $\beta(\cdot)$ is defined as a simple low-pass filter from (E5) we obtain a constrained linear diffusion. This diffusion scheme is thoroughly discussed in [10] (where it is addressed to as “robust diffusion”). See CDIFFUS in Table 3.

4.2 Non-PDE based approach - a novel-type nonlinear diffusion formulation

Contrary to previous models that were derived from a PDE formulation, recently a novel-type nonlinear diffusion formulation [30] was proposed that is a non-PDE related approach. This model shows a superior performance compared to the PDE-related schemes in filtering and signal reconstruction when the signal is piecewise-linear and the signal-to-noise ratio is very low. Through the rest of this section this method will be presented and analyzed. It is shown that an algorithmic CNN approximation is possible though even the simplified solution has higher VLSI complexity than the previous models.

The form assumed for the diffusion equation is as follows ($I = I(t)$):

$$\frac{d}{dt}I_{ij} = J_w(I_{i,j-1} - I_{i,j}) + J_n(I_{i-1,j} - I_{i,j}) + J_e(I_{i,j+1} - I_{i,j}) + J_s(I_{i+1,j} - I_{i,j}) \quad (\text{E13})$$

here the diffusion connection weights are functions of the region of the image within the window N_r as shown in Fig. 2 (N_r is a union of four subwindows $N_{r,e}$, $N_{r,n}$, $N_{r,w}$, and $N_{r,s}$). Let us define the local variances and the total variance:

$$\begin{aligned} V_e &= \sum_{kl \in N_{r,e}} (I_{k,l} - I_{i,j})^2 \\ V_n &= \sum_{kl \in N_{r,n}} (I_{k,l} - I_{i,j})^2 \\ V_w &= \sum_{kl \in N_{r,w}} (I_{k,l} - I_{i,j})^2 \\ V_s &= \sum_{kl \in N_{r,s}} (I_{k,l} - I_{i,j})^2 \\ V_t &= V_e + V_n + V_w + V_s \end{aligned} \quad (\text{E14})$$

The diffusion coefficients can now be defined:

$$\begin{aligned}
J_e &= \frac{V_w}{V_t} \left(1 - e^{-\frac{V_t}{\lambda}}\right) + 0.25 e^{-\frac{V_t}{\lambda}} \\
J_n &= \frac{V_s}{V_t} \left(1 - e^{-\frac{V_t}{\lambda}}\right) + 0.25 e^{-\frac{V_t}{\lambda}} \\
J_w &= \frac{V_e}{V_t} \left(1 - e^{-\frac{V_t}{\lambda}}\right) + 0.25 e^{-\frac{V_t}{\lambda}} \\
J_s &= \frac{V_n}{V_t} \left(1 - e^{-\frac{V_t}{\lambda}}\right) + 0.25 e^{-\frac{V_t}{\lambda}}
\end{aligned} \tag{E15}$$

If the noise free image were approximately constant near a given pixel, one would want the connection coefficients to all be 0.25 (defining a linear diffusion). In this case it is possible for one of the local variances to have a value comparable to the total variance and simply defining the diffusion coefficients as the quotient of the local and total variance would lead to a model that is extremely noise sensitive. By introducing the ‘‘variance scale factor’’ λ in (E15) as in the homotopy methods this problem can be avoided. Furthermore, it has been found that performance is relatively insensitive to the value of λ over a moderate range.

(E13) can be rewritten into the following CNN cell equation operating in the linear region of the piecewise-linear output characteristics:

$$\begin{aligned}
C \frac{d}{dt} v_{x_{ij}}(t) &= -R^{-1} v_{x_{ij}}(t) + \sum_{kl \in N_r} A_{ij,kl}^{ij} v_{y_{kl}} + I_{ij} \\
v_{y_{ij}}(t) &= f(v_{x_{ij}}(t))
\end{aligned} \tag{E16}$$

Setting $R = 1$ and $C = 1$ the corresponding nonlinear template of a CNN solution can be formulated in the form of (ANISOD3: $X(O)$ - ORIGINAL IMAGE, B_c - ZF):

$$A^{ij} = \begin{bmatrix} 0 & J_n & 0 \\ J_w & 0 & J_e \\ 0 & J_s & 0 \end{bmatrix}, \quad I_{ij} = 0 \tag{E17}$$

where A^{ij} is a space-variant linear feedback template.

The nonlinear diffusion model defined by (E13)-(E15) does not have an associated PDE limit model with similar properties (an analysis can be performed based on [7]). Using some simplifications it is possible to derive an iterative CNN approximation that implements (E16)-(E17) at a VLSI complexity level close to the PDE related models based on (E9)-(E10), (E12) and still keeps some favorable properties of the original model. The flowchart of this algorithm (referred to as NLDIFF in later sections) with the prototype templates can be seen in Fig. 3.

Two simplifications seems to be necessary in diffusion coefficient calculation. First, the local variance estimation (E14) should be performed by employing the absolute value function in the nearest neighborhood. These estimates can be calculated by simple nonlinear templates (see VAR_E,

...VAR_S in Fig. 3) and stored in LAMs ($L_1..L_4$). The second problem arises from the weight normalization (ϵ_{15}), since it can not be assumed that division is implemented in local arithmetical units at the cell level. In this specific case normalization can be approximated with an iterative solution that forces the sum of the weight values to stay close to the unity (necessary condition for complete stability) without significantly altering the relative ratio of these values. Fig. 3 illustrates this solution. First, the sum of the LAM values ($L_1+L_2+L_3+L_4$) is compared to the lower threshold value ($\vartheta_L \approx 0.1$). If this value is not reached (small variance in all subwindows predicting a homogeneous region) then all LAMs ($L_1..L_4$) are set to 0.25 and the model reverts back to the linear diffusion. Otherwise, the variance estimates are scaled (SCALE: a quanta for each weight is calculated) and 4 additional LAMs (to be used in weight estimation) are set to zero ($L_5..L_8$). Then, in each iteration step the value of these LAMs is incremented by the corresponding quanta (e.g. $L_5=L_5+L_1$) and the sum ($L_5+L_8+L_5+L_8$) is compared to the upper threshold value ($\vartheta_H \approx 0.9$). When the upper threshold is exceeded it can be assumed that the sum of the weights is close to the unity within the required precision. In the last step these values are transferred to the initial LAMs ($L_1..L_4$) that are used to control the diffusion process.

Shifting the weight values to the proper locations (SHIFT) now the diffusion process can be ran controlled from the LAMs ($L_1..L_4$) for some time T. This should be repeated in a cycle (n) always updating the weight values before the diffusion is performed. Since the output of this nonlinear diffusion model converges to a nontrivial solution timing (nT) is not a critical issue (only a lower limit value should be given). It is not likely that any further simplification would be possible (based on the hardware considered in this analysis). *The necessity of space-variant programming is due to the same property of the model that makes impossible the derivation of an associated PDE: the weight values do not correspond to the subregions where the diffusion is performed.*

The described approximation requires 4 LAMs ($L_1..L_4$) and it is assumed that 4 of the special purpose LAMs ($U_1, U_2, X, Y, I, M \rightarrow L_5..L_8$) can also be used as buffers in the algorithm. Though this solution does not need a complex nonlinear CNN template, it operates with space-variant linear templates (the diffusion is controlled from local analog memories) that sets its VLSI complexity above the existing CNN Universal Chip implementations and it is more complex than the PDE related models (see a detailed analysis in Section 7). The mentioned advantage of the non-PDE related model comes from the observed property that in case of heavy noise corruption it is capable of reconstructing piecewise-constant signals better than templates derived from the PDE related models (see e.g. Fig. 4). Initial experiments also proved that this method does not necessarily brake down if the preliminary assumptions about the noise process (additive Gaussian) are violated (e.g. additive double-tailed exponential or multiplicative Gaussian). Intuitively, this robustness feature can be explained, since the estimation of the weight values in the non-PDE related model is based on the second order statistics of a subregion while it relates only to the local geometry in case of the PDE based models.

5. Analogic Algorithm Designed for Segmentation and Edge Detection

In this section the analogic CNN algorithm designed for image segmentation and edge detection is introduced. Though a powerful segmentation algorithm should for most image models incorporate some global aspects (see e.g. [53]-[54]) it will be shown that in the specific problem discussed here (object separation from an inhomogeneous background) incorporating only local aspects leads to a robust and reliable algorithm. Furthermore, it can be demonstrated how this CNN approach unifies some of the well-known schemes proposed for segmentation and edge detection.

The algorithm (see the flowchart in Fig. 5, outputs in Fig. 6) consists of three computational blocks: (i) linear or nonlinear diffusion based pre-filtering, (ii) local threshold estimation, (iii) locally adaptive segmentation and edge detection. Though the framework is “classical” there are two peculiar features of the approach. First, higher order statistics is also employed in the estimation process resulting in a better morphological description of the object boundary. Second, “locality” is not restricted to the nearest neighbors, since propagating-type CNN templates can be used in filtering, estimation and segmentation process. The following Subsections give a detailed analysis of the major computational blocks of the algorithm.

5.1 Diffusion Based Pre-filtering Strategies

The first block of the algorithm (**DIFF_FILT**) implements an image filter designed for noise reduction and edge enhancement. Let us denote this gray-scale mapping by:

$$\mathcal{F}: Y = \mathcal{D}(U) \tag{E18}$$

where \mathcal{D} stands for a linear or nonlinear diffusion-type filter, U is the input image and Y is the output image. The additive Gaussian noise model requires the application of a low-pass or a band-pass filter that justifies the use of a diffusion-type operator.

To keep the VLSI complexity at a lowest possible level a linear diffusion model is motivated. Templates corresponding to this approach are variants of **DIFFUS** or **CDIFFUS** specified in Table 3. It has been recognized [21] that running a linear diffusion for some time t corresponds to convolving the image with a Gaussian kernel characterized by σ (single parameter tuned low-pass filter). Consequently, CNN templates implementing a linear diffusion can also be tuned setting the transient length of the CNN. The frequency domain characteristics of the simplest diffusion operation (performed by **DIFFUS**) is going to vary from an “all-pass filter” to the ideal “DC-filter” as time goes to infinity (the output asymptotically converges toward the average of the image). **CDIFFUS** is a typical constrained diffusion (“robust diffusion” in [10]) template (the weights of the A and B term are equally set satisfying the stability conditions) where the B term is used to constrain the diffusion process, meaning that the characteristics of this filter converges toward a low-pass limit model determined predominantly by this term (the output will asymptotically converge toward the weighted average in a “large” neighborhood). Decreasing the weights of the A term and increasing the weights

in B term the **CONVOL** template will be obtained, that has the broadest low-pass characteristics when time goes to infinity (it calculates the weighted average in the nearest neighborhood) and can be considered as the upper limit model of the **CDIFFUS** family. (A thorough analysis of the CNN based linear filtering and some aspects of this discussion the reader is referred to [8]-[12].) Clearly, in a practical implementation the robustness and stability is traded against “neighborhood extension” in constrained linear diffusion templates balancing the sum of the weights of the A and B terms. Increasing the weights in A term (and decreasing the weights in B) allows “computation” (information spread) in a larger neighborhood but also increases the sensitivity of the model. In the linear scale-space of the constrained linear diffusion templates the additive Gaussian noise corruption is very efficiently removed but the object edges will also be blurred and shifted from their original locations.

At a cost of higher VLSI complexity, templates **ANISOD1** and **ANISOD2** derived from the PDE-related nonlinear models are capable of dealing with this problem since they possess an adaptive band-pass filtering characteristics (Section 4 and [39]). These templates are controlled by two parameters (K, t) the so-called noise level estimate and the transient length of the diffusion, respectively. If a priori information is available on the underlying noise model K can be fixed (e.g. directly relates to the std if the noise corruption is additive Gaussian), otherwise an algorithmic CNN solution can give a robust estimate as shown in [39]. As a consequence, these templates can also be tuned through the transient length parameter t as discussed earlier for the linear diffusion. The advantages of the nonlinear scale-space created by these templates is already analyzed in Section 4 (see also [39] for additional information). Evaluation of the output and robustness considerations strongly suggests that the nonlinear constrained diffusion (**ANISOD2**) is the optimal pre-filter in bubble-debris segmentation problems (see the input and output in Fig. 6(a)-(b), respectively) according to the image model enunciated in Section 3.

When the signal-to-noise level is very poor or the additive noise model is altered (e.g. additive double-tailed exponential or multiplicative Gaussian) then the algorithmic CNN approximation (**NLDIFF**) of the space-variant diffusion template (**ANISOD3**) derived from the non-PDE related approach is superior compared to the previous templates (tested for the image model outlined in Section 3). This also implies that using this model has advantages when the a priori information on the process is very limited. The significant drawback is the necessity of space variant programming in the implementation of this algorithm, though we strongly believe that similar powerful CNN based signal processing schemes may crucially affect the development trends in VLSI implementation of CNN Universal Chips (detailed analysis in Section 7).

Remark: it should be noted though that for some specific noise models, such as the additive double-tailed exponential (the so-called “impulse” noise model) efficient statistical filters has already been developed in the CNN framework based on the **MEDIAN** and **RO_CLASS** templates (see Table 3). In this case the optimal edge detection and segmentation strategy would also be

different (based on morphological and statistical operators) than the approach described and analyzed in this paper.

In [31]-[32] some alternative CNN-based nonlinear diffusion approximations are described that can also serve as the basis for noise filtering and image enhancement.

5.2 Local Threshold Estimation

The threshold estimation performed by the second computational block can be described by the following gray-scale to binary mapping:

$$7: I = \alpha \varepsilon_r(U) + \beta \text{var}_r(U) \quad (\varepsilon 19)$$

where $\varepsilon_r(U)$ and $\text{var}_r(U)$ are the images composed of the mean and variance estimates in local neighborhood N_r , respectively. U is the input image and I is the threshold estimate (bias map of the CNN for further processing⁷). ($\varepsilon 19$) defines a space-variant threshold level as a linear combination of the first and second order local statistics, and given that r (the neighborhood radius) is fixed it is specified by the parameters (α, β) . The (space-variant) threshold image should support an optimal separation of the objects from the background. The motivation behind the above formulation is to obtain a threshold estimate that better describes the object boundaries than an estimate based poorly on the average in a local neighborhood. By incorporating the term that calculates the variance statistics this goal can be achieved, since in general the variance is significantly higher at the boundaries than in homogeneous regions. Decreasing the mean estimate at these locations produces a better threshold estimate (bias map of the CNN) for further processing ($\alpha > 0, \beta < 0$). Determining the optimal values of (α, β) for different image models is still an ongoing research.

In a CNN implementation the mean can be estimated (**MEAN_EST**) through a constrained linear diffusion process **CDIFFUS**, while the approximation of the variance estimation (**MEAN_EST**) is completed using the absolute value function in a nonlinear CNN template **ABSVAL**. These two outputs are scaled (**SCALE** with central elements α, β) and added up (**ADD**) to create the bias map of the adaptive thresholding. The templates can be found in Table 2 and Table 3. Fig. 6(c), (d) and (e) demonstrate the mean estimate, variance estimate and the bias map ($\alpha = 0.5, \beta = -0.7$) for the input shown in Fig. 6(a).

5.3 Locally Adaptive Segmentation and Edge Detection

The proposed segmentation (third computational block) performs the following binary mapping:

$$S: Y' = S_t(Y, I + I_0) \quad (\varepsilon 20)$$

where Y is the diffusion filtered output, I is the space-variant threshold level calculated from local statistics, I_0 is a constant threshold level set according to a priori information and Y' is the binary

⁷ It is important to note that the bias map of the CNN is equal to the threshold estimate with an inverted sign.

output of the mapping. \mathcal{S}_r compares the filtering output to the threshold in local neighborhood N_r and determines the binary output.

In a CNN implementation of the adaptive segmentation (**ADSEGM**) a single-template operation (**ADTHRES**) gives the binary segmentation output using the output of the diffusion-type filtering and the space variant bias map containing the threshold estimate (Fig. 6(f)). It can be observed that the output of this computational step might contain a number of pixels and small patches that do not belong to any of the objects. If the a priori information on the minimum object size is at hand (e.g. minimal debris dimensions that are still important in bubble-debris experiments), a size classification block (**SIZECLASS**) can eliminate the objects being smaller than a specified size (**WPROP-RECALL**, Fig. 6(g)). The final edge map (Fig. 6(h)) can be obtained using binary morphology (**BW_MORPH** \rightarrow **EDGE**).

6. Analysis of the Algorithm Performance

From (ε18)- (ε20) the extended formulation of the segmentation is as follows:

$$\mathcal{S}: Y' = \mathcal{S}_r(\mathcal{D}(U), \alpha \varepsilon_r(U) + \beta \text{var}_r(U) + I_0) \quad (\varepsilon21)$$

If $\alpha = 1$, $\beta = 0$ and $I_0 = 0$ the main part of the algorithm boils down to a difference of Gaussians (*DoG*) type operator [20] since both the filtering and the mean estimation can be performed by a linear diffusion process:

$$\mathcal{D}_{oG}: Y' = \mathcal{S}_r(\mathcal{D}_{\sigma_1}(U) - \mathcal{D}_{\sigma_2}(U)) \quad (\varepsilon22)$$

When $\alpha = 0$, $\beta = 0$ and $I_0 \neq 0$ the method reverts back to a fixed thresholding approach that employs only a pre-filtering and a constant (globally optimal) threshold level set according to the a priori information on the underlying image model:

$$\mathcal{F}_{\alpha\beta}: Y' = \mathcal{S}_r(\mathcal{D}(U) - I_0) \quad (\varepsilon23)$$

Combination of both statistical estimates ($\alpha \neq 0$, $\beta \neq 0$) turned out to be an efficient and robust tool for different image and noise models.

The algorithm has been tested on number of real images containing both bubbles and debris particles (Fig. 6, Fig. 7, **APPENDIX B-C**). Note, that in the examples shown most significant details that can be observed in the original image are preserved in the last output containing the edge map. Artificial test images were also generated and tested corresponding to the general image model that have similar global histogram to the original images (Fig. 8, **APPENDIX A**). The segmentation algorithm has been tuned to minimize the output error in the nonlinear Hausdorff metric, since this approach has been explored in the companion paper designing a CNN algorithm for object classification.

The performance and robustness of the algorithm was examined altering the image and noise model parameters (shifting the object gray-scale levels, changing the background illumination,

violating the additive Gaussian noise assumption etc.). These experiments showed a better performance of the locally adaptive (nonlinear) method compared to a globally optimal fixed thresholding, edge detectors based on DoG-type operators and Canny's edge detector [23]. Further test are necessary for a reliable quantitative characterization.

Remark: since the analyzed image model assumes additive Gaussian noise the CNN-based segmentation problem discussed here could also be treated in the MRF-framework as it is discussed in [33].

7. Implementation in a CNN Universal Chip Environment

In this section we address important implementation issues, analyze the expected time performance of the algorithm in a CNN Universal Chip environment and show the VLSI implementation complexity of different solutions.

In Table 1 a possible ordering of the VLSI implementation complexity of different CNN hardwares are listed and numbered taking into the consideration the current trends in the analog VLSI technology. Level 0 marks the capabilities of the currently available VLSI CNN Universal Chips, levels 1-3 stand for the predicted hardware capabilities of the next generation to be fabricated in the near future (1-2 is an ongoing design). At present it is hard to judge the trend of the implementation above this level: 4-6a assume the extension of the neighborhood radius to $r = 2$, while levels 4-6b stand for the implementation of the space-variant programming. Comparing 4a and 4b the former seems to be a shorter and quicker step toward a more complex CNN Universal Chip (taking into account both the area complexity and the time needed to complete a new design). On the other hand, it should be noted that the algorithm development for real-time applications strongly motivates the insertion of at least 4 additional LAMs in a CNN core cell. When these LAMs are available 4b can be the next step (and would be certainly less complex in the area than 4a) resulting in a flexible usage of some of these LAMs as both data and program memories. Space-variant linear programming is required by number of recently published CNN solutions (e.g. [34] and see also the nonlinear non-PDE related diffusion model in this paper).

The minimal coding of all linear templates used in the analogic algorithms designed for image segmentation (not exceeding complexity level 3) is shown in Table 2. They are grouped and shaded with increasing gray-scale levels according to the type of the input-output mapping (binary input-output, gray-scale input - binary output, gray-scale input-output) they represent and their dynamic coupling (coupled, uncoupled).

Similarly to the linear templates, the minimal coding of all nonlinear templates used in the current study (not exceeding complexity level 3) is given in Table 3. All templates are gray-scale input-output operators and require zero flux (ZF) boundary condition setting. The nonlinear templates are grouped and shaded with increasing gray-scale levels according to the complexity of the nonlinear interactions and their dynamic coupling.

7.1 Stability and Robustness of Linear and Nonlinear Templates

All linear templates used in this paper are completely stable spatial operators. The template value specifications given in Table 2 are not necessarily the most robust solutions⁸. A number of these templates can be found in [47] in a globally optimized form. Algorithms and software packages (e.g. [48]-[49]) have been developed and are available for optimizing similar templates both for software simulators and for the currently available CNN Universal Chips.

Less is known about the nonlinear template family, though templates used in this study do not give rise to stability concerns. A comprehensive analysis of the stability of these types of templates is given in [39] including the base models of the nonlinear diffusion templates. Robustness tests are currently being examined.

7.2 VLSI Implementation- and Computational Complexity

The proposed and tested analogic CNN algorithm synthesized for image segmentation uses linear and nonlinear templates with nearest neighbor interactions and can be implemented on the complexity level 3 regarding the numbering of Table 1 if the PDE-related nonlinear diffusion templates (**ANISOD1-ANISOD2**) are used in pre-filtering. Applying a constrained linear diffusion both in mean estimation and pre-filtering (**CDIFFUS**), furthermore replacing the **ABSVL** template by **LAPLACE** (CNN approximation of the Laplacian) in “variance” estimation reduces the complexity level to 1 since the “bias map technique” can not be eliminated. Nevertheless, after some analysis one can recognize that in this very specific case in all templates where the bias map is used the linear B term is zero, therefore the bias map can be “forwarded” through the CNN input and the realization of the space-variant bias is not necessary. Consequently, the core of the algorithm that does not contain nonlinear cell interactions can be implemented on complexity level 0 where the current test experiments are being performed. The described “linear” version of the algorithm will produce slightly worst output compared to the simulations shown in this paper employing nonlinear cell interactions but could serve as a good prototype for the experiments on a real CNN hardware. (Remark: application of the non-PDE related model in pre-filtering sets the complexity level to 4b as discussed earlier).

The algorithm requires two additional LAMs that can be replaced by the two general analog input buffers U_1 and U_2 if the arithmetical unit is designed to access these memories. The fixed-state map technique and local logics are not used, consequently LLMs are not required either.

Table 4 shows the approximate timing of the templates at different processing levels of the entire analogic algorithm used in test examples. It is important to note that this timing is not critical, the behavior of most templates are more sensitive to the change of the template parameters (the spatial weights) than to the temporal uncertainty. The transient length is given in time constants and even in worst case it does not exceed 200τ . Since the time constant determined by the VLSI

⁸ The experiments have been completed based on the MATCNN [46], the analogic CNN toolbox for Matlab.

implementation of a core cell presently varies between 50-400 nsec ([15]-[19]), this implies 80 μ sec in worst case for the entire algorithm.

A very important topic to address is the resolution of the CNN chip. In spite of its dynamic coupled nature the algorithm can be implemented as a line by line video-flow processing since all propagating-type templates are local (or can be closely approximated by a local processing), i.e. they involve only a restricted cell neighborhood into the computation. Depending on the original image size a 16x512 or 32x256 array seems to be satisfactory to accomplish the front-end filtering and segmentation task discussed in this study.

8. Conclusions

We have proposed an algorithmic CNN approach based on (nonlinear) constrained diffusion and adaptive morphology for image segmentation and edge detection. Both PDE and non-PDE related diffusion models have been presented and thoroughly discussed. Special emphasis was given to the implementation issues in the environment of a CNN Universal Chip. The algorithm was tested on real and artificial images within the frame of the bubble-debris classification experiments and showed stable, robust performance with satisfactory output at a VLSI complexity level that is realizable in the near future.⁹

9. Acknowledgment

The helpful discussions and comments on various issues connected to this study are greatly acknowledged to Ricardo Carmona, Kenneth R. Crouse and Tibor Kozek.

This work has been supported by the grant of the ONR (No. N00014-89-J-1402) and the grant of the NSF (No. INT-9413186).

⁹ Comments on bubble-debris experiments: it should be noted that for the original images at hand the difference in performance between the proposed locally adaptive technique and a segmentation employing a fixed thresholding only has not been significant enough to justify the use of a complex algorithm. If the image quality obtained through the CCD sensor can be further improved then a proper noise filtering technique followed by fixed thresholding (at a global threshold level that is *a priori* known from preliminary image analysis) might be adequate to accomplish the front-end task outlined in this study. Though, it is certain that this type of implementation won't be able to deal with any kind of disturbances that are likely to happen within a system operating in a very noisy environment.

10. References

- [1] L. O. Chua and L. Yang, "Cellular Neural Networks: Theory", *IEEE Trans. on Circuits and Systems*, Vol. 35, pp. 1257-1272, Oct. 1988.
- [2] L. O. Chua and L. Yang, "Cellular Neural Networks: Applications", *IEEE Trans. on Circuits and Systems*, Vol. 35, pp. 1273-1290, Oct. 1988.
- [3] L. O. Chua, and T. Roska, "The CNN Paradigm", *IEEE Trans. on Circuits and Systems*, Vol. 40, pp.147-156, March 1993.
- [4] T. Roska and L. O. Chua, "The CNN Universal Machine: an Analogic Array Computer", *IEEE Trans. on Circuits and Systems*, Vol. 40, pp. 163-173, March 1993.
- [5] L. O. Chua and T. Roska, "Cellular Neural Networks: Foundations and Primer", *UCB/ERL M97*, Electronics Research Laboratory, University of California at Berkeley, 1997.
- [6] F. Werblin, T. Roska and L. O. Chua, "The Analogic Cellular Neural Network as a Bionic Eye", *International Journal of Circuit Theory and Applications*, Vol. 23, pp. 541-549, 1995.
- [7] J. P. Keener, "Propagation and its Failure in Coupled Systems of Discrete Excitable Cells", *SIAM J. Appl. Math.*, Vol. 47, 556-572, 1987.
- [8] B. E. Shi and L. O. Chua, "Resistive Grid Image Filtering: Input/Output Analysis via the CNN Framework", *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, Vol. 39, pp. 531-548, 1992.
- [9] B. E. Shi, *Spatio-temporal Image Filtering with Cellular Neural Networks*, PhD Thesis, University of California at Berkeley, 1994.
- [10] K. R. Crouse and L. O. Chua, "Methods for Image Processing in Cellular Neural Networks: A Tutorial", *IEEE Trans. on Circuits and Systems*, Vol. 42, No. 10, pp. 583-601, October 1995.
- [11] B. E. Shi, T. Roska and L. O. Chua, "Design of Linear Cellular Neural Networks for Motion Sensitive Filtering", *IEEE Trans. on Circuits and Systems*, Vol. 40, No. 5, pp. 320-331, October 1993.
- [12] K. R. Crouse, *Image Processing Techniques for Cellular Neural Network Hardware*, PhD Thesis, University of California at Berkeley, 1997.
- [13] B. E. Shi, "Order Statistic Filtering with Cellular Neural Networks", in *Proceedings of 3rd International Workshop on Cellular Neural Networks and Their Applications*, pp. 441-444, Rome, December 1994.
- [14] Á. Zarándy, A. Stoffels, T. Roska, and L. O. Chua, "Implementation of Binary and Gray-Scale Morphology on the CNN Universal Machine", *Memorandum UCB/ERL M96/19*, Electronics Research Laboratory, University of California at Berkeley, April 1996.
- [15] J. M. Cruz, L. O. Chua, and T. Roska, "A Fast, Complex and Efficient Test Implementation of the CNN Universal Machine", in *Proceedings of 3rd International Workshop on Cellular Neural Networks and Their Applications*, pp. 61-66, Rome, December 1994.
- [16] S. Espejo, R. Carmona, R. Domínguez-Castro, and A. Rodríguez-Vázquez, "CNN Universal Chip in CMOS Technology", *International Journal of Circuit Theory and Applications*, Vol. 24, pp. 93-111, 1996.
- [17] R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, and R. Carmona, "A 0.8 μm CMOS 2-D Programmable Mixed-Signal Focal-Plane Array Processor with On-Chip Binary Imaging and Instructions Storage", *IEEE J. Solid State Circuits*, pp. 103-1026, July 1997.
- [18] J. M. Cruz and L.O. Chua, "A 16x16 Cellular Neural Network Universal Chip: the First Complete Single-chip Dynamic Computer Array with Distributed Memory and with Gray-scale Input-output", *Journal Analog Integrated Circuits and Signal Processing*, in press, 1997.
- [19] Ari Paasio, Adam Dawidziuk, Kari Halonen and Veikko Porra, "Minimum Size 0.5 Micron CMOS Programmable 48 by 48 CNN Test Chip", in *Proceedings of the ECCTD '97*, pp. 154-156, Budapest, August 1997.

- [20] D. Marr and E. Hildreth, "Theory of Edge Detection", *Proc. Roy. Soc., London Ser. B*, pp. 187-217, 1980.
- [21] A. Witkin, "Scale-Space Filtering" in Int. Joint Conference Artificial Intelligence, Karlsruhe, West Germany, pp. 1019-1021, 1983.
- [22] J. Koenderink, "The Structure of Images", *Biolog. Cybernetics*, Vol. 50, pp. 363-370, 1984.
- [23] J. Canny, "A Computational Approach to Edge Detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, pp. 679-698, November 1986.
- [24] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 629-639, July 1990.
- [25] N. Nordström, "Biased Anisotropic Diffusion - A Unified Regularization and Diffusion Approach to Edge Detection", *Image Vision Computers*, Vol. 8, No 4, pp. 318-327, July 1990.
- [26] M. Nitzberg and T. Shiota, "Nonlinear Image Smoothing with Edge and Corner Enhancement", *Techn. Report 90-2*, Division of Applied Sciences, Harvard University, Cambridge, MA, 1990.
- [27] G. Gerig, O. Kübler, R. Kikinis, and F. A. Jolesz, "Nonlinear Anisotropic Filtering of MRI Data", *IEEE Transactions on Medical Imaging*, Vol. 11, No. 2, June, 1992.
- [28] F. Catté, P. L. Lions, J. M. Morel, and T. Coll, "Image Selective Smoothing and Edge Detection by Nonlinear Diffusion", *SIAM Journal on Numerical Analysis*, Vol. 29, No. 1, pp. 182-193, 1992.
- [29] L. Alvarez, P. L. Lions, and J. M. Morel, "Image Selective Smoothing and Edge Detection by Nonlinear Diffusion. II", *SIAM Journal on Numerical Analysis*, Vol. 29, No. 3, pp. 845-866, 1992.
- [30] A. Schultz, "A Nonlinear Diffusion method for Image Segmentation", Report, Radar Division, Naval Research Laboratory, 1993.
- [31] T. Roska and T. Szirányi, "Classes of Analogic Algorithms and Their Practical Use in Complex Image Processing Tasks", in *Proceedings of the NSIP'95*, pp. 767-770, Halkidiki, June 1995.
- [32] M. Csapodi and T. Roska, "Dynamic Analogic Algorithms for Complex Recognition Task - A First Step Towards the Bionic Eyeglass", *International Journal of Circuit Theory and Applications*, Vol. 24, No. 1, pp. 127-145, 1996.
- [33] T. Szirányi and J. Zerubia, "Markov Random Field Image Segmentation Using Cellular Neural Network", *IEEE Transactions on Circuits and Systems*, Vol. 44, pp. 86-69, January 1997.
- [34] P. Szolgay, I. Szatmári and K. László, "A Fast Fixed Point Learning Method to Implement Associative Memory on CNNs", *IEEE Transactions on Circuits and Systems*, Vol. 44, No. 4, pp. 362-366, April 1997.
- [35] Á. Zarándy, F. Werblin, T. Roska and L. O. Chua, "Novel Types of Analogic CNN Algorithms for Recognizing Bank-Notes", *Memorandum UCB/ERL M94*, Electronics Research Laboratory, University of California at Berkeley, April 1994.
- [36] Á. Zarándy, T. Roska, Gy. Liszka, J. Hegyesi, L. Kék and Cs. Rekeczky, "Design of Analogic CNN Algorithms for Mammogram Analysis", in *Proceedings of 3rd International Workshop on Cellular Neural Networks and Their Applications*, pp. 255-260, Rome, 1994.
- [37] Á. Zarándy, F. Werblin, T. Roska and L. O. Chua, "Spatial-logic Algorithms Using Basic Morphological Analogic CNN Operations", *International Journal of Circuit Theory and Applications*, Vol. 24, pp. 283-300, 1996.
- [38] Cs. Rekeczky, A. Ushida, and T. Roska, "Rotation Invariant Detection of Moving and Standing objects Using Analogic Cellular Neural Network Algorithms Based on Ring-Codes", *IEICE Transactions on Fundamentals and Information Sciences*, Vol. E78-A, No. 10, pp. 1316-1330, October 1995.
- [39] Cs. Rekeczky, T. Roska and A. Ushida, "CNN Based Self-adjusting Nonlinear Filters", *DNS-4-96, Technical Report*, Analogical and Neural Computing Systems Laboratory, Computer and Automation Institute of the Hungarian Academy of Sciences, April 1996.
- [40] Cs. Rekeczky, T. Roska, and A. Ushida, "CNN Based Self-adjusting Nonlinear Filters", in *Proceedings of 4th International Workshop on Cellular Neural Networks and Their Applications*, pp. 369-374, Seville, June 1996.

- [41] Cs. Rekeczky, T. Roska and A. Ushida, "CNN-based Difference-controlled Adaptive Nonlinear Image Filters", to appear in *International Journal of Circuit Theory and Applications*, 1998.
- [42] Cs. Rekeczky, Y. Nishio, A. Ushida, T. Roska, and Á. Tahy, "Image Segmentation by Soft Computing on the CNN Universal Machine", in *Proceedings of the 3rd NOLTA '96*, pp. 213-216, Kochi, October 1996.
- [43] Cs. Rekeczky, Á. Tahy, Z. Végh, and T. Roska, "Spatio-temporal Nonlinear Filtering and Endocardial Boundary Detection in Echocardiography", in *Proceedings of ECCTD '97*, pp. 677-672, Budapest, August 1997.
- [44] Cs. Rekeczky, Á. Tahy, Z. Végh, and T. Roska, "The Use of CNN Technology in Echocardiography: Spatio-temporal Nonlinear Filtering and Contour Detection", *DNS-7-1997, Technical Report*, Analogical and Neural Computing Systems Laboratory, Computer and Automation Institute of the Hungarian Academy of Sciences, May 1997.
- [45] Cs. Rekeczky, Á. Tahy, Z. Végh, and T. Roska, "Spatio-temporal Nonlinear Filtering and Endocardial Boundary Detection in Echocardiography", to appear in *International Journal of Circuit Theory and Applications*, 1998.
- [46] Cs. Rekeczky, "MATCNN - Analogic Simulation Toolbox for Matlab", Version 1.0, *DNS-11-1997, Technical Report*, Analogical and Neural Computing Systems Laboratory, Computer and Automation Institute of the Hungarian Academy of Sciences, September 1997.
- [47] T. Roska, L. Kék, Á. Zarándy, and L. Nemes, ed. "Analogic CNN Program Library", Version 7.0, Analogical and Neural Computing Laboratory, Computer and Automation Institute, Hungarian Academy of Sciences, 1997.
- [48] The CNN Workstation Toolkit, Version 6.0, MTA SZTAKI, Budapest, 1994
- [49] L. Nemes, L. O. Chua, and T. Roska, "LocRule User' Guide", Version 2.2, Electronics Research Laboratory, University of California at Berkeley, March 1997.
- [50] Á. Zarándy, T. Roska, F. Werblin, and L. O. Chua, "Intelligent Image Resolution Enhancement by the CNN Universal Machine and Its Relevance to TV Picture Enhancement", in *Proc. of 3rd NOLTA '95*, pp. 701-706, Las Vegas, December 1995.
- [51] T. Kozek, K. R. Crouse, T. Roska, and L. O. Chua, "Smart Image Scanning Algorithms for the CNN Universal Machine", in *Proc. of 3rd NOLTA '95*, pp. 707-712, Las Vegas, December 1995.
- [52] I. Szatmári, A. Schultz, Cs. Rekeczky, T. Roska, and L.O.Chua, "Bubble-debris Classification on CNN via Autowave Metric Implementation and Binary Morphology", *Memorandum UCB/ERL M97/97*, Electronics Research Laboratory, University of California at Berkeley, December 1997.
- [53] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation", Report No. UCB/CSD-97-940, University of California at Berkeley, May 1997.
- [54] J. Shi and J. Malik, "Motion Segmentation and Tracking Using Normalized Cuts", Report No. UCB/CSD-97-962, University of California at Berkeley, June 1997.

Figure Captions

Figure 1 Two fundamental VLSI building blocks of nonlinear interactions, (a) piecewise-linear sigmoid function, (b) piecewise-linear radial basis function.

Figure 2 The four symmetric subwindows surrounding each pixel used in variance estimation. The western (W) subwindow is shaded with gray and the pixels involved in the estimation process ($r=2$) are drawn with black. The figure also shows the first ($r=1$) and second ($r=2$) order neighborhood of the central pixel (i,j).

Figure 3 Flowchart of the analogic CNN algorithm estimating the non-PDE related diffusion model

Figure 4 Comparison of CNN based diffusion methods in filtering, (a) original image corrupted by additive Gaussian (b) output of the constrained nonlinear diffusion derived from PDE formulation, (c) output of the nonlinear diffusion derived from non-PDE approach.

Figure 5 Flowchart of the analogic algorithm designed for image segmentation.

Figure 6 Intermediate processing results of the analogic CNN algorithm designed for image segmentation, (a) original image, (b) output of the nonlinear diffusion, (c) mean estimate, (d) variance estimate, (e) bias map, (f) output of the adaptive segmentation, (g) output of the sorting process, (h) output of the edge detection.

Figure 7 Test image composed from original images containing bubbles and debris particles, (a) subwindow from the original image, (b) original debris particles, (c) composed image (the first 5 debris particles from figure (b) were inserted to the original bubble image), (d) image histogram of (c).

Figure 8 Test image synthesized based on the image model, (a) artificial image created according to the image model, (b) the “ground truth” segmentation for reference, (c) artificial image corrupted by Gaussian noise and blurred, (d) image histogram of (c).

Table Captions

Table 1 A possible ordering of the VLSI implementation complexity of the CNN Universal Machine core cell and local interactions. The complexity of different CNN hardware are listed and numbered taking into the consideration the current trends in the analog VLSI technology. Level **0** marks the capabilities of the currently available VLSI CNN Universal Chips (first order cells with nearest neighbor linear templates), levels **1-3** stand for the predicted hardware capabilities of the next generation to be fabricated in the near future (fixed-state map, bias map and the most simple nonlinear interactions - n_1 stands for the pwl sigmoid, n_2 for the pwl radial basis function, m marks the multiplication at the nonlinear interactions). Levels **4-6a** indicate the complexity of the implementation of templates with a larger neighborhood ($r = 2$) and levels **4-6b** the space-variant (linear) programming.

Table 2 Minimal coding of all linear templates used in segmentation algorithm. They are grouped and shaded with increasing gray-scale levels according to the type of the input-output mapping (binary input-output, gray-scale input - binary output, gray-scale input-output) they represent and their dynamic coupling (coupled, uncoupled). In boundary condition specification (B_c): X - don't care, ZF - zero flux, $[-1\ 1]$ - constant. *Binary input-output* templates: **THRES-EDGE** (uncoupled), **ADTHRES-RECALL** (coupled); *gray-scale input - binary output* template: **COMPARE** (uncoupled), *gray-scale input-output* templates: **SCALE-CONVOL** (uncoupled) and **DIFFUS-CDIFFUS** (coupled).

Table 3 Minimal coding of all nonlinear (DCN - difference-controlled nonlinear) templates used in the segmentation algorithm. Except for the **GRADIENT** all these templates are gray-scale input-output operators and require zero flux (ZF) boundary condition setting. The nonlinear templates are grouped and shaded with increasing gray-scale levels according to the complexity of the nonlinear interactions and their dynamic coupling. *Static difference-controlled* (uncoupled) templates: **ABSVAL-GRADIENT** (pwl radial basis function); *dynamic difference-controlled* (coupled) templates: **MEDIAN-RO_FILT** (pwl sigmoid-type), **ANISOD2-ANISOD1** (pwl radial basis and multiplication).

Table 4 Running time (transient length) specification for all templates used in the segmentation algorithm. The timing is given in time constant measure ($\tau = RC$). t_F stands for all uncoupled templates (a fixed transient length) and it is set to 5τ . t_{AR} and t_{LO} are the transient length of the arithmetic and logic operations on the cell level, respectively (1τ), while t_{RP} stands for the time necessary for reprogramming the cells from the central unit (approximately equal to the fixed transient length 5τ of a non-propagating type CNN template). The time needed for signal transfers between LAMs at the cell level is neglected. The table gives an approximate timing for all templates used in analogic subroutines and an estimated upper limit for all processing levels (note, that timing is not critical, though certain timing ratios are important). Timing of the alternative template solutions are specified in parenthesis. It turns out that even in worst case the algorithm completes the detection for a single frame in less than 200τ , i.e. less than $80\ \mu\text{sec}$ in worst case taking into the consideration the characteristics of the presently available VLSI implementation of a CNN core cell ($\tau = 100\text{-}200\ \text{nsec}$).

Figures

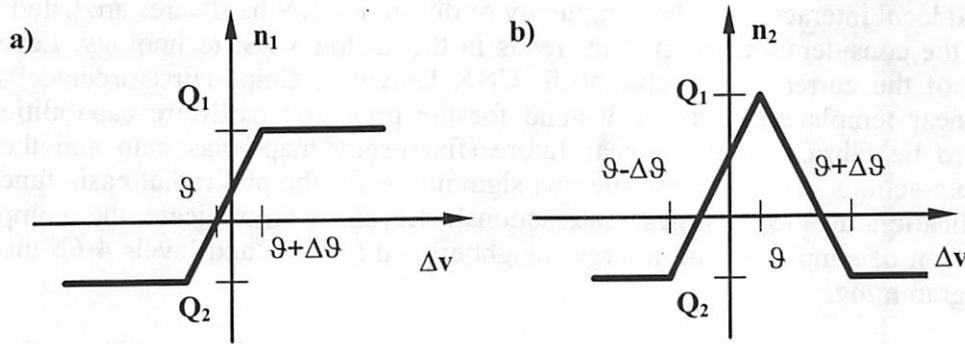


Figure 1 Two fundamental VLSI building blocks of nonlinear interactions, (a) piecewise-linear sigmoid function, (b) piecewise-linear radial basis function.

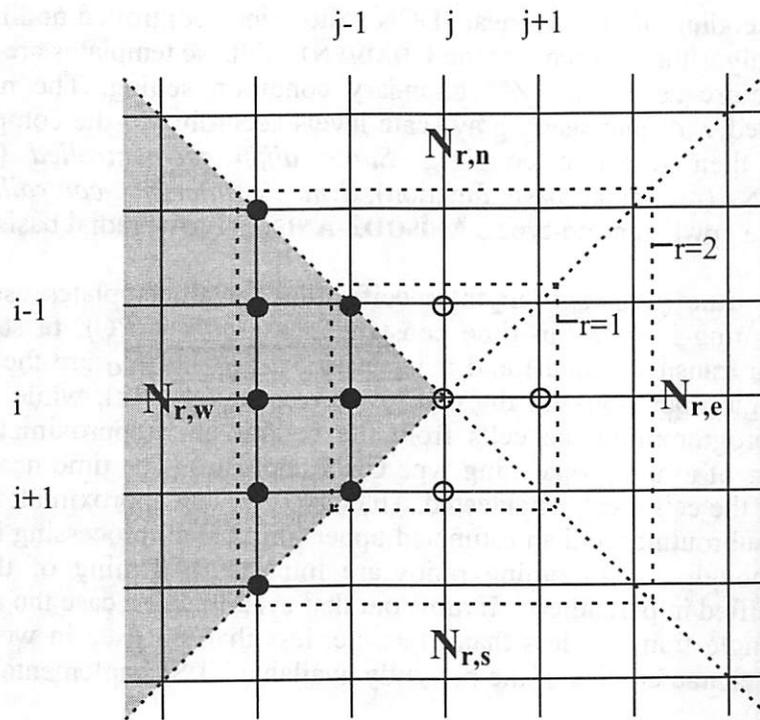


Figure 2 The four symmetric subwindows surrounding each pixel used in variance estimation. The western ($N_{r,w}$) subwindow is shaded with gray and the pixels involved in the estimation process ($r=2$) are drawn with black. The figure also shows the first ($r=1$) and second ($r=2$) order neighborhood of the central pixel (i,j).

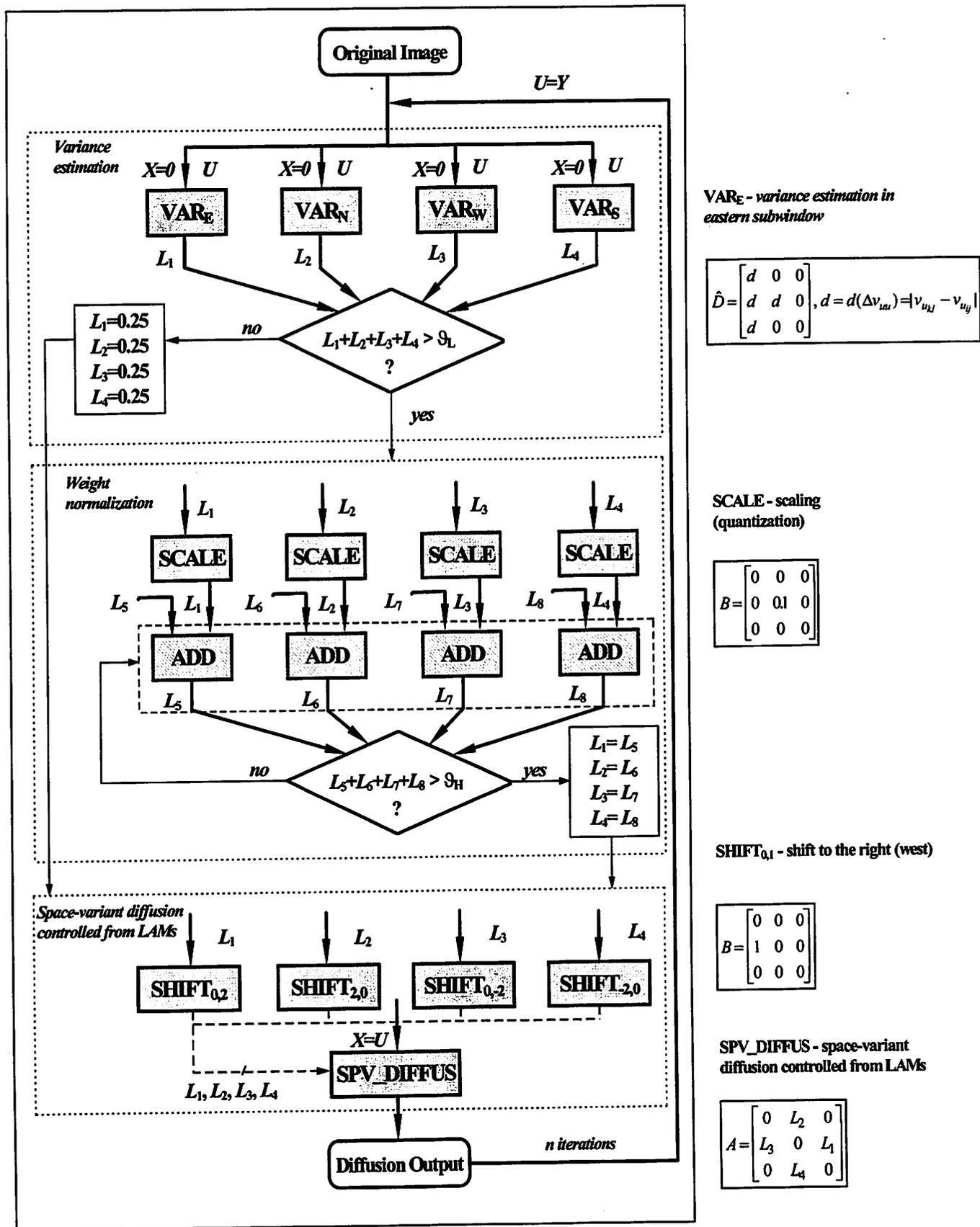


Figure 3 Flowchart of the analogic CNN algorithm estimating the non-PDE related diffusion model

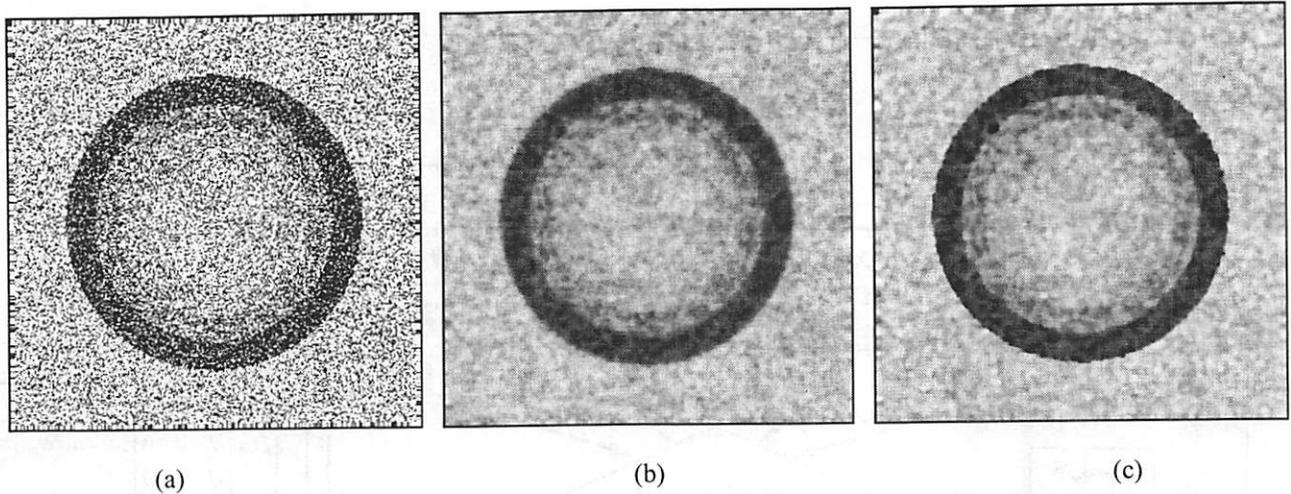


Figure 4 Comparison of CNN based diffusion methods in filtering, (a) original image corrupted by additive Gaussian noise, (b) output of the constrained nonlinear diffusion derived from PDE formulation, (c) output of the nonlinear diffusion derived from non-PDE approach.

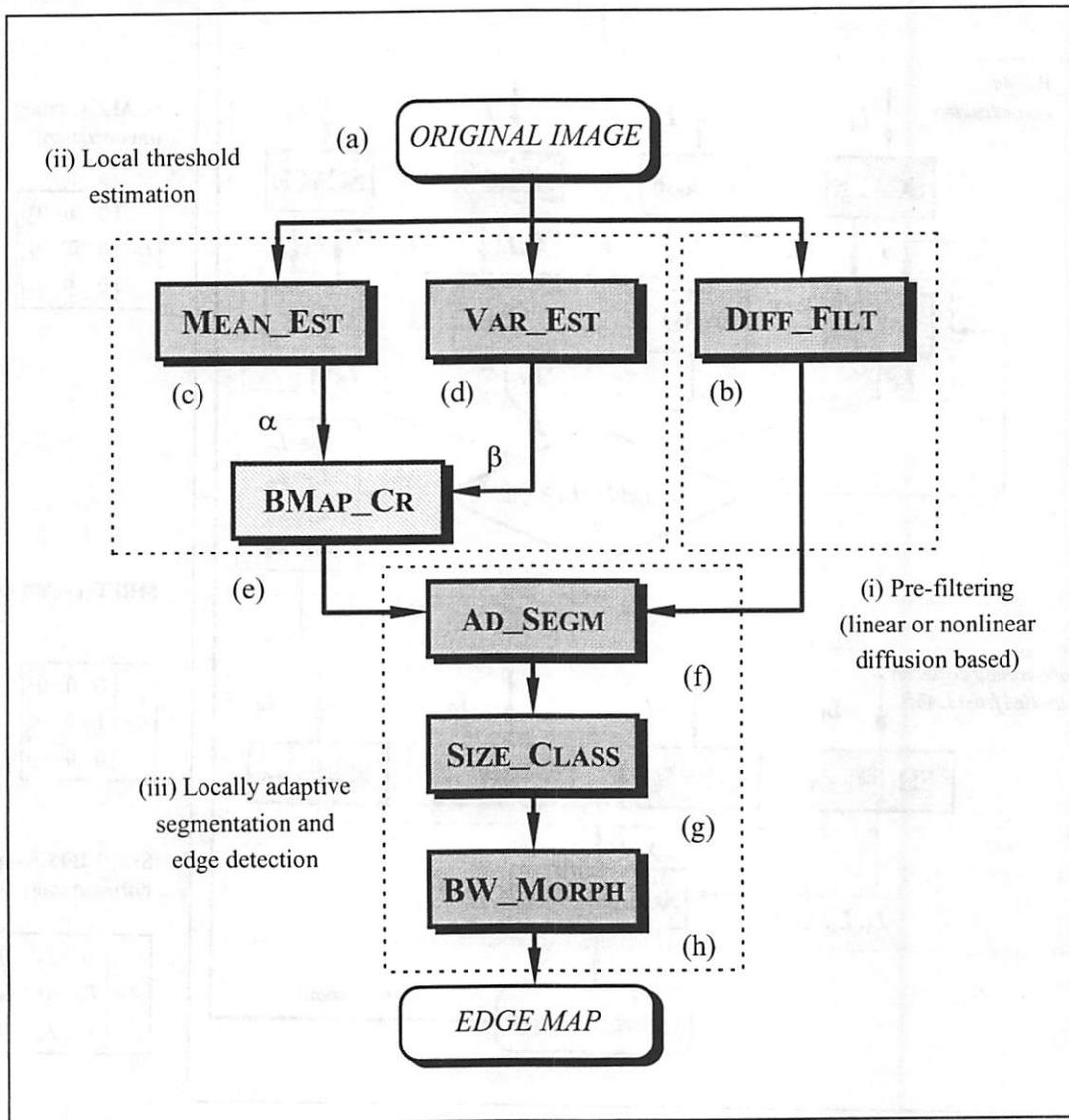


Figure 5 Flowchart of the analogic algorithm designed for image segmentation.

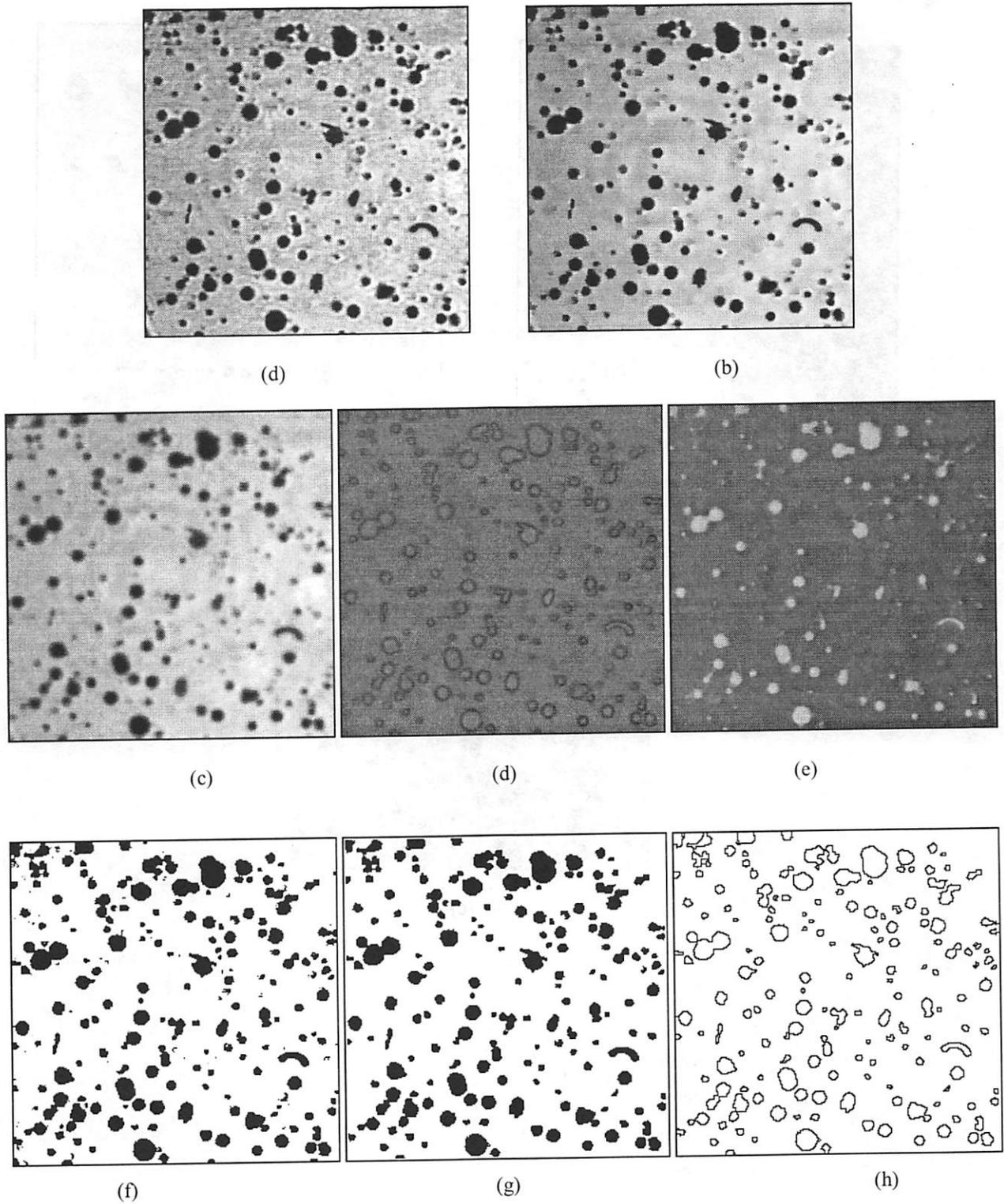
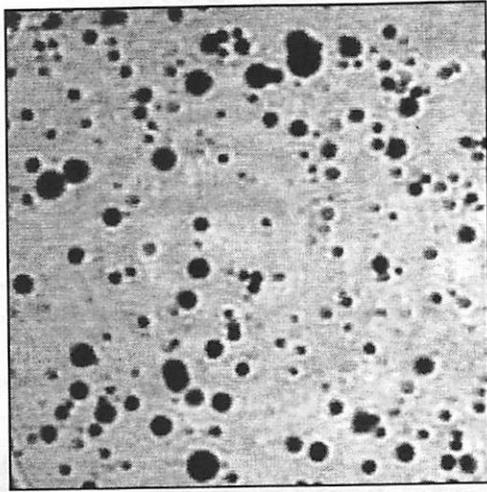
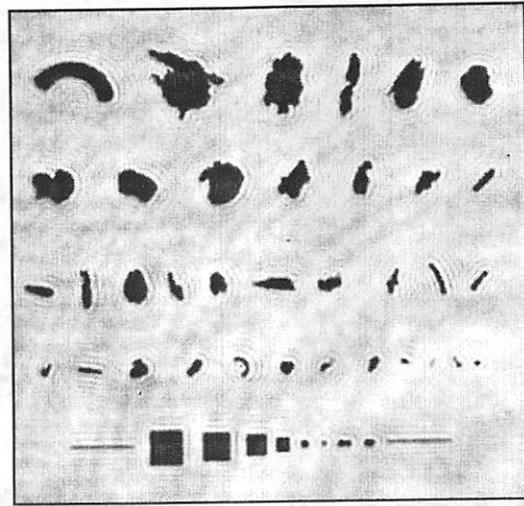


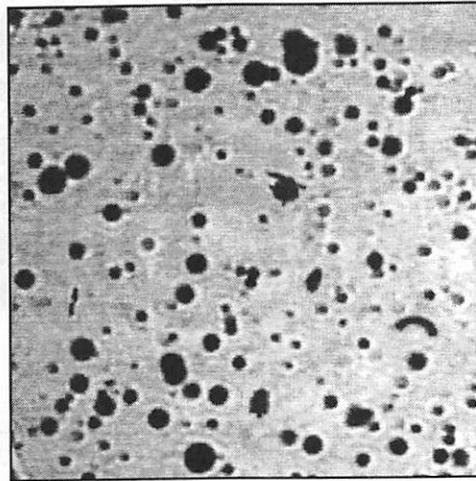
Figure 6 Intermediate processing results of the analog CNN algorithm designed for image segmentation, (a) original image, (b) output of the nonlinear diffusion, (c) mean estimate, (d) variance estimate, (e) bias map, (f) output of the adaptive segmentation, (g) output of the sorting process, (h) output of the edge detection.



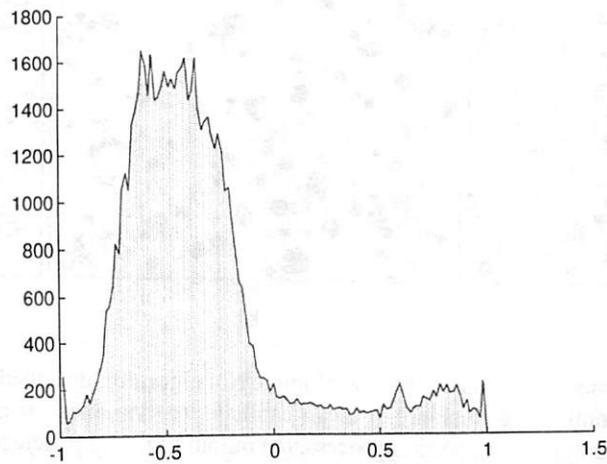
(a)



(b)

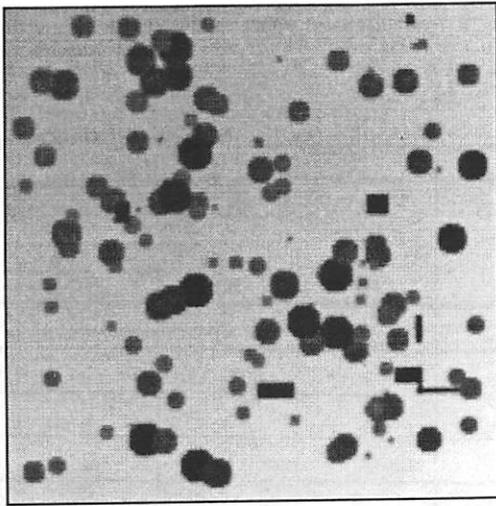


(c)

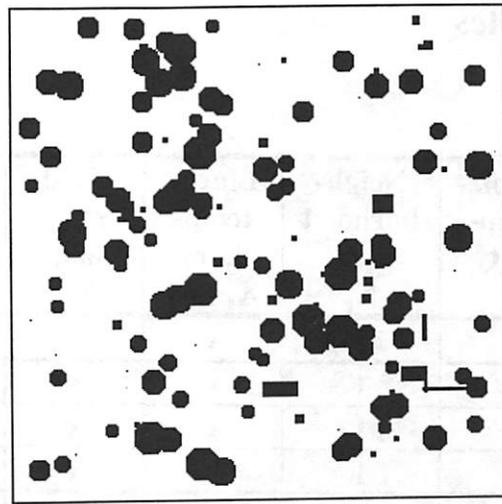


(d)

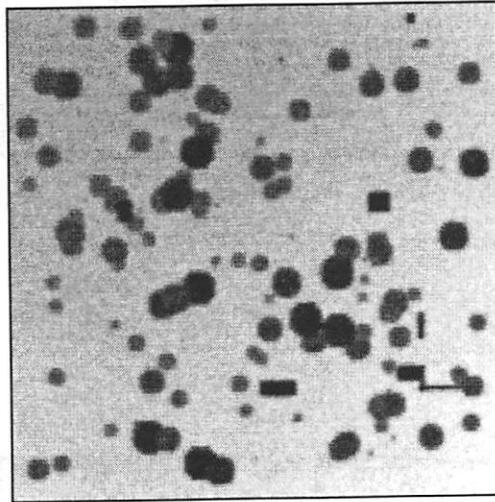
Figure 7 Test image composed from original images containing bubbles and debris particles, (a) subwindow from the original image, (b) original debris particles, (c) composed image (the first 5 debris particles from figure (b) were inserted into the original bubble image), (d) image histogram of (c).



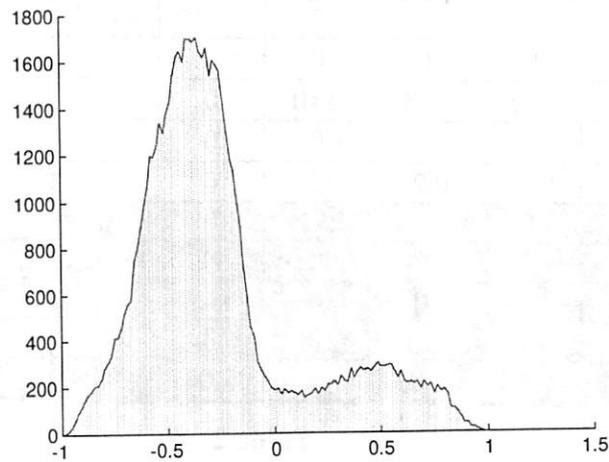
(a)



(b)



(c)



(d)

Figure 8 Test image synthesised based on the image model, (a) artificial image created according to the image model, (b) the “ground truth” segmentation, (c) artificial image corrupted by Gaussian noise and blurred, (d) image histogram of (c).

Tables

| Impl. complexity | Neighborhood N_r | Linear templates A, B, I | Fixed-state map M_{ij} | Bias map I_{ij} | Nonlinear templates | | | Space-variant templ. |
|------------------|-----------------------|-----------------------------|-----------------------------|----------------------|---------------------|-------|-----|----------------------|
| | | | | | n_1 | n_2 | m | |
| 0 | 1 | X | - | - | - | - | - | - |
| 1 | 1 | X | X | X | - | - | - | - |
| 2 | 1 | X | X | X | X | X | - | - |
| 3 | 1 | X | X | X | X | X | X | - |
| 4a | 2 | X | X | X | - | - | - | - |
| 5a | 2 | X | X | X | X | X | - | - |
| 6a | 2 | X | X | X | X | X | X | - |
| 4b | 1 | X | X | X | - | - | - | X |
| 5b | 1 | X | X | X | X | X | - | X |
| 6b | 1 | X | X | X | X | X | X | X |

Table 1

| Template (time spec) | Feedback (A) | | | Control (B) | | | Current I | Bcond B_c |
|-------------------------|--------------|-------|-------|-------------|-------|-------|--------------|----------------|
| | a_0 | a_1 | a_2 | b_0 | b_1 | b_2 | | |
| THRES (t_F) | 2 | 0 | 0 | 0 | 0 | 0 | -0.5 | X |
| EDGE (t_F) | 2 | 0 | 0 | 2 | -0.25 | -0.25 | -1.5 | -1 |
| ADTHRES (t_A) | 2 | 0.2 | 0.1 | 0 | 0 | 0 | -0.5 | ZF |
| BPROP (t_B) | 3 | 0.2 | 0.3 | -2 (0) | 0 | 0 | 3.75 | -1 |
| WPROP (t_W) | 3 | 0.2 | 0.3 | 0 | 0 | 0 | -3.75 | 1 |
| RECALL (t_R) | 2 | 0.25 | 0.25 | 2 | 0 | 0 | 0.75 | -1 |
| SCALE (t_F) | 0 | 0 | 0 | 0.1 | 0 | 0 | 0 | X |
| LAPLACE (t_F) | 0 | 0 | 0 | 4 | -1 | 0 | 0 | ZF |
| CONVOL (t_F) | 0 | 0 | 0 | 0 | 0.15 | 0.1 | 0 | ZF |
| DIFFUS (t_D) | 0 | 0.15 | 0.1 | 0 | 0 | 0 | 0 | ZF |
| CDIFFUS (t_{CD}) | 0 | 0.08 | 0.05 | 0 | 0.07 | 0.05 | 0 | ZF |

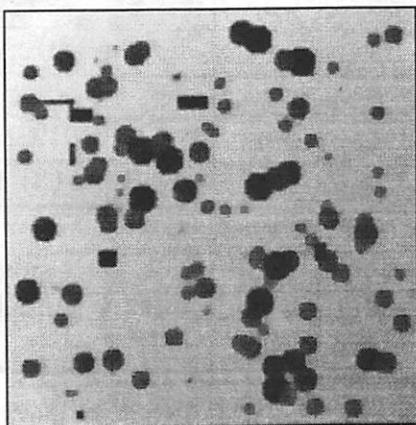
Table 2

| Template (time spec) | Operator | Multiplic. | Nlin function | Amplitude level | Thres- hold level | Weighting | Self- feed- back | Cell cur- rent |
|-------------------------|------------|------------|------------------|--------------------|-----------------------------------|--------------------------------|------------------------|----------------------|
| | Δv | m | n_i | [Q_1 Q_2] | [ϑ $\Delta\vartheta$] | [η_0 η_1 η_2] | a_0 | I |
| ABSVAL (t_F) | 2 | 0 | 2 | [0 1] | [0 1] | [1 0 0] | 0 | 0 |
| GRADIENT (t_F) | 2 | 0 | 2 | [0 1] | [0 2] | [0 1 0.7] | 2 | -7 |
| MEDIAN (t_F) | 4 | 0 | 1 | [0.5 -0.5] | [0 .05] | [0 (1) 1 1] | 1 | 0 |
| RO_FILT (t_F) | 4 | 0 | 1 | [0.5 -0.5] | [0 .05] | [0 (1) 1 1] | 1 | -4÷4 |
| ANISOD2 (t_{AN}) | 3 (1) | 1 | 2 | [1 0] | [0 .15] | [0 1 0.5] | 0 | 0 |
| ANISOD1 (t_{AN}) | 3 (1) | 1 | 2 | [1 0] | [0 .15] | [0 1 0.5] | 1 | 0 |

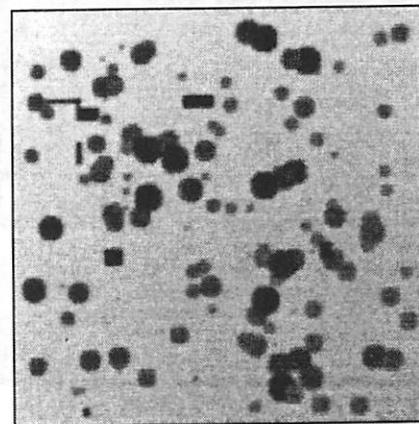
Table 3

| Process. Level | t_F | t_A | t_B | t_W | t_R | t_D | t_{CD} | t_{AN} | t_{LO} | t_{AR} | t_{RP} | SUM |
|-------------------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|-----|
| DIFF_FILT | (5) | - | - | - | - | (10) | (10) | 15 | - | - | 5 | 20 |
| MEAN_EST | (5) | - | - | - | - | (15) | 25 | - | - | - | 5 | 30 |
| VAR_EST | 5 | - | - | - | - | - | - | - | - | - | 5 | 10 |
| BMAP_CR | 10 | - | - | - | - | - | - | - | - | 1 | 10 | 21 |
| AD_SEGM | - | 15 | - | - | - | - | - | - | - | - | 5 | 20 |
| SIZE_CLASS | - | - | (5) | 15 | 25 | - | - | - | - | - | 10 | 50 |
| BW_MORPH | 5 | - | - | - | - | - | - | - | - | - | 5 | 10 |
| CNN ALG | | | | | | | | | | | | 161 |

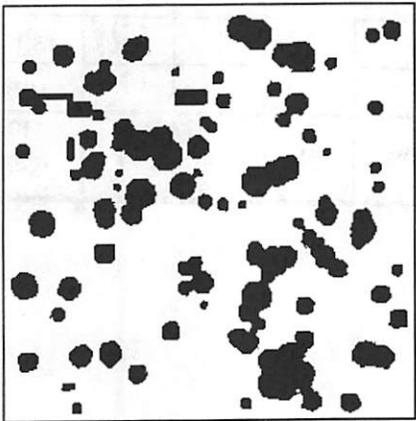
Table 4



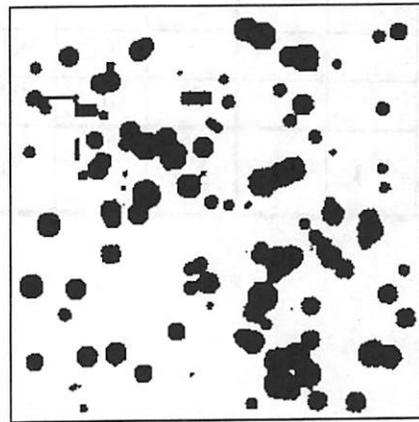
(b) (nonlinear) diffusion output



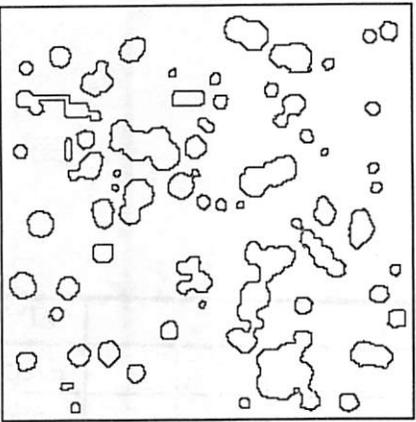
(a) artificial image



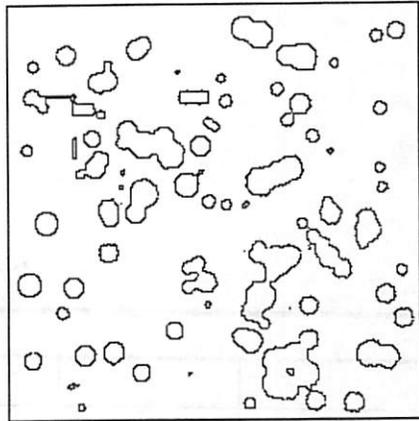
(d) output of adaptive segmentation



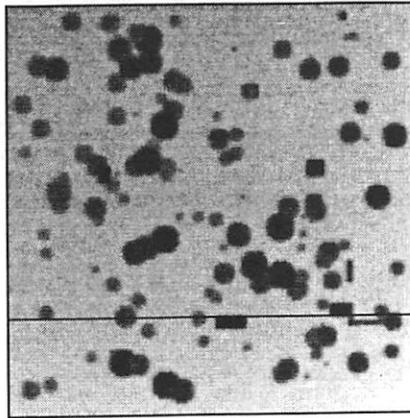
(c) output of fixed thresholding



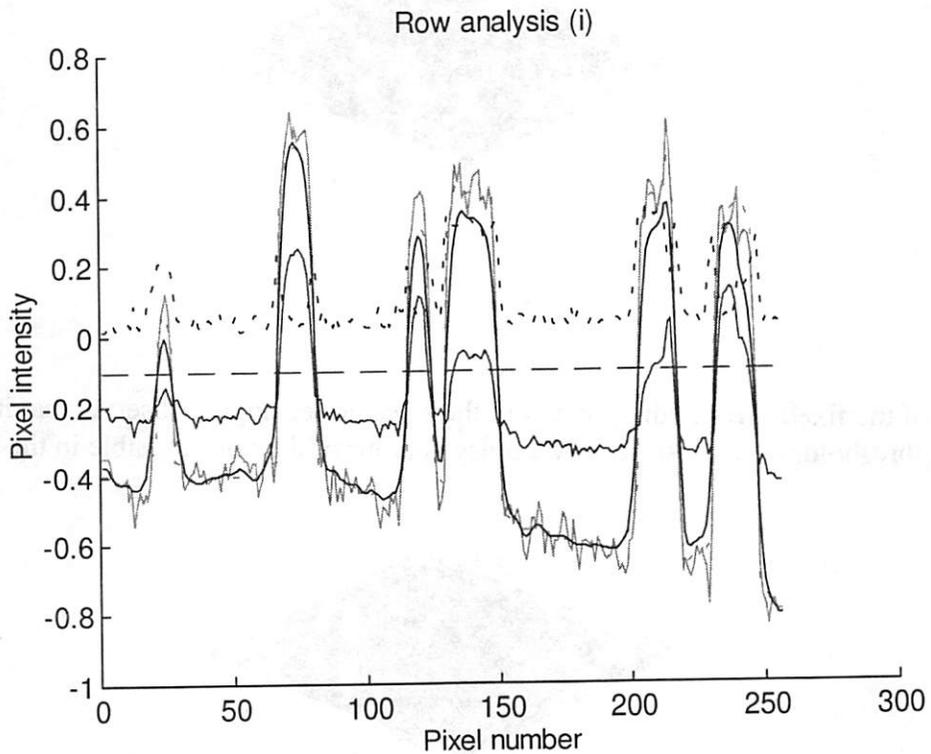
(f) edge map of adaptive segmentation



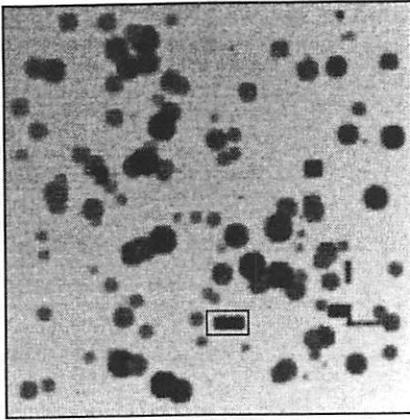
(e) edge map of fixed thresholding



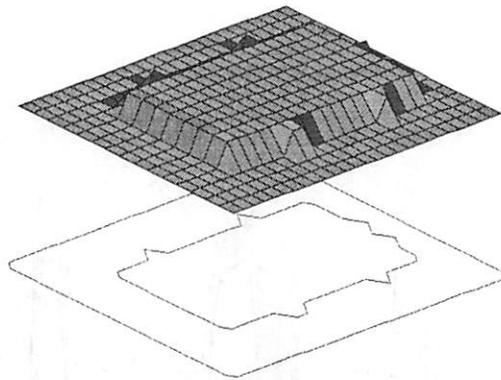
(g) selected row from the original image for 1D analysis



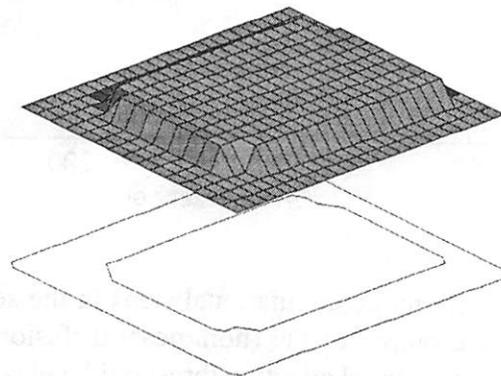
(h) 1D view of the outputs of different computational steps in the segmentation algorithm. The figure shows the original data (-g), the output of the (nonlinear) diffusion (:g), the mean estimate (-b), the variance estimate (--b), and the computed adaptive threshold level (-r) along the selected row (see (g)). The optimal fixed threshold is drawn at the level of -0.1 (--r).



(i) selected rectangle for 3D analysis (containing an artificial “debris” particle)

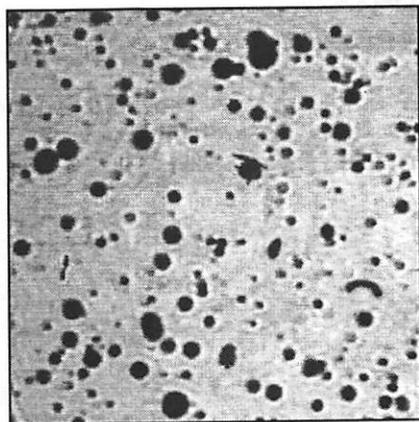


(h) 3D view of the fixed-thresholding output in the selected rectangle. Observe, that in this example optimal fixed thresholding can restore only a noisy contour of the object visible in the original image.

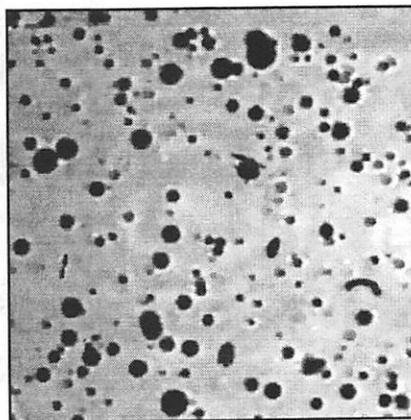


(i) 3D view of the adaptive thresholding output in the selected rectangle. Observe, that the original shape (rectangle) is better restored compared to (h).

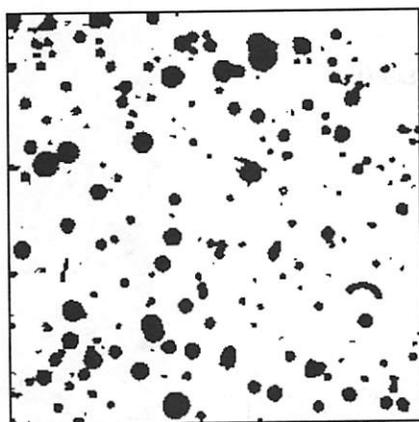
APPENDIX B



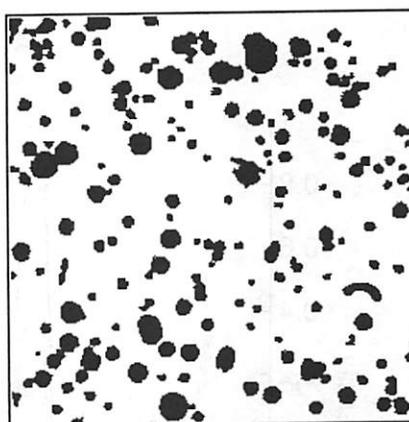
(a) original image



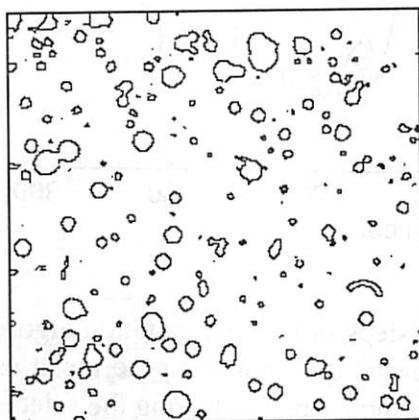
(b) (nonlinear) diffusion output



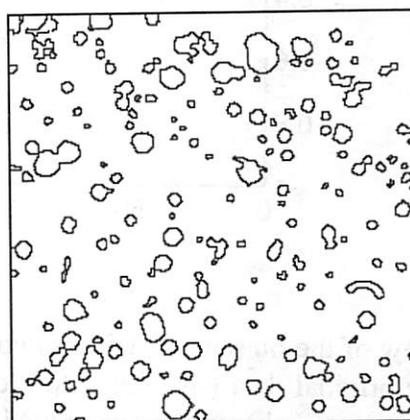
(c) output of fixed thresholding



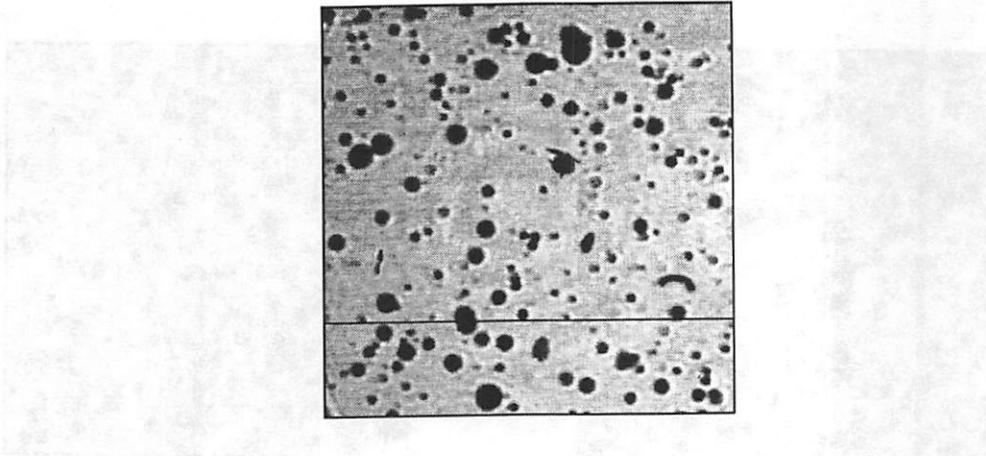
(d) output of adaptive segmentation



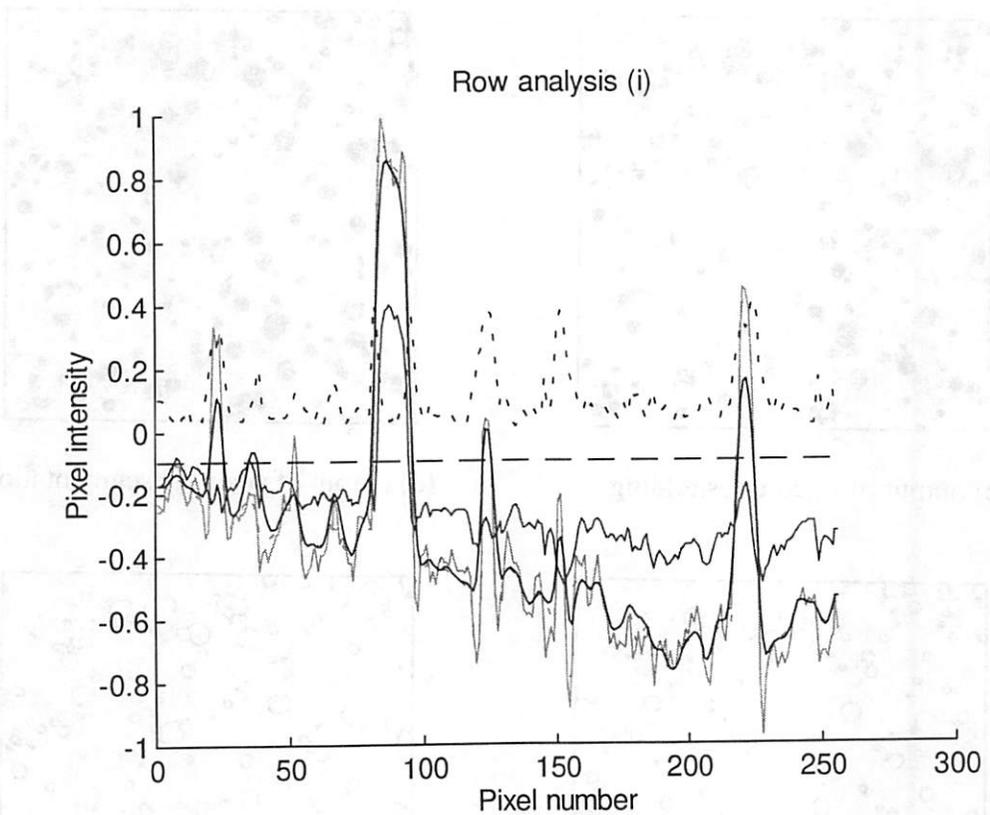
(e) edge map of fixed thresholding



(f) edge map of adaptive segmentation

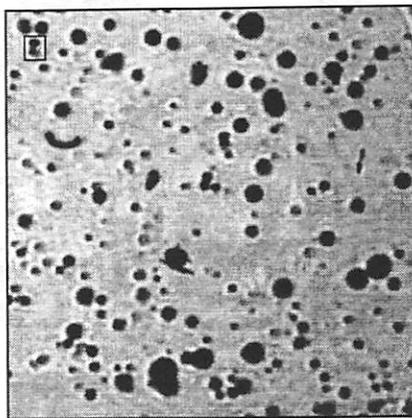


(g) selected row from the original image for 1D analysis

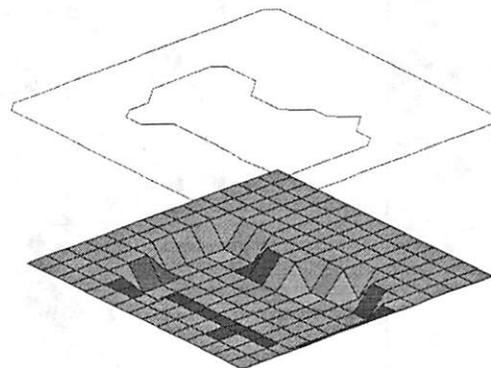


(h) 1D view of the outputs of different computational steps in the segmentation algorithm. The figure shows the original data (-g), the output of the (nonlinear) diffusion (:g), the mean estimate (-b), the variance estimate (--b), and the computed adaptive threshold level (-r) along the selected row (see (g)). The optimal fixed threshold is drawn at the level of -0.1 (--r).

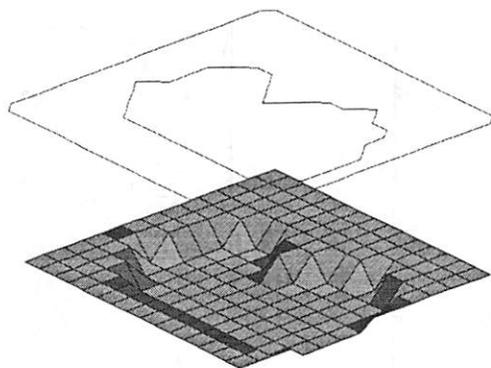
(i) selected rectangle for 3D analysis (containing a bubble-bubble having a gray-scale level close to the background illumination)



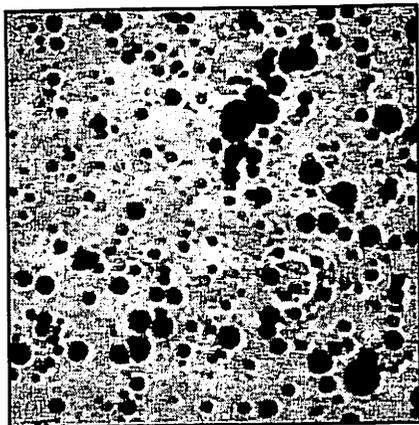
(h) 3D view of the fixed-thresholding output in the selected rectangle. Observe, that in this example optimal fixed thresholding can restore only part of the objects visible in the original image.



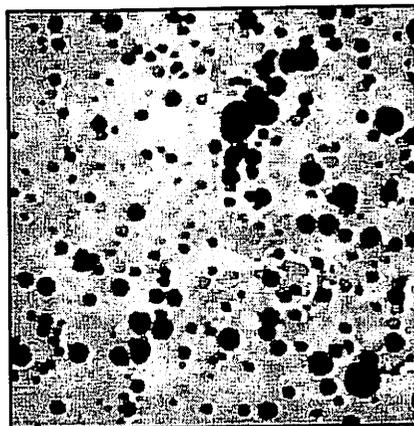
(i) 3D view of the adaptive thresholding output in the selected rectangle. Observe, that the assumed original shape (overlapping circles) is better restored compared to (h).



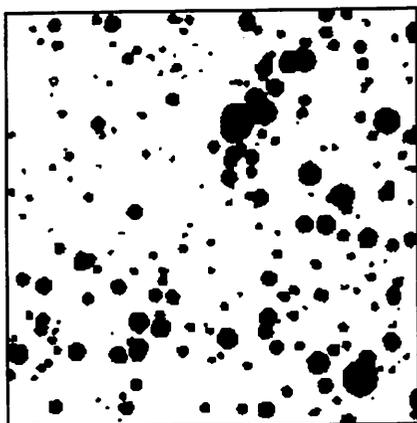
APPENDIX C



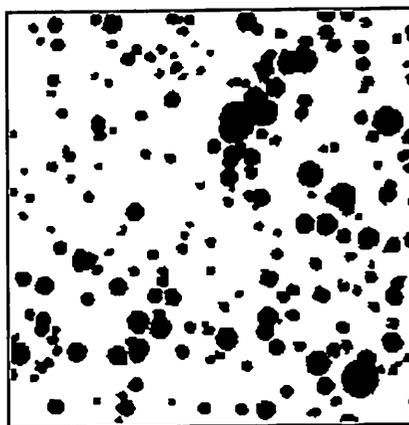
(a) original image



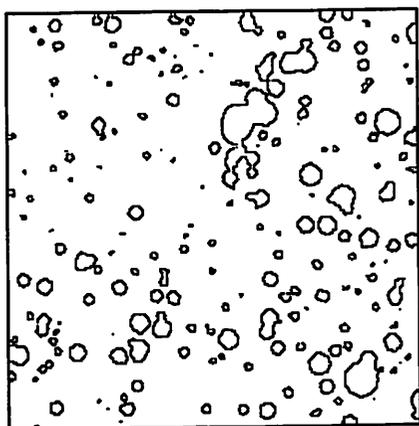
(b) (nonlinear) diffusion output



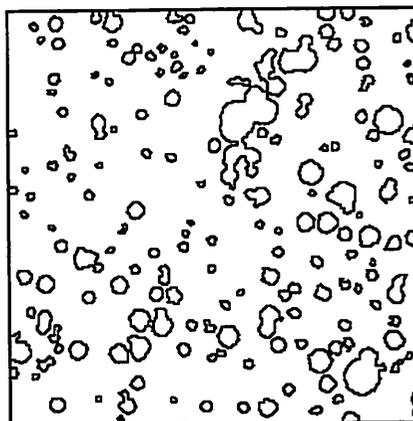
(c) output of fixed thresholding



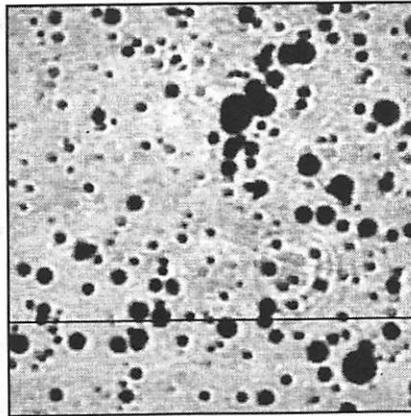
(d) output of adaptive segmentation



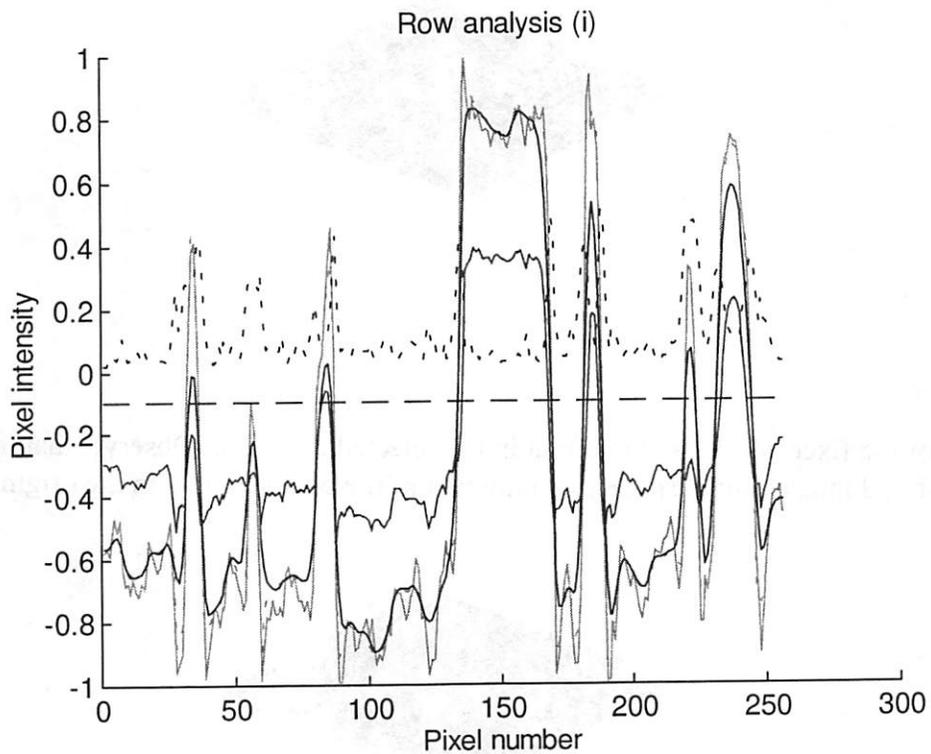
(e) edge map of fixed thresholding



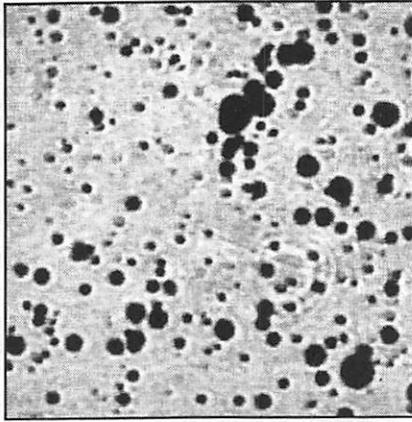
(f) edge map of adaptive segmentation



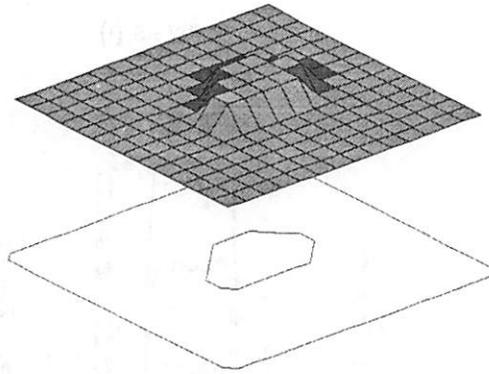
(g) selected row from the original image for 1D analysis



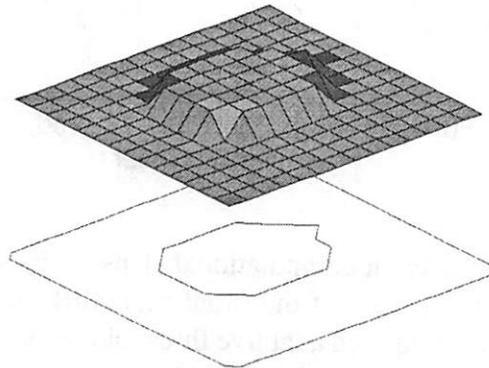
(h) 1D view of the outputs of different computational steps in the segmentation algorithm. The figure shows the original data (-g), the output of the (nonlinear) diffusion (:g), the mean estimate (-b), the variance estimate (--b), and the computed adaptive threshold level (-r) along the selected row (see (g)). The optimal fixed threshold is drawn at the level of -0.1 (--r).



(i) selected rectangle for 3D analysis (containing a bubble that has a gray-scale level close to the background illumination)



(h) 3D view of the fixed-thresholding output in the selected rectangle. Observe, that in this example optimal fixed thresholding can restore only part of the object visible in the original image.



(i) 3D view of the adaptive thresholding output in the selected rectangle. Observe, that the assumed original shape (circle) is better restored compared to (h).