

Copyright © 1996, 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.

**IMPLEMENTATION OF BINARY AND GRAY-SCALE
MATHEMATICAL MORPHOLOGY ON THE CNN
UNIVERSAL MACHINE**

by

Akos, Zarándy, André Stoffels, Tamás Roska,
and Leon O. Chua

Memorandum No. UCB/ERL M96/19

3 May 1996

**IMPLEMENTATION OF BINARY AND GRAY-SCALE
MATHEMATICAL MORPHOLOGY ON THE CNN
UNIVERSAL MACHINE**

by

Akos, Zarándy, André Stoffels, Tamás Roska,
and Leon O. Chua

Memorandum No. UCB/ERL M96/19

3 May 1996

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

IMPLEMENTATION OF BINARY AND GRAY-SCALE MATHEMATICAL MORPHOLOGY ON THE CNN UNIVERSAL MACHINE

Ákos Zarándy*, André Stoffels†, Tamás Roska*, and Leon O. Chua‡

ABSTRACT

The Cellular Neural/Nonlinear Network (CNN) paradigm [1,2,3] became a successful framework for many spatio-temporal problems. The CNN Universal Machine [4] enabled the algorithmic use of such spatio-temporal CNN instructions [5]. It excelled especially in computationally expensive linear and nonlinear image processing tasks. Among them, the emerging discipline of Mathematical Morphology offers a new geometrical approach to image understanding and processing. This paper introduces different analogic realizations of Binary and Gray-Scale Mathematical Morphology on the CNN Universal Machine (CNUM). Alternatively, a CNN Morphology Engine can be built for processing morphology operators with the highest possible speed on silicon.

1 INTRODUCTION

A distinct discipline within image analysis, mathematical morphology [6] studies the application of some basic geometrical operators to different problems in the field of image analysis. Mathematical morphology has been used to describe objects by considering them as subsets of a Euclidean space, which results in emphasis on their shapes, their volumes and their textures, as well as on their luminosity and color at each point. Morphological operators can reduce bodies to their essentials and so enable their description and detection in a much easier way.

In a well defined sense, the mathematical morphological operators erosion and dilation are generic. A morphological operator can be implemented by the iterative use of them (like the AND, OR, NOT in the Boolean logic) hence, the fact that CNN templates can implement erosion and dilation means that all morphological operators can be implemented by the CNUM.

We consider first the case of two-valued (binary) images to introduce the basic definitions of binary mathematical morphology (Section 2), and show their CNN implementation (Section 3). Then in Section 4, we present the definition of the basic gray-scale morphological operations. In Section 5, we present two implementation methods using different CNN structures: one with a nonlinear A template, and a more effective method with a nonlinear D template.

* Computer and Automation Institute of the Hungarian Academy of Science, Budapest, Hungary, and Nonlinear Electronics Laboratory of the University of California, Berkeley, USA, E-Mail: akos@fred.eecs.berkeley.edu, roska@fred.eecs.berkeley.edu

† Chair of Electrical Engineering and Computer Systems of the RWTH Aachen, University of Technology, Aachen, Germany, and Nonlinear Electronics Laboratory of the University of California, Berkeley, USA, E-Mail: stoffels@fred.eecs.berkeley.edu

‡ Nonlinear Electronics Laboratory of the University of California, Berkeley, USA, E-Mail: chua@fred.eecs.berkeley.edu

2 BINARY MATHEMATICAL MORPHOLOGY

The two main operations of the mathematical morphology are the *erosion* and *dilation* of an image. Erosion of an image P by a structuring element set S is defined by either of the following two equivalent definitions [6]:

$$\mathcal{E}(P, S) = \{ \xi : S_\xi \subseteq P \} \quad \text{or} \quad \mathcal{E}(P, S) = \bigcap_{\xi \in S} P_{-\xi} \quad (1)$$

Here the vector symbol ξ describes translations in the two-dimensional image domain. Thus S_ξ is the structuring element set S translated by ξ . The first definition says that the *fit* function (Figure 1) is equivalent to the erosion operation [6], while the second shows a Boolean algebraic approach. The *fit* function is true in a pixel location, if the structuring element set (S) is translated to the particular pixel position (S_ξ) “fits” entirely in an object on image P (Figure 1). The other basic operation of binary mathematical morphology is dilation. Dilation of an image P by a structuring element set S is defined by either of the following two equivalent definitions [6]:

$$\mathcal{D}(P, S) = \{ \xi : \hat{S}_\xi \cap P \neq \emptyset \} \quad \text{or} \quad \mathcal{D}(P, S) = \bigcup_{\xi \in S} P_\xi \quad (2)$$

Here the hat-operator is the reflection-operator, and \emptyset stands for the empty set. The first definition says that the dilation is equivalent to the *hit* function (Figure 1) of P and \hat{S} [6], while the second shows a Boolean algebraic approach. The *hit* function is true in a pixel location, if the reflected structuring element set (\hat{S}) is translated to the particular pixel position (\hat{S}_ξ) has at least one common (“hitting”) pixel with an object on image P (Figure 1). In our paper we do not devote too much attention to opening and closing, because those operators can be implemented on CNN the same way like in other machines, namely, applying erosion and dilation operators one after the other. Figure 1 shows an example for erosion and dilation and explains *hit* and *fit* functions.

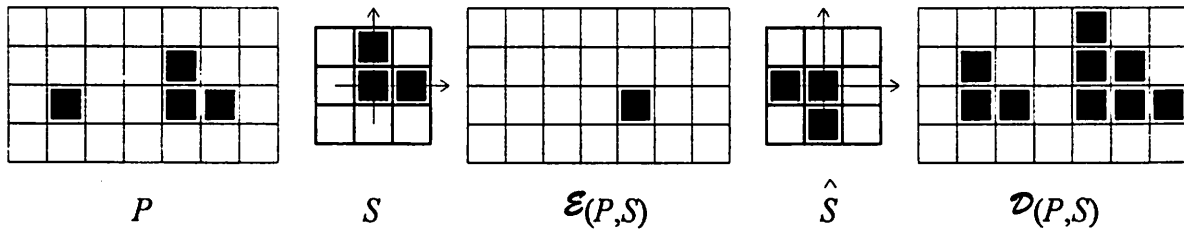


Figure 1. Example for the erosion and the dilation. Note that structuring element set S fits only in one position into the original image P (erosion image). The *hit* function for the P and the \hat{S} images was satisfied in 9 pixel positions (dilation image).

3 IMPLEMENTATION OF BINARY MATHEMATICAL MORPHOLOGY

In this paper we use the following CNN dynamics:

$$\dot{v}_{xij}(t) = -v_{xij}(t) + \sum_{kl \in N_r(ij)} \mathcal{A}(y_{ij}, y_{kl}) + \sum_{kl \in N_r(ij)} \mathcal{B}(u_{ij}, u_{kl}) + \sum_{kl \in N_r(ij)} \mathcal{C}(x_{ij}, x_{kl}) + \sum_{kl \in N_r(ij)} \mathcal{D}(x_{ij}, u_{kl}) + I$$

$$y_{ij} = f(x_{ij}). \quad (3)$$

where: \mathcal{A} , \mathcal{B} , \mathcal{C} , and \mathcal{D} are nonlinear two input single output functions, and f is the sigmoid function [1].

The basic operations of the mathematical morphology are *erosion*, *dilation*, *opening*, and *closing*. These operations are defined on two images, one being the operand, the other the structuring element set. In the CNN implementation, the former is the input, while the function (the templates) itself depends on the latter. Due to the limited template size of the VLSI implementation (3×3 in the current existing chips [8,9]), the implementation method drastically depends on the extension of the structuring element set. If it does not exceed the size of the CNN template the *dilation* and *erosion* operators can be implemented with a single CNN template (Section 3.1). If it does, an analogic CNN algorithm can implement the operator (Section 3.2).

In the followings black locations of a binary image are associated with pixels having logic true value, and are represented by +1 in the CNN dynamic space. Analogously, white locations are associated with pixels having logic false value, and represented by -1.

3.1 SINGLE TEMPLATE IMPLEMENTATION OF BINARY MORPHOLOGICAL FILTERS

The *erosion* and *dilation* operators of binary mathematical morphology can be considered as a convolution followed by a thresholding [10]. For example, if we consider the same structuring element set as shown in our examples (Figure 1), we can calculate the morphological filters in the following ways:

- *Erosion* is equivalent to convolution by the matrix $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ followed by a thresholding at level 2.

Note, that this matrix is the direct mapping of the structuring element set S in Figure. In general, due to the binary nature of the convolution kernel and the ± 1 value representation of image P , the result of the convolution can be any value from the discrete set: $\{-n, -n+2, \dots, n-2, n\}$, where n is the number of the 1s in the convolution kernel. In those pixel positions, where the *hit* function is true, the result of the convolution is exactly n . Hence, thresholding with level $(n-1)$ delivers the result of the *erosion* operator.

- *Dilation* is equivalent to convolution by the matrix $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ followed by a thresholding at level -

2. Note, that this matrix is the direct mapping of the structuring element set \hat{S} in Figure 1. In general, in those pixel positions, where the *hit* function is true, the result of the convolution is greater than $-n$. Hence, thresholding with level $(1-n)$ delivers the result of the *dilation* operator.

The convolution followed by a thresholding method can be implemented directly on the CNNUM with a single template. The template synthesis method is the following:

The **A** template matrix contains only one non-zero element, a 1 in the center position both the *erosion* and *dilation* cases. In the case of *erosion* the structuring element set S should be directly mapped to the **B** template and the I value should be equal to $(1-n)$, where n is the number of 1s in the **B** template matrix. In the case of *dilation* the reflected structuring element set \hat{S} should be directly mapped to the **B** template and the I value should be equal to $(n-1)$, where n is the number of 1s in the **B** template matrix. When calculating the operator, the image should be put to the input of the CNN, and the initial condition is zero everywhere. Equation (4) shows the synthesized erosion and equation (5) the dilation templates for the structuring element set given in Figure 1.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}, I = -2 \quad (4)$$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, I = 2 \quad (5)$$

Notes:

1. If we select current value I between $(1-n)$ and $(n-1)$, we can implement the *weak mathematical morphology*, which is closely related to *fuzzy sets* [10]. The concept of the hit, fit, or miss transform is too strict in many cases. Usually, an “almost hit” or a “nearly miss” statement is more useful. Especially pattern recognition uses this fuzzy description of the images.
2. When calculating the single template erosion and dilation operations, the transient settles in 1τ time, where τ is the time constant of the analog CNN ($\tau=100-300$ ns depending on the implementation [8,9]).

3.2 ON THE STABILITY OF THE TEMPLATES

In this chapter, we give a proof, that the erosion template (4) is stable, and works correctly. In the same way, the same property of the dilation template (5) can be proved [17].

3.2.1 GLOBAL TASK (EROSION)

Given: static binary image P and a binary pattern structure S within $S_r(ij)$ defined by black pixels in Figure 1.

Input: $U(t)=P$

Initial State: $X(0)= \text{zero}$

Output: $Y(t) \Rightarrow Y(\infty)$ (binary image) containing those black pixels of which are inside of P relative to the pattern S . This means: a pixel p is inside P if the pattern S placed on p is also inside P .

3.2.2 LOCAL RULES

| $u_{ij}(0)$ | | $y_{ij}(\infty)$ |
|--------------------|--------|---|
| 1. arbitrary pixel | black, | if the pixel is inside of P relative to the pattern S . |
| 2. arbitrary pixel | white, | if the pixel is not inside of P relative to the pattern S . |

3.2.3 EXAMPLE

See example in Figure 1.

3.2.3 MATHEMATICAL ANALYSIS

Consider the Driving Point (DP) plot shown in Figure 2. If all the (three) black pixels of S are inside P , then

$$w = w_3 = I + \sum_{kl} B_{kl} u_{kl} = -2 + n - (3 - n) = 1 \text{ where } n = 3 \quad (6)$$

On this DP plot, starting from 0 $x_{ij} \rightarrow Q_3 > 1$, hence $y_{ij} \rightarrow 1$.

If at least one black pixel of S is outside P then

$$w = w_n = I + \sum_{kl} B_{kl} u_{kl} = -2 + n - (3 - n) < -1 \text{ where } n = \{0, 1, 2\} \quad (7)$$

On the DP plot, starting from 0 $x_{ij} \rightarrow Q_n < -1$ ($n = \{0, 1, 2\}$, the number of the black pixels of S , which is inside P), hence $y_{ij} \rightarrow -1$.

Q_0, Q_1, Q_2, Q_3 , are globally stable equilibrium points.

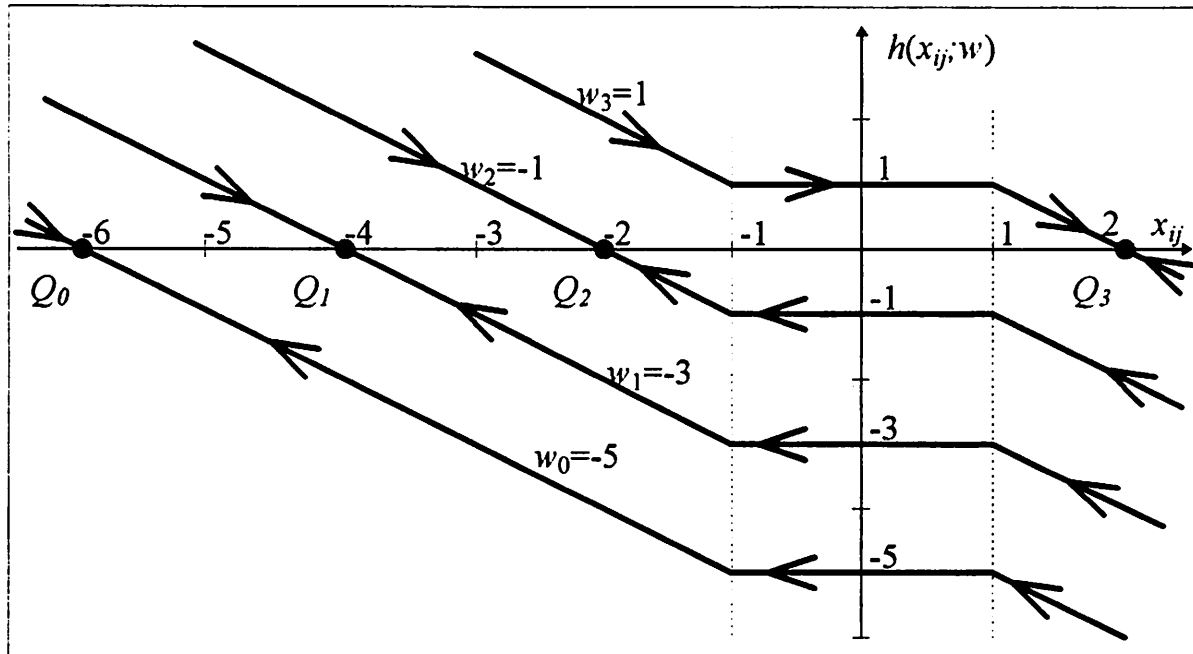


Figure 2. The Driving Point plot of the CNN with the erosion template.

4 GRAY-SCALE MATHEMATICAL MORPHOLOGY

Gray-scale mathematical morphology is a more recent vintage than its binary version. We present its definitions using the one-dimensional continuous (in space) signals for the sake of an improved understanding of the main issues and the two-dimensional discrete image space to visualize its applications to image processing. Nevertheless it has to be emphasized that the general body of the mathematical gray-scale morphology is independent of any domain dimensionality.

In contrast to binary morphology where images are interpreted as sets $\{-1, +1\}$ and operations are therefore discussed in terms of set theory, gray-scale morphology deals with operations using certain subsets of the set of real numbers. In the analog gray-scale space of the CNNUM, for instance, input-output values vary within $[-1, +1]$. Therefore the previously applied set operations such as intersections or unions, are now replaced by the minimum and the maximum operators.

In the case of gray-scale images, an image P is generally defined as a distribution of gray-scale values $P = P(\xi)$ where $\xi \in D_P$, i.e., ξ denotes a pixel location in the 2D *spatial image domain* D_P . It has to be emphasized that the same rules are applied to the structuring element set S , which is characterized by an arbitrary gray level distribution $S = S(\xi)$ where $\xi \in D_S$. This leads to another major difference to the binary case, which is the existence of a third translational degree of freedom in the gray-scale image space. Besides the spatial translation in the image domain described by ξ in D_P , we can now think about a *vertical translation* or *offset* ζ along the gray-scale intensity axis. The spatial translation of an image P by ξ has been referred to as P_ξ . Then the vertical translation of an image P by ζ is defined by $P + \zeta$, which equals to a pixel-wise addition of the image with the gray level described by ζ . The same is true for the structuring element set (S) as well.

Gray-scale *erosion* of an image P by a structuring element set S is described by the following point-wise term:

$$\mathcal{E}(P, S)(\xi) = \sup\{\zeta: S_\xi + \zeta \preccurlyeq P\}, \quad (8)$$

where the relation \preccurlyeq stands for *beneath*. An image S is *beneath* an image P if $D_S \subseteq D_P$, and for $\forall \xi \in D_S$, $S(\xi) \leq P(\xi)$. Geometrically, to find the erosion of a signal by a structuring element set at a point ξ , slide the structuring element set spatially so that its origin is located at ξ and then find the largest ζ , for the structuring element set to be lifted along the intensity axis with, and still be *beneath* the signal. This effect is illustrated in Figure 3 in the 1D signal space for a semicircular and a flat structuring element set.

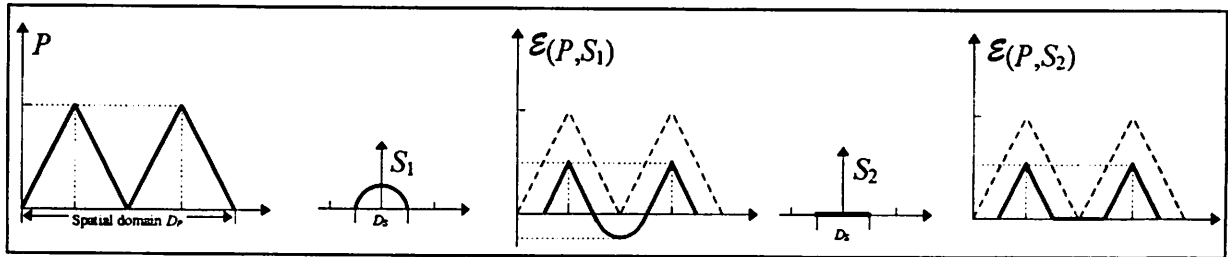


Figure 3. Erosion of a signal P featuring two succeeding triangles by a semicircular (S_1) and a flat (S_2) structuring element sets. In both cases the spatial domains of the structuring

element sets are equal to half of the spatial domain of a triangle. Therefore the domains of $\mathcal{E}(P, S_1)$ and $\mathcal{E}(P, S_2)$ are identically reduced when compared to the original image P . $\mathcal{E}(P, S_1)$ is found by tracing out the graph with placing S_1 always to the highest position as it is still *beneath* P . $\mathcal{E}(P, S_2)$ follows strictly the same idea.

Morphological gray-scale *dilation* of an image P by a structuring element set S can be defined by this point-wise definition:

$$\mathcal{D}(P, S)(\xi) = \inf\{\zeta: \hat{S}_\xi + \zeta \geq P\} \quad (9)$$

where $\hat{S}(\xi) = -S(-\xi)$ is the reflected (to the origin of S) and inverted (the values themselves) structuring element set (see in Figure 4), and \geq stands for *above*. An image S is *above* an image P if $D_S \cap D_P$ is not an empty set, and for all $\xi \in D_S \cap D_P$, $S(\xi) \geq P(\xi)$. Geometrically, to find the dilation of a signal by a structuring element set at a point ξ , slide the inverted and reflected structuring element set spatially so that its origin is located at ξ and then find the smallest ζ , for the structuring element set to be lifted along the intensity axis with, and still be *above* the signal. This effect is illustrated in Figure 4 in the 1D signal space for a semicircular and a flat structuring element set.

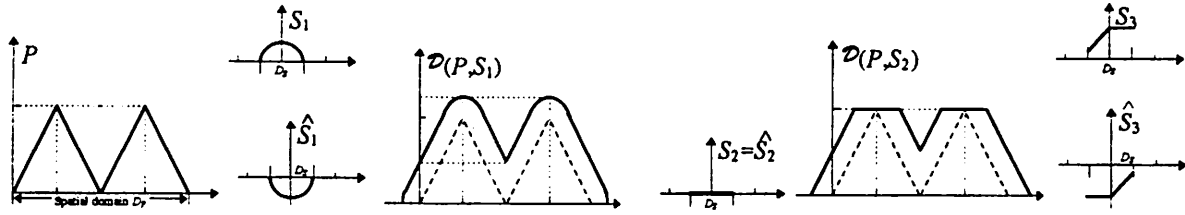


Figure 4. Dilation of a signal P by S_1 and S_2 structuring element sets. Dilating of the graphs of the signals parallels the extension of the signals' spatial domains. In both cases the spatial domains of the structuring element sets are equal to half of the spatial domain of a triangle. Therefore the domains of $\mathcal{D}(P, S_1)$ and $\mathcal{D}(P, S_2)$ are identically enlarged when compared to the original image P . $\mathcal{D}(P, S_1)$ is found by tracing out the graph with placing \hat{S}_1 always to the lowest position as it is still *above* P . $\mathcal{D}(P, S_2)$ follows strictly the same idea. S_3 structuring element set is introduced to show the effect of the reflection and inversion for a general (non-symmetric) signal.

Figure 5 shows an example for the dilation and the erosion applied to gray-scale images.



Figure 5. Example for the effect of the gray-scale erosion and dilation on gray-scale images. (a) is the original frame, (b) is the erosion, (c) is the dilation image. In the erosion image the brighter regions are emphasized and widened, while in the dilation image the darker regions are emphasized and widened.

5 IMPLEMENTATION OF GRAY-SCALE MATHEMATICAL MORPHOLOGY

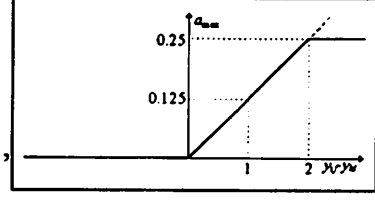
Here we show two different implementation methods for the gray-scale morphology. First, we show a multi-step method, which uses non-linear **A** templates. Then, we show a single template method, which uses non-linear **D** templates. The reason while we show both is, that so far only simple CNN chips with **A** and **B** templates were realized, however the second method exploits better the resources of a CNN Universal Machine with **D** template.

5.1 IMPLEMENTATION OF GRAY-SCALE MATHEMATICAL MORPHOLOGY ON A CNN UNIVERSAL MACHINE WITH NON-LINEAR **A** TEMPLATE

The main computational task occurring while tackling gray-scale morphological operations is the query for local maxima or minima in a certain search area. This area is equivalent to the size of the used structuring element. The task of finding local extrema can be considered as restricting the search for a global extremum to a prescribed search range only. The term of an extremum refers here to both, maxima and minima, because their computation is very similar and needs no specific differentiation. Global tasks in a given image can be generally resolved by designing a feedback template **A** and enabling the transient to propagate during a certain time to reach every part of the image. In our case, [12] already presented a cloning template that computes the global maximum.

The limitation to an operation of only local native now implies the use of *fixed-state-maps*. These maps mark in the binary case with their black pixels those cells that remain "frozen" when the transient is running. An appropriate fixed state mask allows for the computation of local extrema. The pattern of the mask depends on the structure of the chosen **A** template. Generally we define two different shapes of templates (or structuring elements): The *two-dimensional $A_{max,C}$ -template* and the *$A_{max,D}$ -template* as given in (10, 11) in case of the maximum operation. The processing of the minima requires the inverted and reflected nonlinear function.

$$A_{max,C} = \begin{bmatrix} 0 & a_{max} & 0 \\ a_{max} & 1 & a_{max} \\ 0 & a_{max} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 0, \quad (10)$$



$$A_{max,D} = \begin{bmatrix} a_{max} & 0 & a_{max} \\ 0 & 1 & 0 \\ a_{max} & 0 & a_{max} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 0, \quad (11)$$

Both operations use two different fixed-state patterns to solve the problem. In case of the $A_{max,D}/A_{max,C}$ template, the resulting mask has a line/checkerboard pattern. The mask's dimensions equal those of the original image. Looking at the general structure of an $A_{max,D}/A_{max,C}$ template, the underlying logic is easy to understand. Because the cell's center coincides with a white pixel in the fixed-state-map its volume can change. At the same time it should be guaranteed that all other neighbored cells that can influence the operation remain fixed. This is important because the calculated extreme would otherwise be global and not local. In fact, the main purpose of the fixed-state-map is to forestall the propagation of the A template. If one follows this logic it is easy to see that the operation can be calculated in two stages using a binary fixed-state-map and its inverse as shown in Figure 6.

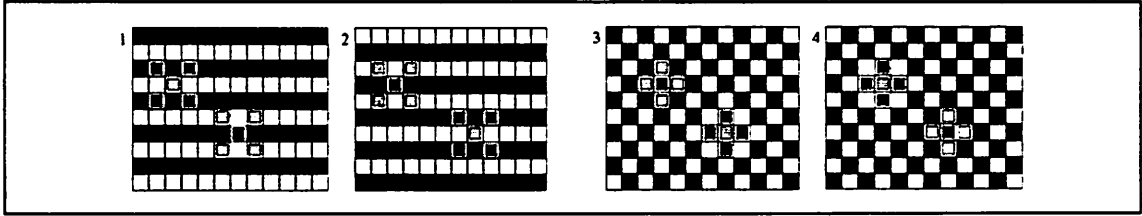


Figure 6. Picture 1(3) shows in the upper left side the case of $A_{max,D}/A_{max,C}$ template whose center cell coincides with a white pixel (light gray), while the connected neighbors are all fixed (dark gray). This cell is able to adapt the result of the extreme calculation. The opposite case is the $A_{max,D}/A_{max,C}$ template in the lower right part of picture 1(3). Here the center is fixed (dark gray) and keeps the state value independently from the operation to be applied. Picture 2(4) points out that a simple inversion of the fixed-state-map also inverses the impact on the center cells. A superposition of both cases simulates finally the effect of a local extreme operation on the whole array.

The combination of both $A_{max,C}$ and $A_{max,D}$ templates leads to the more complex case of a square structuring element. In the case of a morphological gray-scale dilation, for instance, we compute independently the supreme with both $A_{max,C}$ and $A_{max,D}$ templates and calculate finally the pixel-wise maximum of both output frames. The gray-scale dilation by a square structuring element S is then given by:

$$\mathcal{D}(P,S)(\xi) = \max\{\mathcal{D}(P,S_C)(\xi), \mathcal{D}(P,S_D)(\xi)\} \quad (12)$$

where ξ denotes a pixel in the *image domain* D_P . Following the same remarks, the gray-scale erosion is defined by:

$$\mathcal{E}(P, S)(\xi) = \min\{\mathcal{E}(P, S_C)(\xi), \mathcal{E}(P, S_D)(\xi)\}. \quad (13)$$

The flat structuring elements we dealt with until now were entirely described by the two-dimensional $A_{max,C}$ and $A_{max,D}$ templates. We can now enlarge the body of this theory to include the full range of *symmetrical three-dimensional structuring elements*. Generally a symmetrical

structuring element shows the following form:

| | | |
|-----|-----|-----|
| d | c | d |
| c | e | c |
| d | c | d |

. Here c , d , and e are intensity levels

fitting within the amplitude range used in the CNN context, this means: $c, d, e \in [-1, +1]$. The previous case that was subject of our previous considerations is included into this model by setting $c = d = e = 0$. The general symmetric case of the gray-scale *erosion* is then given by:

$$\mathcal{E}(P, S)(\xi) = \min\{\mathcal{E}(P+c, S_C)(\xi), \mathcal{E}(P+d, S_D)(\xi), P(\xi)+e\}. \quad (14)$$

where $P+c$ denotes the addition of the offset c to the amplitude distribution of the object P . Here it has to be mentioned that the "new" object $P+c$ has still the same size and location as the original object P . Only its intensity levels are shifted by c . The same considerations are of course valid for all coefficients c , d , and e . This model also includes the case of the general symmetric gray-scale *dilation* described by:

$$\mathcal{D}(P, S)(\xi) = \max\{\mathcal{D}(P+c, S_C)(\xi), \mathcal{D}(P+d, S_D)(\xi), P(\xi)+e\}, \quad (15)$$

5.2 IMPLEMENTATION OF SINGLE TEMPLATE EROSION AND DILATION ON THE CNN UNIVERSAL MACHINE

In this section first, we show the implementation of gray-scale morphological operators from the mathematical aspect. Then, we show the details of the single template implementation of the *erosion* and *dilation* operators.

Let us analyze the point-wise definition of the *erosion* as given in (8). In the discrete lattice this equation leads to a set of ordering relations at every ξ position. Figure 7 shows a simple case, where the derivation of the ordering relations is easily understandable. A discrete gray-scale 1D signal (Figure 7a) and a discrete structuring element set are given in Figure 7b. Equation (8) expresses that in position ξ the largest possible ζ elevation of S_ξ is sought where $(S_\xi + \zeta)$ is still *beneath* P . Figure 7c shows the vertically translated S_ξ with ζ (square shaped discrete points) *beneath* P (round shaped discrete points).

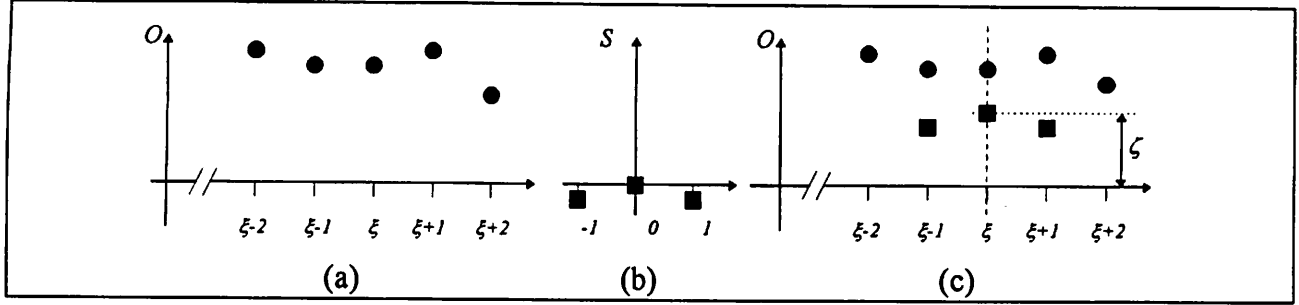


Figure 7. (a): discrete 1D function P . (b): Discrete structuring element set S . (c): S_ξ is raised with ζ , but still beneath P .

The largest ζ at position ξ in the example of Figure 7c is equivalent to the largest solution of the following ordering relation system:

$$\begin{aligned} S(-1) + \zeta &\leq P(\xi-1) \\ S(0) + \zeta &\leq P(\xi) \\ S(1) + \zeta &\leq P(\xi+1) \end{aligned} \tag{16}$$

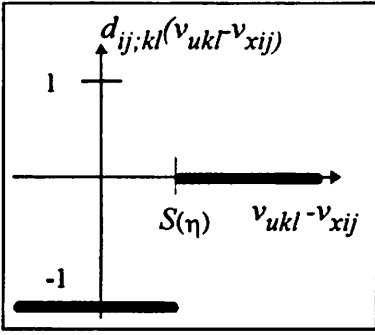
Note:

1. Each relation stands for an element of the structuring element set. The terms on the left-hand-side in relation (16) are illustrated by the height of the squares in the chart in Figure 7c, whereas the terms on the right-hand-side are represented by the height of the discs.
2. We did not take advantage of the 1D signal, hence (8) with arbitrary finite discrete structuring element set leads to a system of relations in every ξ position on the same way. The general form of the relations is:

$$S(\eta) + \zeta \leq P(\xi + \eta), \quad \text{where } \eta \in D_S \text{ and } \xi \in D_P \tag{17}$$

Now, we show a parallel method for finding the largest solution of the relation system (17) in a particular ξ position. First, an appropriate small ζ value is selected, which satisfies all the relations in (17). Then, ζ is gradually increased and all the relations are continuously checked in parallel. When the two sides of a relation become equal, the ζ value will freeze (stop increasing). This frozen ζ value is the largest solution of the relation system (17). For solving (17) at every ξ position, we have to do the same method for every ξ . It seems to be very complex, however, it can be directly implemented on a CNN array with nonlinear \mathbf{D} template (3). (Similar \mathbf{D} template was used to implement median and rank operators [16] on the CNN Universal Machine.) If the structuring element set S does not exceed the template size, the relation system (17) can be implemented with a single template on this CNN structure. The idea of the implementation is as follows:

The ζ values are represented by the state of the CNN, and P is represented by the input of the CNN. The ζ_0 values are chosen as -1 (the lowest initial condition value of the CNN gray-scale dynamic). We create a template, which starts increasing all the ζ values simultaneously, and freezes each of them individually at the last moment when it still satisfies the relation system. The template is the following:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} d_{i,j,i-1,j+1} & d_{i,j,i,j+1} & d_{i,j,i+1,j+1} \\ d_{i,j,i-1,j} & d_{i,j,i,j} & d_{i,j,i+1,j} \\ d_{i,j,i-1,j-1} & d_{i,j,i,j-1} & d_{i,j,i+1,j-1} \end{bmatrix}, I = 1$$


where $S(\eta)$ is the real value of the structuring element set (S) at point $\eta=(k,l)$, if not $d_{ij;kl} \equiv 0$. This template finds the largest solution of the relation system (19), which is equivalent with (17).

$$S(\eta) \leq P(\xi + \eta) - \zeta, \text{ where } \eta \in D_S \text{ and } \xi \in D_P \quad (19)$$

At the very beginning, all $P(\xi + \eta) - \zeta_0$ values are larger than $S(\eta)$, hence the contribution of the \mathbf{D} template is zero at every cell. Due to the positive I , all state values will start to increase, i.e., the value on the right-hand-side of equation (19) will decrease. (This term is represented by the horizontal axis of the graph bound to (18). When a particular raising state value ζ reaches the point, where

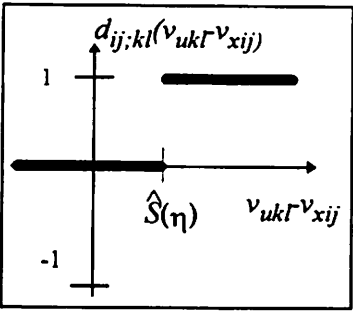
$$S(\eta) = P(\xi + \eta) - \zeta \quad (20)$$

the contribution of the \mathbf{D} template becomes -1, which eliminates the effect of the I term, hence freezes the particular state value at ζ . After the state values in every cells are frozen, the output of the CNN will show the result of the erosion. Due to the +1 current value in the I term, the changing speed of the state value is equal to $1/\tau$. Hence, the longest cell transient settles within 2τ , because the dynamic space of the CNN is limited $[-1, +1]$. If there are some ζ_0 , which does not satisfy the relations, P and S should be linearly rescaled as to allow $\zeta_0 = -1$ to satisfy all the relations.

The definition of the dilation (9) also leads to an ordering relation system at every point ξ . The only difference is, that the direction of the relation signal is the opposite.

$$\hat{S}(\eta) + \zeta \geq P(\xi + \eta), \text{ where } \eta \in D_S \text{ and } \xi \in D_P \quad (21)$$

Similarly to the *erosion* operator, the *dilation* operator with a 3×3 structuring element set can be implemented with a single template on the modified CNN structure. The differences are that here the initial condition of the CNN ζ_0 is +1 (instead of -1) and after starting the transient, all the values start decreasing (instead of increasing). The template which implements the 3×3 dilation is the following:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} d_{i,j,t-1,j+1} & d_{i,j,t,j+1} & d_{i,j,t+1,j+1} \\ d_{i,j,t-1,j} & d_{i,j,t,j} & d_{i,j,t+1,j} \\ d_{i,j,t-1,j-1} & d_{i,j,t,j-1} & d_{i,j,t+1,j-1} \end{bmatrix}, I = -1 \quad (22)$$


where: $\hat{S}(\eta)$ is the real value of the reflected and inverted structuring element set (\hat{S}) at point $\eta=(k,l)$. If it is not defined in a η point $d_{ij;kl} \equiv 0$.

The template works exactly the same way as the erosion template does. The transient settles in 2τ , and the output of the CNN shows the result of the dilation operator.

6 IMPLEMENTATION MATHEMATICAL MORPHOLOGY WITH LARGE STRUCTURING ELEMENT SET

Morphological operators with large structuring element sets can also be implemented on the CNN Universal Machine with analogic CNN algorithms. The algorithms are similar in both, the binary and in the gray-scale cases. Without going into details, we list the main steps of the implementation here:

1. Separate the large structuring element set into smaller ones, which ones can be implemented with a single template.
2. Calculate the operations with all the small structuring element subsets, and shift the results to the central position.
3. Apply AND (binary erosion), OR (binary dilation), MIN (grayscale erosion), MAX (grayscale dilation) operators to the subresults.

7 CONCLUSION

Here we presented three different methods to implement the functions of the binary and gray-scale mathematical morphology for different structuring element sets on the CNUM. As it turned out the *dilation* and *erosion* operators can be implemented with a single CNN template, however these operations are extremely computational hungry tasks of a digital image processing unit. The significantly decreased computational complexity allows extremely fast morphological image processing. Considering the additional resources provided by the CNUM, one can state that a CNN based mathematical morphology lends itself to time critical applications. Among those, mathematical morphology is useful in the emerging field of object-oriented video compression techniques [7][13][18]. The CNN matches all requirements. i.e., programmability, flexibility, and high performances, that are asked for a real-time “morphology engine” [14].

ACKNOWLEDGMENT

The supports of the *Hungarian State Eötvös Fellowship*, the *Otto-Junker-Stiftung*, Aachen, the *Konrad-Adenauer-Stiftung*, Sankt Augustin, Germany, and the grant *INT 90-01336* of the National Science Foundation in cooperation with the Hungarian Academy of Sciences, are greatly acknowledged.

REFERENCES

- [1] L.O. Chua and L. Yang, "Cellular Neural Networks: Theory", *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, October 1988, pp. 1257-1272
- [2] L.O. Chua and L. Yang, "Cellular Neural Networks: Applications", *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, October 1988, pp. 1273-1290
- [3] L.O. Chua and T. Roska, "The CNN Paradigm", *IEEE Transactions on Circuits and Systems - I*, vol. 40, no. 3, March 1993, pp. 147-156
- [4] T. Roska and L.O. Chua, "The CNN Universal Machine: An Analogic Array Computer", *IEEE Transactions on Circuits and Systems - II*, vol. 40, March 1993, pp. 163-173
- [5] T. Roska, "Analogic Algorithms Running on the CNN Universal Machine", *Proceedings of the 3rd International IEEE Workshop on Cellular Neural Networks and their Application (CNNA-94)*, Rome, pp. 3-8, December 1994.
- [6] C.R. Giardina and E.R. Dougherty, *Morphological Methods in Image Processing and Signal Processing*, Prentice Hall, Englewood Cliffs (NJ), 1988
- [7] P. Salembier, L. Torres, F. Meyer, and C. Gu, "Region-Based Video Coding Using Mathematical Morphology", *Proceedings of the IEEE*, vol. 83, no. 6, June 1995, pp. 843-857
- [8] R. Dominguez-Castro, S. Espejo, A. Rodriguez-Vazquez, R. Carmona, "A CNN Universal Chip in CMOS Technology", (CNNA-94), Rome, December 1994, pp. 91-96
- [9] H. Harrer, J.A. Nossek, T. Roska, and L.O. Chua, "A Current-mode DTCNN Universal Chip", *Proceedings of the International IEEE Symposium on Circuits and Systems*, 1994, pp. 135-138
- [10] J.E. Mazille, "Mathematical Morphology and Convolutions", *Journal of Microscopy*, Vol. 156, Pt 1, October 1989, pp. 3-13
- [11] R.M. Haralick, S.R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, July 1987, pp. 532-550
- [12] L.O. Chua, T. Roska, T. Kozek, and Á. Zarándy, "The CNN Paradigm - A Short Tutorial", *Cellular Neural Networks*, T. Roska and J. Vandewalle, editors, John Wiley & Sons, New York, 1993, pp. 1-14
- [13] M. Kunt, A. Ikonopoulou, and M. Kocher, "Second Generation Image Coding Techniques", *Proceedings of the IEEE*, vol. 73, no. 4, April 1985, pp. 549-575

- [14] P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI Architectures for Video Compression - A Survey", *Proceedings of the IEEE*, vol. 83, no. 2, February 1995, pp. 220-246
- [15] "Analogic CNN Program Library", edited by T. Roska, and L. Kék, Comp. and Auto. Ins. of the Hung. Acad. of Sci. DNS-7-1995, Budapest
- [16] Cs. Rekeczky, T. Roska, A. Ushida, "CNN based self-adjusting nonlinear filters" in preparation
- [17] L.O. Chua and T. Roska, "Cellular Neural Network Foundations and Primer", Lecture notes, U.C. Berkeley, 1996.
- [18] A. Stoffels, T. Roska, and L.O. Chua, "On Object-Oriented Video Coding using the CNN Universal Machine", *IEEE Transactions on Circuits and Systems - I*, accepted for publication in 1996.