BUBBLE-DEBRIS CLASSIFICATION VIA BINARY MORPHOLOGY AND AUTOWAVE METRIC ON CNN

by

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

Memorandum No. UCB/ERL M97/97

15 December 1997

BUBBLE-DEBRIS CLASSIFICATION VIA BINARY MORPHOLOGY AND AUTOWAVE METRIC ON CNN

by

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

Memorandum No. UCB/ERL M97/97

15 December 1997

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Bubble-debris classification via binary morphology and autowave metric on CNN

István Szatmári[1,3], Abraham Schultz[2], Csaba Rekeczky[1,3], Tibor Kozek[1,3],
Tamás Roska[1,3], and Leon O. Chua[1]

[1]Electronics Research Laboratory, College of Engineering,
University of California at Berkeley, Berkeley, CA 94720, USA

[2]Naval Research Laboratory, Radar Division,
4555 Overlook Ave., S.W., Washington D.C. 20375, USA

[3]Analogical and Neural Computing Laboratory, Computer and Automation Institute,
Hungarian Academy of Sciences, H-1111 Budapest, Kende u. 13-17, Hungary

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

Abstract

In this study, we present the initial results of cellular neural network (CNN) based bubble-debris classification experiments. This research focuses on distinguishing oil debris particles from air bubbles and aims to employ CNN technology to create an on-line (in-flight) fault monitoring system with an extremely low false alarm rate for miss-classified bubbles. The designed analogic[1] CNN algorithm detects and classifies single bubbles and bubble groups using binary morphology and autowave metric. The debris particles are separated based on autowave distances computed between bubble models and the unknown objects. Initial experiments indicate that the proposed algorithm is robust and noise tolerant and when implemented on a CNN Universal Chip it provides a high-speed solution.

[1] analog and logic

Contents

1. Introduction

In many sophisticated systems, it is important to obtain information about the state of a system via mechanical wear analysis. System malfunctions need to be predicted and an appropriate alarm signal should be generated for the operators. This is the case in our specific problem where the system is a jet or helicopter engine. To perform the inspection optical sensing and subsequent image analysis was chosen because the strong mechanical vibration and electromagnetic interference render this problem very difficult to be solved using other types of methods.

Cellular Neural Networks (CNNs) [1-3] are promising candidates in image processing problems where real-time signal processing is required. Cellular Neural Networks are cellular, analog, programmable, multidimensional processing arrays with distributed logic and memory. The processing elements are locally connected. The extension of the CNN paradigm is the CNN Universal Machine (CNNUM) where distributed and global memories and logic functions support the running complex analogic algorithms. The key feature of the CNN architecture is its high operation speed.

In this paper we present the initial result of solving the bubble-debris classification problem where the main task is to separate air bubbles from debris particles. The basic idea behind the analysis is a comparison of all particles to bubble models (circles and overlapping circles) and a classification based on the difference error measurement. The problem of distinguishing debris particles from air bubbles is difficult due to the coarse resolution of the images and the requirement for an extremely low false alarm rate for miss-classified bubbles at very high processing speed.

It will be shown that CNN technology is a promising candidate to solve this problem which requires high-speed image processing and compact design. The developed algorithms can separate air bubbles from debris particles but further investigations are necessary to achieve the desired low false alarm rate.

The paper is organized as follows. Section 2 describes the bubble debris problem and analyses a test system. Section 3 shows the advantages of a CNN based solution. Its elements are discussed in Section 4. Section 5 presents a possible algorithm based on binary morphology operations and autowave metric. Finally, Section 6 shows some examples from the experimental results and Section 7 draws the conclusions.

2. System Specifications and the Present Test System

At NRL (Naval Research Laboratory) a bubble-debris classifier is being developed. Figure 1 shows the test environment in which the imaging system is applied. The imager device is located on the gearbox along a tubular opening in which the lubricant flows at about 10 m/s speed. A pulsed laser illuminator projects an image of the fluid and various suspended object in it into a CCD sensor. These images are to be processed and analyzed by the proposed system.

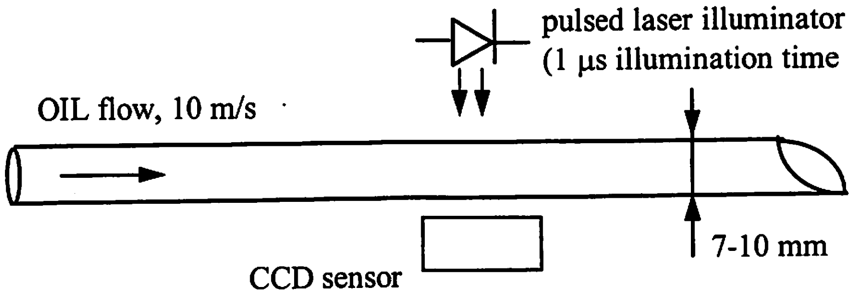


Figure 1 *Optical sensing setup for the condition monitoring system*

The present test system processes images of 512x512 pixels at 500 images/s frequency using a DEC alpha processor-based system. To obtain a deployable solution, however, further miniaturization is necessary.

Figure 2 shows a typical image to be processed. An image contains worn off particles, cuttings, sand, and air bubbles which may be stuck together or appear to be so due to occlusion. The major goal is to separate debris particles from the air bubbles. The problem of distinguishing debris particles from air bubbles is difficult due to the coarse resolution of the images and the requirement for an extremely low false alarm rate for miss-classified bubbles.

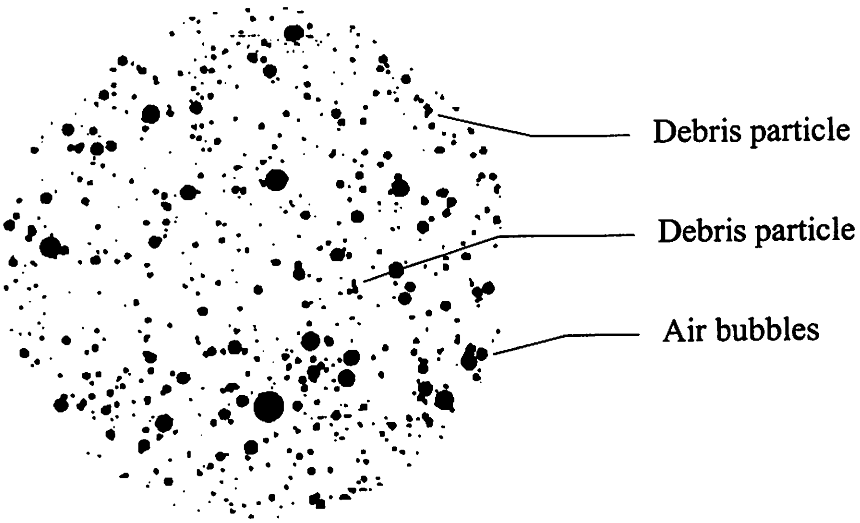


Figure 2 *Typical image of the oil flow with floating particles as recorded by the CCD sensor. The pixel size is about 14μm.*

Figure 3 shows the density and size distribution of various objects present in the oil flow. For the generation of alarm signals only objects larger than about 50 μm are interesting because these include cuttings and worn off particles that provide information about mechanical wear-off. Because the size-ranges of different objects overlap it is not enough to only determine the size of an object, but classification should be carried out based also on its shape, and features such as aspect ratio, external compactness and bending energy.

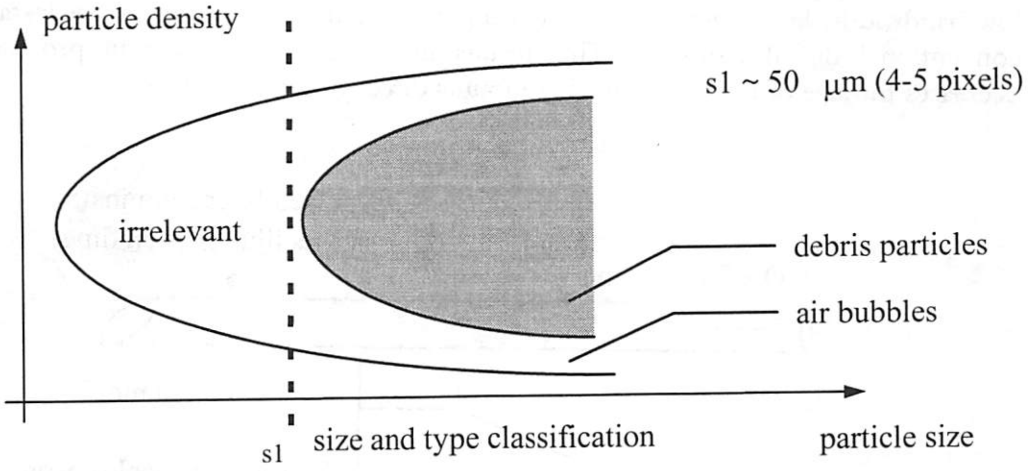


Figure 3 Particle size/density distribution in the lubricant. Small debris particles are excluded because they relate to a benign wear condition.

The approach followed at NRL 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 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 of 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 NRL system cannot be easily implemented in this environment and it is necessary to define a new feature set.

3. CNN-based Solution

CNN technology [1-3] is a promising candidate to solve this problem which requires high-speed image processing and compact design. Figure 4 shows the proposed solution. The main advantage offered by CNN is that high volume image processing takes place close to the optical sensor alleviating the need for high throughput data transfer and only low-bandwidth, high relevance data is output from the sensor that is easily tackled by conventional digital hardware. This results in significant speedup in processing and decreases the size of the apparatus that mounts directly on the gearbox.

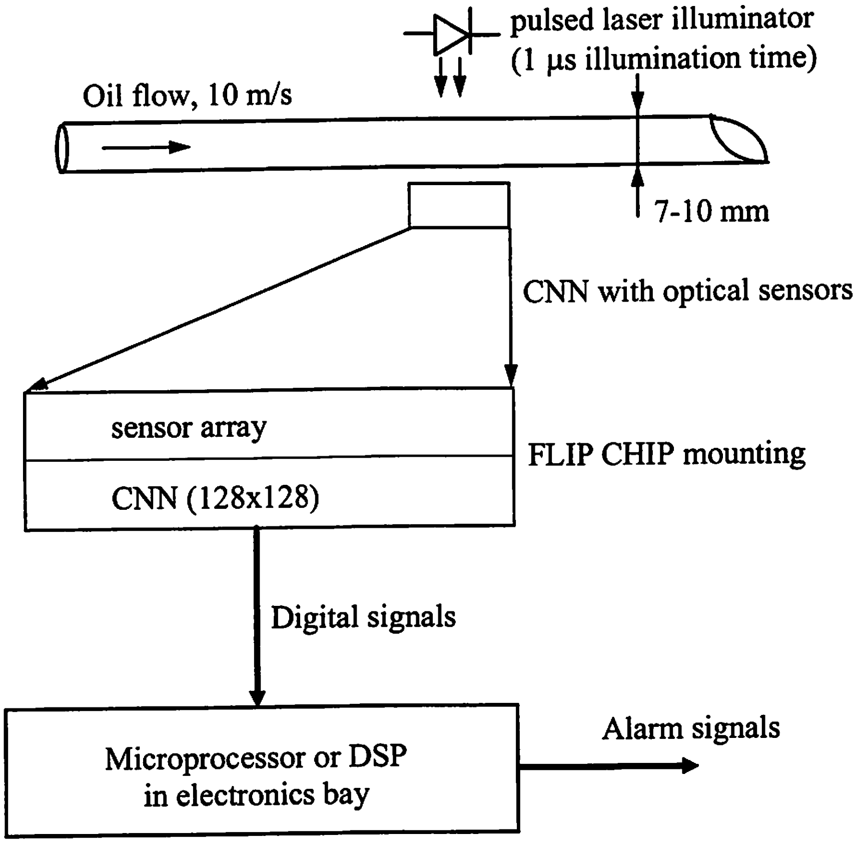


Figure 4 *Outline of the proposed CNN-based solution. By employing image sensors mounted directly on the CNN chip, sensing and high-volume processing can be performed at the same place.*

The CNN processor can be used to solve the following tasks:

A) size classification (to filter too small objects)

B) air bubble / non-bubble detection (to filter extraneous objects)

C) measurement of various object features for subsequent classification (size, aspect ratio, surface roughness)

Figure 5 shows the block diagram of the proposed approach.

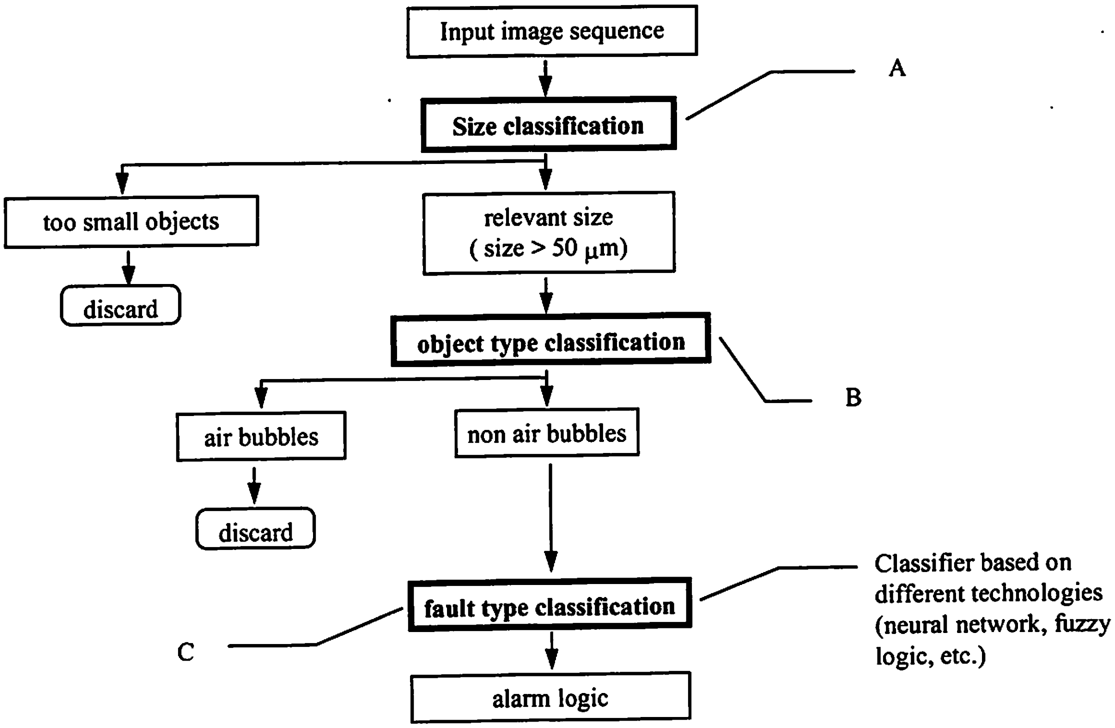


Figure 5 Block diagram of the processing of oil flow images

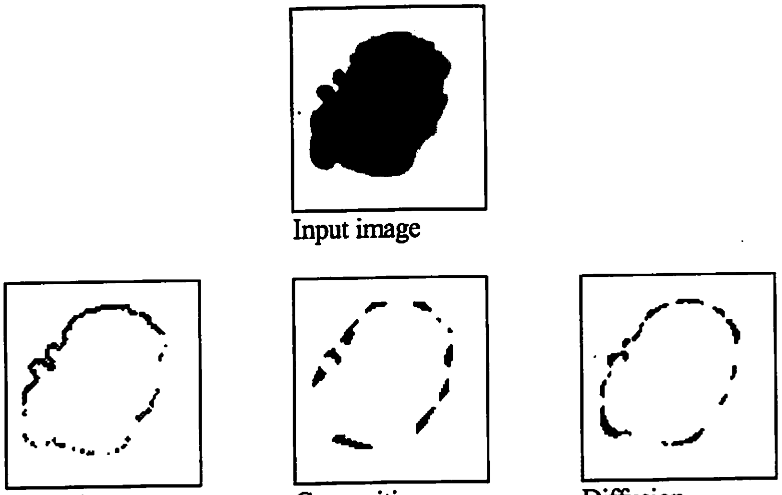
4. Elements of the CNN Solution

4.1. Size Classification

There are well studied methods for size classification in CNN. These usually employ a series of peeling and comparison operators, as well as local logic operations to identify binary objects of specific sizes. These methods are directly applicable to our problem and can be used with little modification.

4.2. Roughness Measurement

Next, we present possible methods to measure the surface roughness of objects. This computed quantity can be used at the second stage of processing (object type classification) and at the third stage (fault type classification), as well. Figure 6 shows the results of three different methods for computing surface roughness.



Input image

Laplacian

Concavities

Diffusion

Figure 6 *Techniques for determining surface roughness: the methods used a Laplacian operator, concavity detector, and diffusion operators, respectively.*

4.2.1. Using the Laplacian

First the Laplacian of an image is computed and the pixels in this image are squared (this can be approximated by the absolute value function that gives a close estimate in the interval considered). This image is then thresholded, and converted into a binary image. The number of connected components of this image is a measure of roughness of the boundary of the original object. Figure 7 shows the steps of the algorithm.

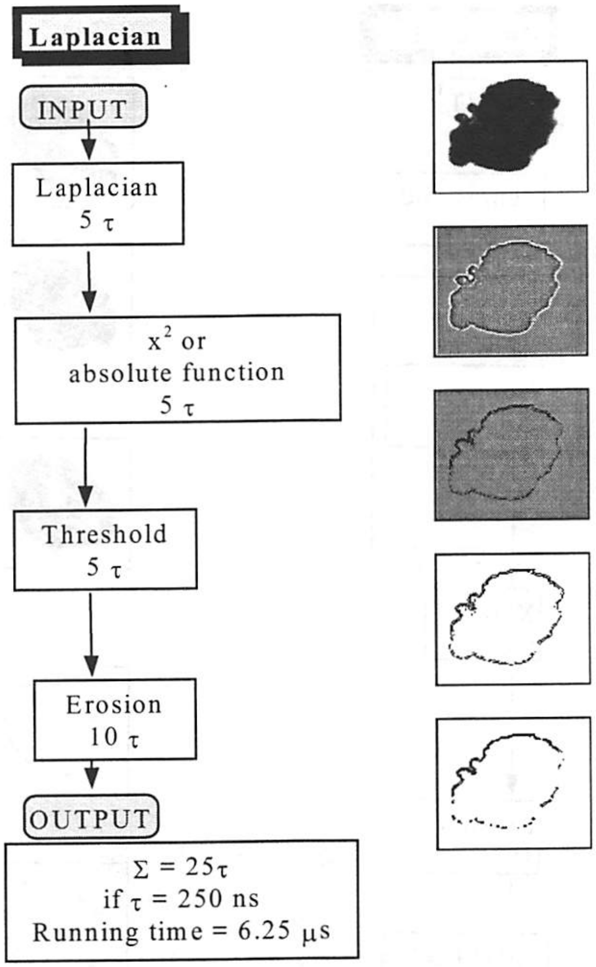


Figure 7 *Measurement of surface roughness based on the Laplacian*

4.2.2. Finding Concavities

The basic idea here is to find the concave parts of objects. First the gray-scale image is converted into binary image via thresholding operation. Next, pixels are driven to black which are located at concave places using the "hollow" template. This template turns black all those white pixels which have at least four black direct neighbors. We call concave those white pixels which are surrounded by black pixels from at least four of the eight possible directions. The network transient must be stopped after a given amount of time depending on the size of the largest holes to be filled in. Next we extract concavities of objects using logical XOR operation between the thresholded image and filled image. Figure 8 shows the steps of the algorithm.

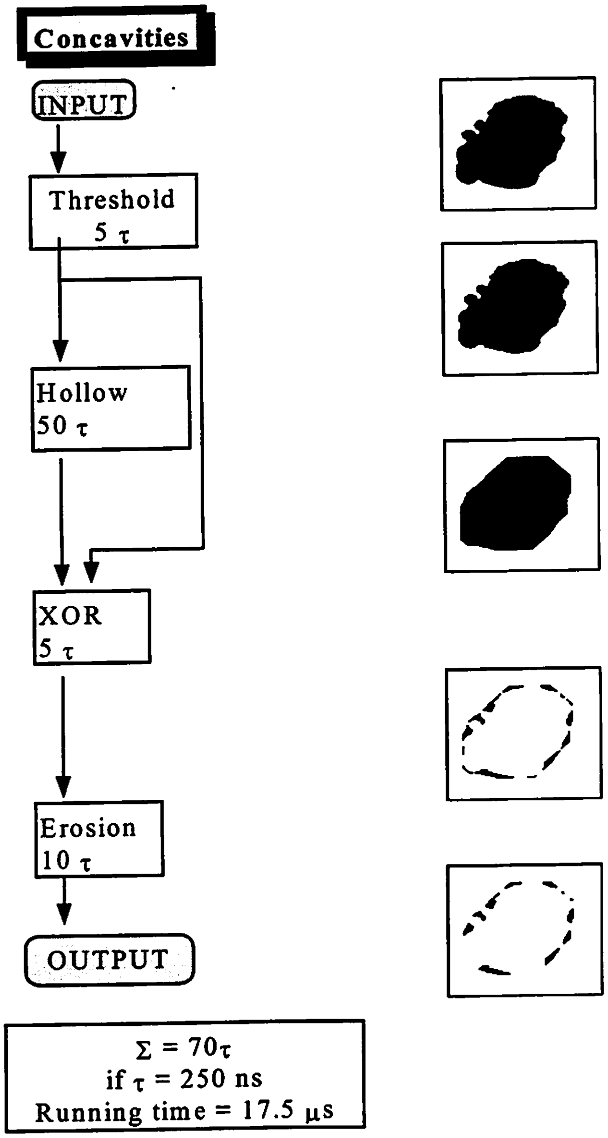


Figure 8 *Measurement of surface roughness based on finding concavities*

4.2.3. Diffusion

This method computes surface roughness using a diffusion operator. The image is smoothed in order to fill the rough parts of the object and between this and the original a difference is computed. First the image is thresholded, then diffusion is applied to the thresholded image. The diffused image is thresholded again and logical XOR operation is applied between the two thresholded images. Spurious isolated pixels are removed by erosion. Figure 9 shows the steps of this algorithm.

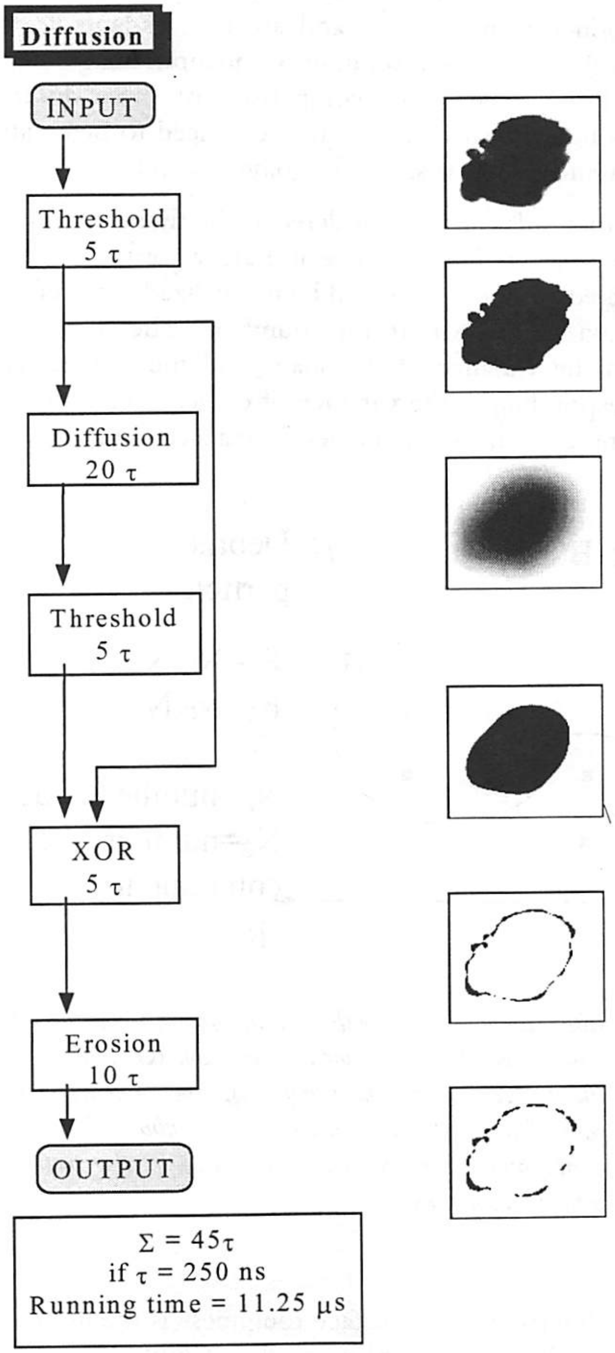


Figure 9 *Measurement of surface roughness based on diffusion*

4.2.4. Comparison of Methods Measuring Roughness

The first method using the Laplacian of the image is the fastest of the three approaches studied here. Its implementation, however, requires the use of a local arithmetic unit for pixel-wise multiplication of the image. This feature, while feasible, may not be available in the first CNN processors. Also, the quality of the output seems to depend on the local contrast of the edge contour. The other two methods involve only simple 3x3 linear templates and local logic (XOR), the functionality readily available in CNN arrays. They both seem to produce robust output and are easily adaptable to varying scales (object sizes). This is desirable since the resolution of the input image may vary with the particular imager used [4]. For a qualitative comparison of these different methods measuring surface roughness an artificial object sequence is need to be created with different shapes and curvature roughness. This is still an ongoing research.

The presented methods can be considered as the first step toward measuring the surface roughness of the objects. In a more elaborate algorithm the detected pixels and the number of connected components could be normalized with reference to the object's area. This way we would obtain two feature numbers. The normalized number of detected pixels referring to the variance of the shape and the normalized number of connected components corresponding to the variance of object's curvature. Based on these numbers a robust classification scheme could be developed (see Figure 10).

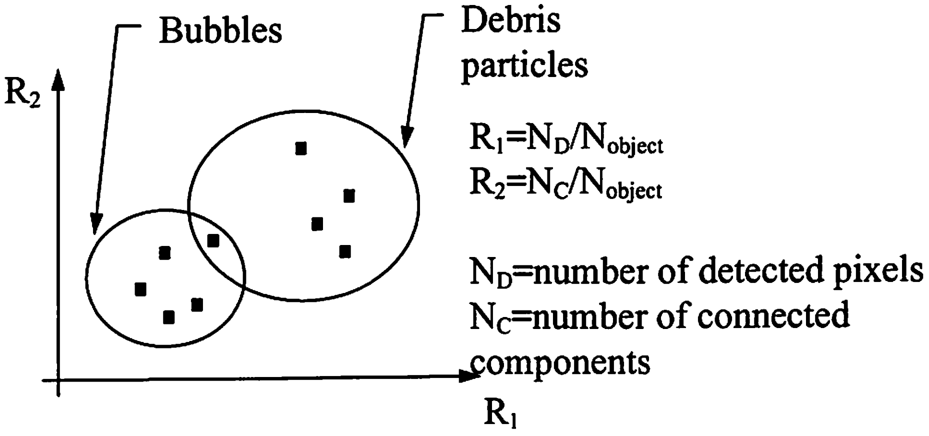


Figure 10 *Illustration of the hypothesis that combining features from roughness measurements could be effectively used for classification. R1 and R2 are the computed features from the surface roughness measurements. The computed characteristics should be larger in case of debris particles than for the bubbles. The size of the intersections of the overlapping sets might be decreased using further methods. This could be exploited in a fuzzy algorithm.*

Another possibility to measure surface roughness is the use of fractal dimensions. With proper simplification this can be implemented on CNN.

These characteristics and the results of the autowave metric (see later on) based detection could be exploited in a fuzzy algorithm. Such a method seems to be necessary to arrive to a robust and reliable solution of this classification problem.

5. Bubble/Debris Type Detection Based on Binary Morphology and Autowave Metric

The approach followed here is to filter out air bubbles using CNN based binary morphology [5-6] and autowave metric. In the experiments we used binary images, the output of the field-programmed gate array that in the current system thresholds all gray-scale input images at a fixed level right after the acquisition. In the companion paper [16] we investigate how the quality of these binary images can be improved using a CNN based locally adaptive front-end filtering and segmentation strategy applied to the original gray-scale images.

Figure 11 shows the flowchart of the bubble-debris classification algorithm containing Feature Extraction, Bubble Models generation, and Autowave Distance computation. The Feature Extraction block detects the center points of all objects and measures their sizes (Radii Map). These two characteristics are used for bubble model generation via autowave. Autowaves propagate from the center points and grow circles around them until the Radii Map stops this propagation (block Bubble Models). The autowave distance is computed between bubble models and the input image (block Autowave Distance). If the measured distance is large the object will be classified as a debris particle.

First, using binary morphology the centers of all objects are detected. The critical part is to find the exact mass center of an object. In case of multiple bubble groups, we need to obtain more than one center point. Around these center points, bubble models are grown for pattern matching. These models are a set union of circles based on the hypothesis that the object is a group of one or more overlapping air bubbles. In the algorithm the erosion operator is used to find the center points and radii of all components of a bubble group. Then the autowave metric is employed to generate a "dynamic bubble model" for pattern matching and compute the distance between two partially overlapping sets. The autowave distance between the bubble model and the unknown object is used for classification. If the object is a multiple bubble this distance will be relatively small compared to what it would be if the object were a debris particle.

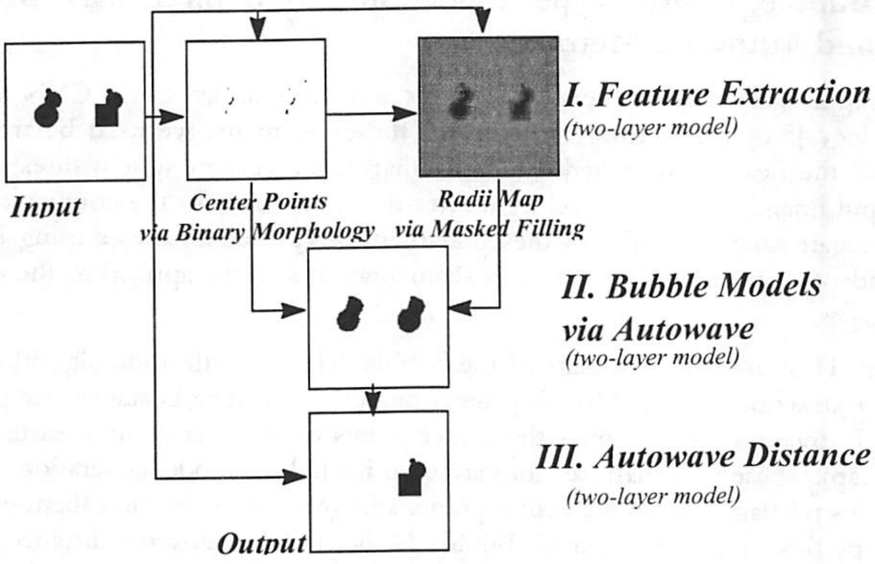


Figure 11 *Flowchart of the bubble-debris classification algorithm containing Feature Extraction, Bubble Models generation, and Autowave Distance computation. The artificial example contains two objects. The left one consists of three overlapping bubbles and the right one contains two circles and one rectangle. This object should be classified as a debris particle. The Feature Extraction block detects the center points of all objects and measures their sizes (Radii Map). These two characteristics are used for bubble model generation via autowave. Autowaves propagate from the center points and grow circles around them until the Radii Map stops this propagation (block Bubble Models). The autowave distance is computed between bubble models and the input image (block Autowave Distance). If the measured distance is larger than a pre-defined threshold level the object will be classified as a debris particle.*

5.1. Feature Extraction

The first block of the classification algorithm (Feature Extraction) detects the center points of all objects and measures their sizes. These are the major characteristics which are required for bubble model generation. Figure 12 shows the steps of the Feature Extraction. Using binary morphology the objects centers are extracted and at the same time a second image is constructed containing information related to radii of objects. Both the computational cost and time are small. Another advantage is that these two characteristics (center points and radii information) can be used for a straightforward bubble model generation.

5.1.1. Object Centers and Radii Map Using Binary Morphology

The term morphology refers to the study of form and structure within an image. All morphological processing depends on the concept of fitting structuring elements. We will only consider binary morphological operators. The primary morphological operators on binary images are erosion and dilation. Erosion represents the probing of an image to see where some primitive shape fits inside the image. Dilation is the dual operation to erosion and is defined in terms of it relative to image complementation. The erosion of set A by set B is defined by

$$E(A, B) = \{x: B+x \in A\} \qquad (1)$$

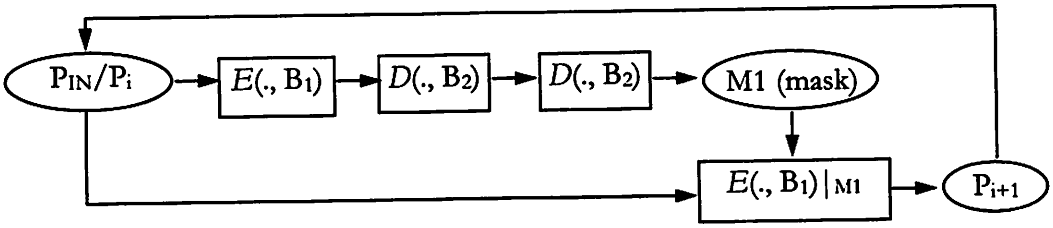
Relative to erosion, we call A the input image and B the structuring element. If the origin is contained within the structuring element, then the eroded image is a subset of the input image and erosion has the effect of shrinking the input image. This property indicates that erosion can be used to find center points of objects.

The second most basic operation of binary mathematical morphology is dilation. It is a dual operation to erosion, meaning that it is defined via erosion by set complementation. The dilation of set A by B is defined by two equivalent formulation

$$D(A, B) = \{[E(A^C, -B)]^C\}, \quad \text{or} \quad D(A, B) = \cup\{A+b: b \in B\} \qquad (2)$$

where A^C denotes the set-theoretic complement of A, and -B denotes that B is rotated around the origin. If B contains the origin, then dilation of A by B results in an expansion of A.

The method to find the center points of the objects is as follows. A morphology operation called masked erosion is now defined. The image contains objects of different sizes that can be specified in number of morphological steps. Using a simple erosion procedure smaller objects would disappear before finding their center points while larger objects would still contain many pixels. A masked erosion approach can prevent smaller objects to disappear i.e. those sets which would vanish with the next application of the erosion operator. One step of masked erosion which is applied several times is



The mask image (M1) is a black & white image where black pixels indicate those locations where a template operation might have an effect. The change of states of those cells that are located at white pixels is prohibited. In our case, the mask should contain only the larger objects which won't totally disappear during an erosion i.e. the erosion operation may have effect there while at small objects should be prohibited. The mask is generated in three steps. First, the erosion operation deletes smaller objects. After that the two dilation operations grows the objects larger than their original size. This mask image (M1) enables the erosion only at locations of larger objects.

The template of erosion is (B term contains the kernel)

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \qquad B = \begin{array}{|c|c|c|}\hline 0.5 & 1 & 0.5 \\ \hline 1 & 1 & 1 \\ \hline 0.5 & 1 & 0.5 \\ \hline \end{array} \qquad z = \boxed{-6.5}$$

The template of dilation is

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \qquad B = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad z = \boxed{8.5}$$

It is important to note that for dilation a slightly different kernel should be used than for the erosion. The dilation should be repeated at least twice. The reason for that lies in the fact that the erosion operation is not invertible in general (E.g. If we use the same kernel for dilation as for the erosion we should apply dilation at least three times otherwise false center points will appear).

Based on a priori information, we know the approximate sizes of the largest objects and it is possible to estimate the maximal number of steps that should be used for the masked erosion in order to get a picture containing the centers of the objects.

While center points are computed the radii information of objects is also extracted. The Figure 12 shows the block Feature Extraction in details.

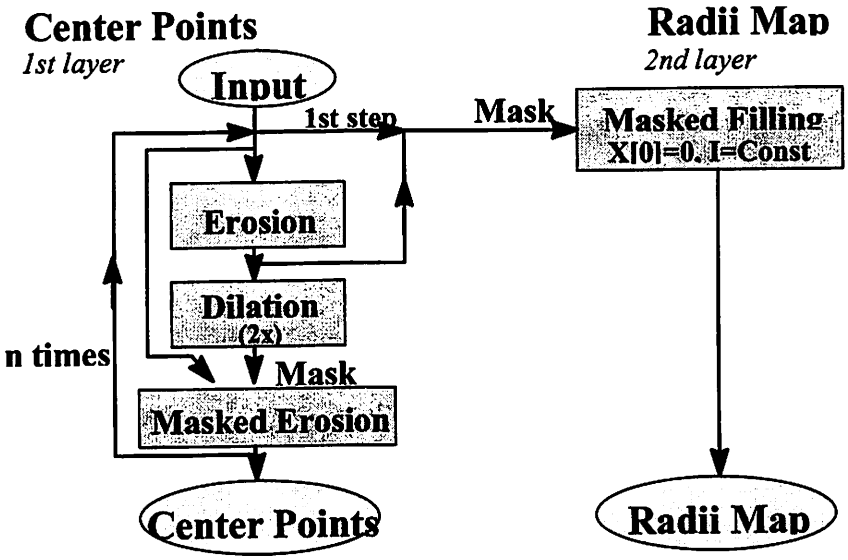


Figure 12 *The Feature Extraction block detects the center points of the objects and measures their sizes constructing two images, namely the Center Points and the Radii Map. The Center Points will be the trigger points of autowaves generating bubble models while the Radii Map will control the propagation of autowaves. To extract the center*

points erosion operation is used shrinking objects step by step. The image contains objects of different sizes that can be specified in number of morphological steps. Using a simple erosion procedure smaller objects would disappear before finding their center points while larger objects would still contain many pixels. A masked erosion approach can prevent smaller objects to disappear i.e. those sets which would vanish with the next application of the erosion operator. While center points are computed the radii information of objects is also extracted. A layer will be filled with constant current and the voltage levels of cells will indicate the sizes of the corresponding objects. Each time the result of the first erosion operation of the masked erosion is used for this filling. At the end of the process the state levels of cells at locations of center points will have the largest values and gradually decreasing around them.

The steps of constructing Radii Map are as follows. A layer will be filled with constant current and the voltage levels of cells will indicate the sizes of the corresponding objects. We use the results of process of center points detection. The output of the first erosion is used as a mask to control this filling. This mask will contain smaller and smaller patches around the center points as erosion goes on and shrinks the objects in several steps. At the end of the process the state levels of cells at locations of center points will have the largest values and gradually decreasing around them. The constant current is set to a given level in such a way that the state values of cells at positions of center points of objects will have values in range [0.0; 1.0]. The largest object results in the largest cell value at its center point.

5.2. Bubble Models

In the next step the autowave approach is used for generating bubble models around the centers and utilize them for pattern matching. Figure 14 shows this process which will be discussed later in detail.

5.2.1. Autowaves

The autowave approach has several advantages for pattern recognition [7-8]. Autowaves represent a particular class of nonlinear waves which spread in active media at the expense of the energy stored in the medium [9-10]. The properties of autowaves basically differ from those of waves in conservative systems, including nonlinear waves. The autowave, being a wave, can diffract and according to the Huygens' principle, bypass obstacles. However, it has unusual properties. Two waves spreading in opposite directions do not pass through each other, as is usual for waves, but mutually annihilate similar to particles. The fundamental properties of nonlinear waves in conservative systems and of autowaves are summarized in Table 1.

Table 1. Properties of waves and autowaves

Properties	Waves	Autowaves
Reversibility	+	–
Conservation of energy	+	–
Conservation of amplitude and waveform	–	+
Reflection	+	–
Annihilation	–	+
Interference	+	–
Diffraction	+	+

Autowaves can be described by a PDE of the form

$$\frac{\partial u}{\partial t} = D\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right] + f(u) \qquad (3)$$

Here, $\partial u / \partial t$ is the rate of change of intensity values of the image intensity u. It is induced by f(u) plus the diffusion term $D\ (\partial^2 u / \partial x^2 + \partial^2 u / \partial y^2)$.

Applications proposed for autowaves [7], can be realized by a CNN structure. Autowaves were observed in a CNN array that have Chua's circuits as cells [11-12]. There the nonlinear resistor of the chaotic oscillators provided the active local dynamics. Such a system can be built using a simpler CNN architecture with the original cell-type [13-14]. There a single-layer architecture was shown where the active local dynamics were generated with a delay-type template which resulted in a bistable system.

We will focus on the simplest type of autowaves called traveling or trigger waves where transition of the state of a cell can propagate in the system. It should be noted that trigger waves do not have the annihilation property. We only need the conservation of amplitude during propagation. The templates generating trigger waves can be

1. (a) linear template

$$A = \begin{array}{|c|c|c|} \hline 0.41 & 0.59 & 0.41 \\ \hline 0.59 & 2 & 0.59 \\ \hline 0.41 & 0.59 & 0.41 \\ \hline \end{array} \qquad z = \boxed{4.5}$$

Given static binary image **P**

initial state: **X**(0) = **P**

The initial state contains the trigger points of the waves.

1. (b) linear template

$$A = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 2 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \qquad z = \begin{array}{|c|}\hline 4.5 \\ \hline \end{array}$$

Given static binary image **P**

initial state: **X**(0) = **P**

input: U(t) = **P**

Both the initial state and the input contain the trigger points. This template is known as the patchmaker template.

Both templates can be derived from the same base template. Although these two linear templates are binary propagating type they do not belong to the class of the equation (3). Their advantages are the easy implementation and already can be tested on CNN chips. The speed can also be adjusted through the bias term although it effects some changes in their properties (e.g. the number of black neighbors which is needed to exceed the activation).

2. approximation of the wave equation (3)

By proper discretization of equation (3) we obtain:

$$\frac{d}{dt}u(x,y,t) = D\ \left(\frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} \right) + f(u)$$

$$\approx D\ \frac{1}{4}\left(u_{ij-1}(t) + u_{ij+1}(t) + u_{i-1j}(t) + u_{i+1j}(t) \right) - Du_{ij}(t) + f(u_{ij})$$

The autowave equation can be directly mapped onto the CNN array (D=1) resulting in the following simple template (**AUTOWAVE.TEM**, ***X(0)*** - *ORIGINAL IMAGE*):

$$A = \begin{bmatrix} 0 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix}, \quad \hat{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & f(u) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = 0, \quad I = 0$$

In the middle of template $\mathcal{A}$ the effect of $-1/R$ in the CNN equation is considered (R=1). Possible modifications and specification of the nonlinearity:

A1 =

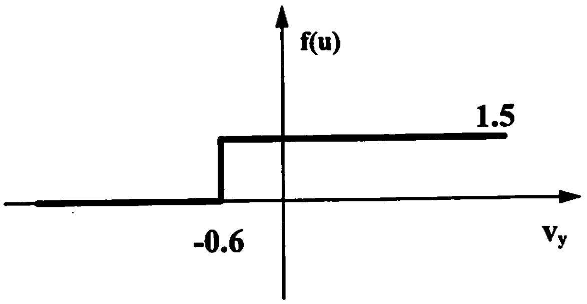
0.1	0.15	0.1
0.15	0	0.15
0.1	0.15	0.1

this is the term $D\ (\partial^2 u / \partial x^2 + \partial^2 u / \partial y^2)$

A2 =

0	0	0
0	f(u)	0
0	0	0

this is the term $f(u)$ where it is



Given static binary image **P**

initial state: **X**(0) = **P**

The initial state contains the trigger points of the autowaves. Although the f(u) is the simplest nonlinearity useable for autowaves it is still not available on the existing CNN chips. The advantages of this implementation is that the speed of the waves can be adjusted.

5.2.2. Generating Bubble Models

We use autowave to generate "dynamic bubble models" as follows. The picture containing center points of bubble groups are the triggers of autowaves. The waves will spread from these points producing larger and larger circles around those center points. Knowing that these set union of circles contain different circles in size, it is necessary to constrain the propagation of the autowave. The mask operator will allow this restricted propagation to take place. This mask should be constructed in such a way that it contains information related to radii of objects. Thus the propagation of the autowaves can be stopped at proper time. The Radii Map will be used as a mask for this task. The Radii Map was extracted at the Feature Extraction. The Figure 13 shows a part of a Radii Map.

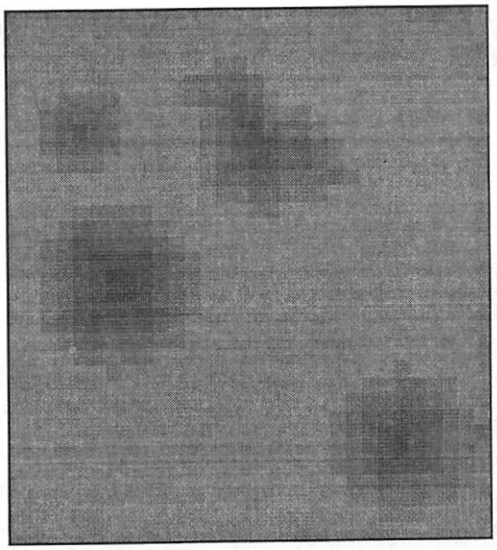


Figure 13 *Part of a Radii Map. The Radii Map contains information related to radii of objects. The cells at location of center points of objects contain the largest state values and cells around them have gradually decreasing levels. The value of a peak at location of a center point indicates the radius of the given object. This image is used as a mask image to control the spread of autowaves generating "dynamic bubble models".*

The Figure 14 shows the process of generating bubble models. The trigger points for autowaves is the image which contains the center points. Without any constraint the waves would propagate through all the cells of the network. The gradually smoothed Radii Map will control the spread of autowaves enabling the propagation only at those position where cell values have positive levels. If the pixels around center points reach the zero level the propagation of autowaves will be stopped.

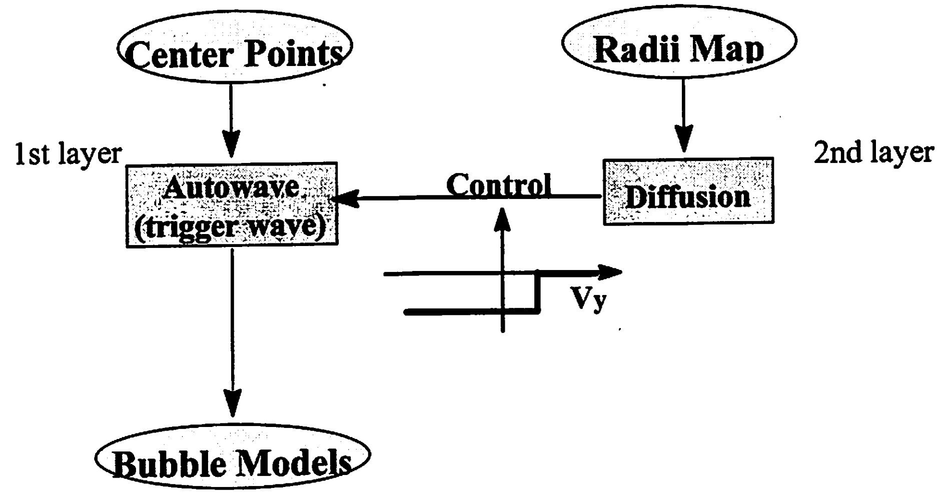


Figure 14 *Generating bubble models. The center points of the objects are trigger points of the autowaves. To stop the spreading of waves a time variant mask is applied. This mask contains information related to radii of objects. Where the cell values of the mask are larger than a given level then the spreads of autowaves are enabled. The diffusion on the Radii Map results in a gradually smoothing. When cell values decrease below a given level then autowaves will be stopped.*

The templates are the following. To generate autowaves the nonlinear template is used which was discussed earlier. The diffusion template is

A1 =

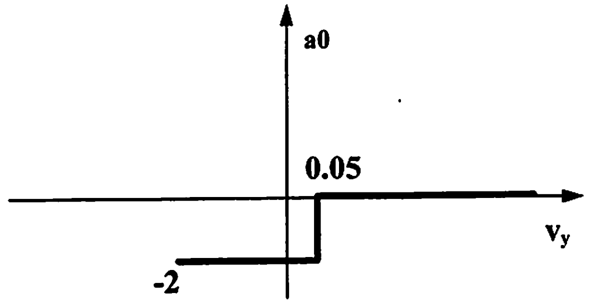
0.1	0.15	0.1
0.15	0	0.15
0.1	0.15	0.1

The template which control the spreads of autowaves is

A2 =

0	0	0
0	a0	0
0	0	0

where a0 is the following function.



The major constraint of the position of the breakpoint is that it should exceed zero.

5.3. Classification Employing the Autowave Metrics

Below we discuss in detail how the autowave approach can be applied to the problem of image classification or recognition via comparison with prototypes (pattern matching). This comparison requires the measurement of the coincidence of two different overlapping point sets. One possibility is to compute the Hamming distance between point sets. The Hamming distance (number of different points) is sensitive to the image shift and presence of noise.

Another known distance metric is the Hausdorff metric which is more tolerant to shift and noise [15]. The Hausdorff distance can be easily measured using autowaves. Such measurement would require the generation of the trigger wave, whose initial position coincides with one image. The wave propagates until all the points belonging to another image become triggered. The time required is equivalent to the asymmetric Hausdorff distance. To measure the Hausdorff distance, we should perform this operation twice with initial position coinciding with another image. The Hausdorff distance is extremely unstable in the presence of noise. The appearance of a noisy spot might drastically change the computed distance.

A variant of the Hausdorff metric called autowave metric which has several advantages over Hausdorff metric will be used in our experiments [8]. Let A and B be two objects to be compared. We consider the case when the compared sets A and B are partially overlapped closed contiguous regions. Let the wave spread only through the points belonging to the union of $A \cup B$, instead of spreading everywhere. The time required for the wave to occupy the union $A \cup B$ can be used to define the measure of the difference between two objects. To measure the asymmetric autowave difference $w_a(A, B)$, the initial position of the wave should coincide with A, and vice versa. The symmetric autowave distance $w_s(A, B)$ is given by

$$w_s(A, B) = \max\left(w_a(A, B), w_a(B, A) \right)$$

The symmetric autowave distance has the following properties:

(1) $w_s(A, B) = 0$ if and only if $A = B$

(2) $w_s(A, B) = w_a(A \cap B, A \cup B)$

To measure the symmetric autowave difference the trigger points of the wave are the points belonging to the intersection of two point sets. (see Figure 15).

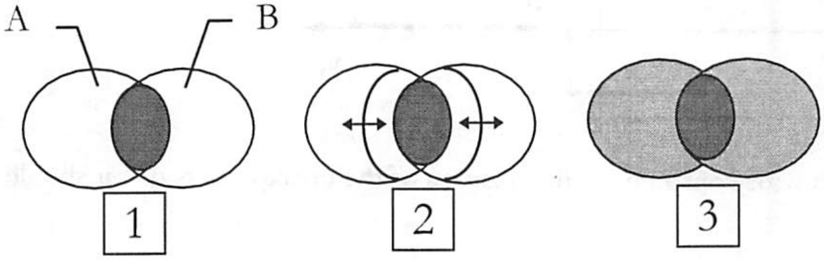


Figure 15 *Autowave distance between point sets. (1) Two partially overlapping point sets. (2) The autowave spreads from the intersection through the union of point sets. (3) The wave propagates until all the points belonging to the union of point sets become triggered. The time required to occupy the union can be used as a measure of the difference between A and B.*

The properties of autowave distance provide increased tolerance to noise effects than Hausdorff distance. For instance, if two images exactly coincide, except for only one exceeding pixel apart from the image, then the Hausdorff distance may be large depending on the position of the exceeding pixel, whereas the autowave distance between these images would be zero.

Figure 16 shows a possible implementation of the autowave metric on CNN. Two images will be compared, namely, the objects and the bubble models. The idea is that autowaves will spread out from intersections of corresponding objects and bubble models and the time is measured while autowaves occupy the union of objects and bubble models. The recall template operation produces an image which contains only those set unions of objects and bubble models where intersections are not empty (i.e. they are partially overlapped). This image is used as a mask (M4) to control the autowaves. The trigger points of autowaves are the intersections of sets to be compared. At the time while autowaves are propagating an another layer is filled with a constant current to measure the time. This is denoted as 'Controlled Filling'. The current has two components. One of them is determined by the mask (M4) the another one comes from the layer where autowaves propagate. Due to the binary wave propagation the current is 2a at a given cell iff the mask is black but the autowave has not reached yet this position otherwise it is zero. At the end of the process the cells at boundaries of the unions of objects and bubble models will contain the highest voltage levels. This will be thresholded and indicate large difference if any. At last those objects will be recalled where these differences are large.

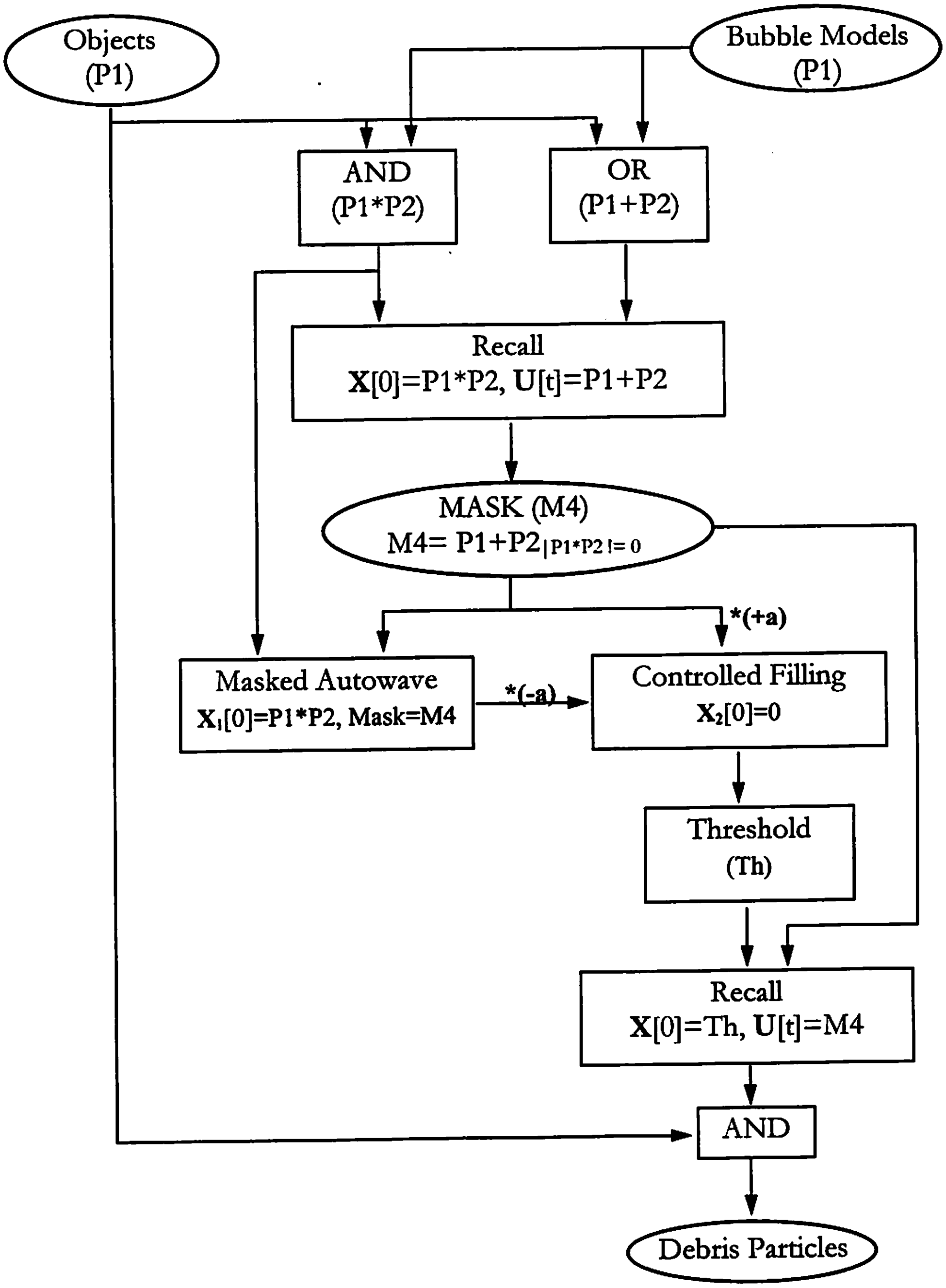
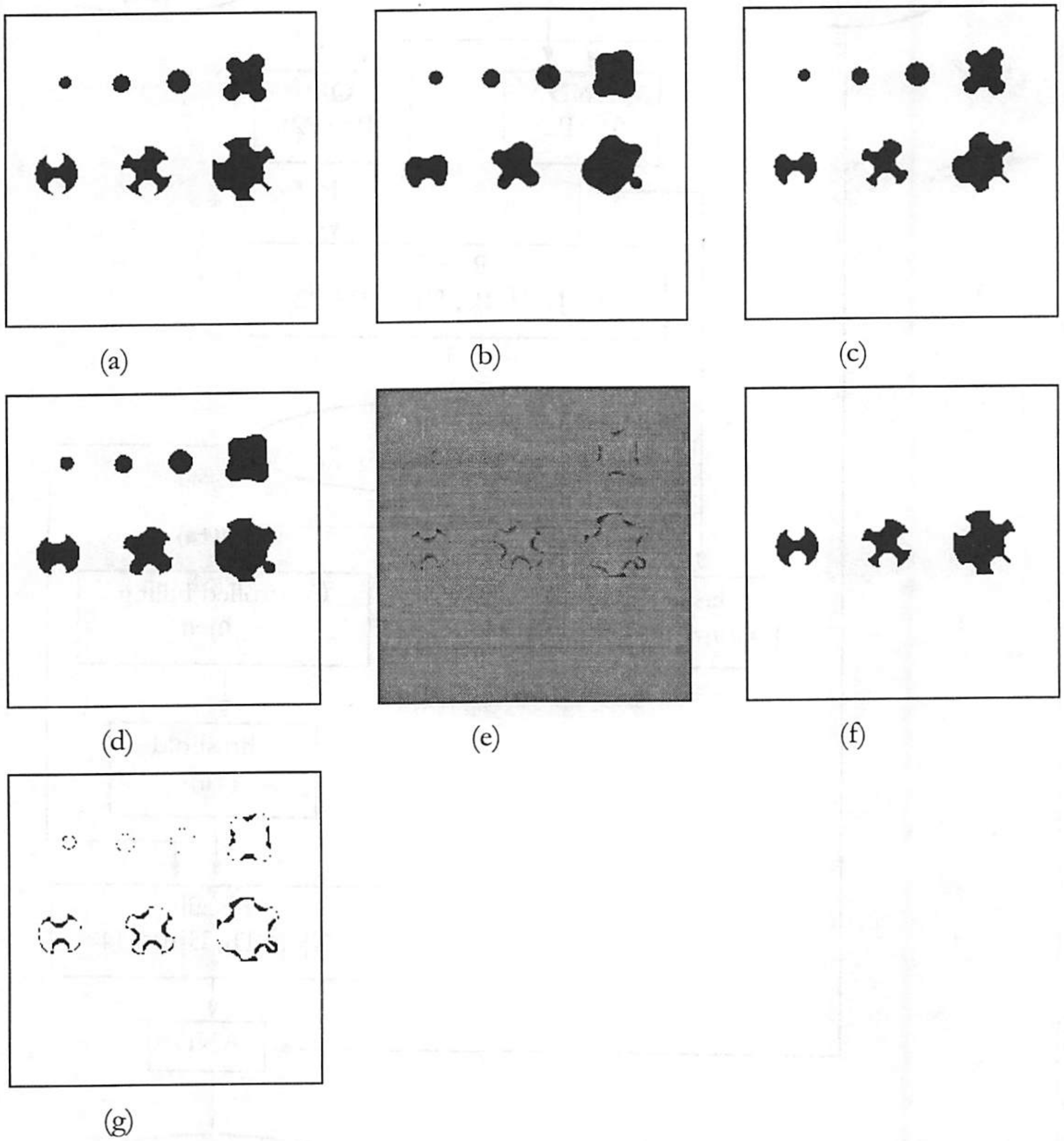


Figure 16 *Implementation of autowave metric on CNN. From the intersections of sets to be compared autowaves propagate and time is measured via constant current filling on a layer.*

Figure 17 shows intermediate steps of this process through an artificial example.



(a) (b) (c)

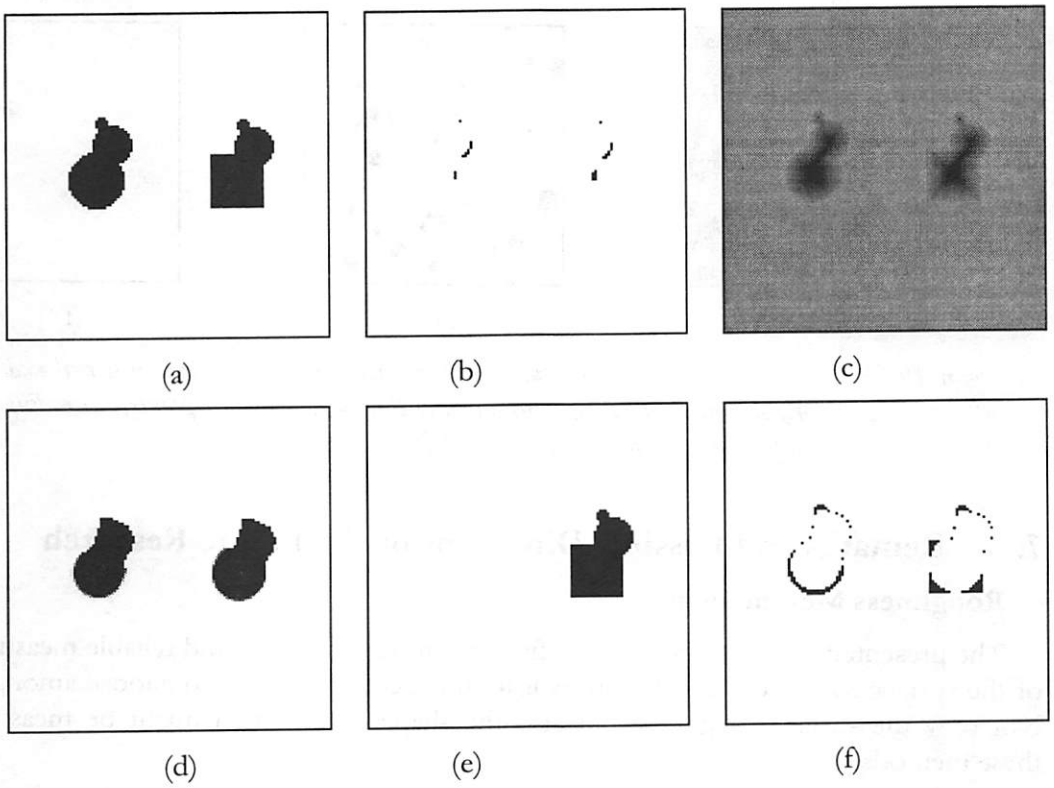
(d) (e) (f)

(g)

Figure 17 *Example demonstrates the process of the autowave metric implemented on CNN. (a) objects to be classified;, the first row contains bubbles or bubble groups and the second row contains debris particles (b) bubble models; (c) intersection of (a) and (b) i.e. AND operation, this contains the trigger points for autowave; (d) OR operation between (a) and (b) and containing only those set unions where objects and bubble models are partially overlapped, this is used as a mask (M4); (e) result of Controlled Filling, at the boundaries the cell levels are proportional with the time was needed to occupy set unions by autowaves; (f) after thresholding (e) the objects are recalled and considered as debris particles; (g) if we simply applied XOR between objects and bubble models the fourth object in the first row would be considered as debris particle.*

6. Examples

Here we present some results of applying this algorithm to real images. The Figure 19 shows consecutive steps of bubble - debris classification algorithm. The gray-scale input picture contains only one debris particle the remaining objects are all bubbles. Although some bubbles are connected together the algorithm can recognize that these are not debris particles. However, further examinations are necessary to tune the algorithm to have desired robustness in order to implement on a real CNN chip and obtain the requirement of very low false alarm rate.



(a) (b) (c)

(d) (e) (f)

Figure 18 *Consecutive steps of the bubble-debris classification algorithm on an artificial example. The object on the left contains three ideal overlapping bubbles forming a group and the object on the right has two ideal bubbles and a square, say, a debris particle forming a group - the method should distinguish them. For the first case the autowave distance from the "dynamic bubble model" should be approximately zero and nonzero for the second). The steps are (a) objects to be classified, (b) center points, (c) radius intensity map, (d) bubble models, (e) detected debris particle, (f) Hamming distance between original objects (a) and bubble models (d)*

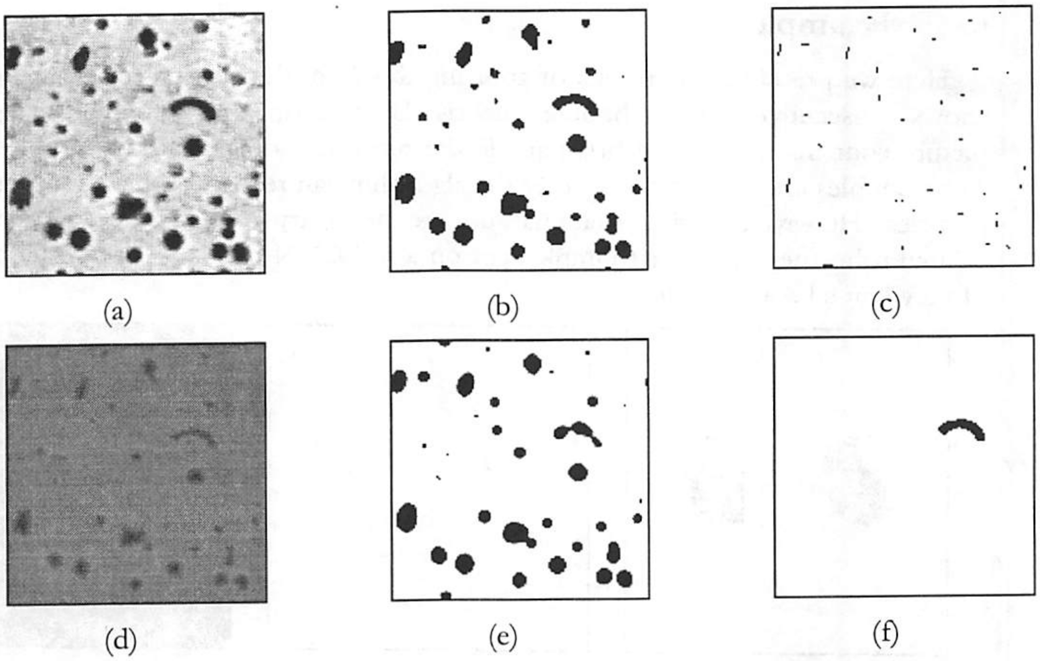


Figure 19 *Consecutive steps of the bubble-debris classification algorithm on a real example. (a) original gray-scale image, (b) adaptive threshold, (c) center points, (d) radius intensity map, (e) bubble models, (f) detected debris particle*

7. Remarks and Possible Directions of the Future Research

Roughness Measurement

The presented methods are only the first step toward a robust and reliable measurement of the surface roughness. Further investigation is needed in order to choose among them. Not only the surface roughness but also the shape deformation might be measured by these methods.

Another possibility to measure surface roughness is the use of fractal dimensions. With proper simplification this can be implemented on CNN.

These characteristics could be the input of a fuzzy algorithm which might be necessary to design in a more elaborate approach to the classification problem.

Center Point Detection

The masked erosion is simple but finds more points than necessary (the center points are not properly localized). It should be improved in order to have less center points. Better bubble models might be constructed if autowaves start propagating from a fewer number of center points.

Radii Map

The construction of a Radii Map is simple but does not provide perfect control for driving the autowaves. The constructed bubble models are similar to the original objects instead of being circles. The proper "smoothing" of the Radii Map using a diffusion process can be a possible approach to improve the model. At a cost of loosing the simplicity a more complex method can also be used to measure radii information.

Bubble Models

The most important task is to grow perfect circles around the center points. The critical part are the number of starting points and their locations. Another question which has a significant effect on the generated models is the measure of the object's size. We can use the largest circle which lies totally in the object, the smallest one which covers totally the object, or a circle between them. The difference between the largest fitting or the smallest covering circle might also help in classification.

Autowave Metrics versus Hamming Distance

Although the autowave metrics is more tolerant to shift and noise than the Hamming distance calculation it can not be clearly stated that for bubble-debris classification the autowave distance provides significantly better results. This is not surprising since there are no real shift errors (registration errors) and disturbing noise effects. The noise is filtered out at the front-end gray-scale to black-and-white conversion and the shift error caused during the generation of bubble models is negligible. The reason why the autowave metrics is proposed to solve this classification problem is the following. Computation of the autowave distance is more favorable for CNN than the Hamming distance from implementation point of view. To compute the weighted Hamming distance needed for classification the number of detected points should be normalized with reference to the object's area. This means that we cannot define a global threshold level and each object should be separately examined. The implemented autowave metric measures differences between objects and their models converting the CNN transient length to a corresponding gray-scale level. In this case a global threshold level can be used for all objects in the classification step.

8. Conclusions

We have presented a possible solution of the bubble-debris classification problem where the major requirements are real-time processing with a very low false alarm rate for miss-classified bubbles. Initial experiments and calculations indicate that the CNN technology can fulfill these requirements and give an efficient solution. The proposed algorithm based on the autowave metric exploits the advantages of the CNN architecture for a high speed implementation. It should be noted that further examinations are necessary to test the sensitivity and robustness of the synthesized algorithm.

9. Acknowledgment

The support of the Hungarian State Eötvös Fellowship is highly acknowledged. This work was also supported by the grant of the ONR (No. N00014-89-J-1402) and the grant of the NSF (No. INT-9413186).

10. References

[1] L. O. Chua and L. Yang, "Cellular Neural Networks: Theory," *IEEE Trans. Circuits Syst.*, Vol. 36, pp. 1257- 1272, 1988.

[2] L. O. Chua and T. Roska, "The CNN Paradigm," *IEEE Trans. Circuits Syst.*, Part I, Vol. 40, pp. 147-156, 1993.

[3] T. Roska and L. O. Chua, "The CNN Universal Machine: An Analogic Analog Computer," *IEEE Trans. Circuits Syst.*, Part II, Vol. 40, pp. 163-173, 1993.

[4] T. Kozek, K. R. Crounse, T. Roska, and L. O. Chua, "Multi-Scale Image Analysis on the CNN Universal Machine", *CNNA'96, Fourth IEEE International Workshop on Cellular Neural Networks and their Application*, pp. 69-74, Seville, Spain, June 24-26, 1996.

[5] Á. Zarándy, A. Stoffels, T. Roska, F. Werblin, and L. O. Chua, "Implementation of Binary and Gray-Scale Mathematical Morphology on the CNN Universal Machine", *Memo No. UCB-ERL*, Univ. of Cal. Berkeley, 96/19, 1996.

[6] Á. Zarándy, A. Stoffels, T. Roska, and L. O. Chua, "Morphological Operators on the CNN Universal Machine", *CNNA'96, Fourth IEEE International Workshop on Cellular Neural Networks and their Application*, pp. 151-156, Seville, Spain, June 24-26, 1996.

[7] V. Krinsky, V. Biktashev, and N. Efimov, "Autowaves principles for parallel image processing", *Physica D*, Vol. 49, pp. 247-253, 1991.

[8] V. Biktashev, V. Krinsky, and H. Haken, "A wave approach to pattern recognition (with application to optical character recognition)", *International Journal of Bifurcation and Chaos*, Vol. 4, No. 1, pp. 193-207, 1994.

[9] H. Haken, *Synergetics*, Springer, Berlin, 1978.

[10] V. Krinsky, ed., *Self-Organization. Autowaves and Structures Far from Equilibrium. Synergetics*, Vol. 28, Springer, Berlin, 1984

[11] L. O. Chua, M. Hasler, G. S. Moschytz, and J. Neirynck, "Autonomous Cellular Neural Networks: A Unified Paradigm for Pattern Formation and Active Wave Propagation", *IEEE Trans. Circuits Syst.*, Part I, Vol. 42, No. 10, pp. 559-577, 1995.

[12] V. Perez-Munzuri, V. Perez-Villar, and L. O. Chua, "Autowaves for image processing on a two-dimensional CNN array of excitable nonlinear circuits: flat and wrinkled labyrinths", *IEEE Trans. Circuits Syst.*, Vol. 40, pp. 174-181, Mar. 1993.

[13] T. Roska, L. O. Chua, D. Wolf, T. Kozek, R. Tetzlaff, and F. Puffer, "Simulating Nonlinear Waves and Partial Differential Equations via CNN-Part I: Basic Techniques", *IEEE Trans. Circuits Syst.*, Part I, Vol. 42, No. 10, pp. 807-815, 1995.

[14] T. Kozek, L. O. Chua, T. Roska, D. Wolf, R. Tetzlaff, F. Puffer and K. Lotz, "Simulating Nonlinear Waves and Partial Differential Equations via CNN-Part II: Typical Examples", *IEEE Trans. Circuits Syst.*, Part I, Vol. 42, No. 10, pp. 816-820, 1995.

[15] D. P. Huttenlocher, G. A. Klanderman, W. Rucklidge, "Comparing Images Using the Hausdorff Distance", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 15, No 9, pp. 850-863, September, 1993.

[16] Cs. Rekeczky, A. Schultz, I. Szatmari, T. Roska, and L. O. Chua, "Image Segmentation and Edge Detection via Constrained Diffusion and Adaptive Morphology: a CNN Approach to Bubble/Debris Image Enhancement", *Memorandum UCB/ERL M97/96*, Electronic Research Laboratory, University of California at Berkeley, December 1997.