Peter Pazmany Catholic University, Budapest

Cellular Wave Computers

Analogic Cellular PDE Machines

Course Material

Peter Pazmany Catholic University, Budapest

Contents

- *Analogic PDE Machines*
- *PDE formulation of reaction-diffusion processes*
- *Implementation of active contour methods in target detection and tracking*
- *Implementation of wave-metrics in object comparison and classification*
- *PDE formulation of contrast enhancement with denoising*
- *Implementation of contrast enhancement methods with denoising*
- *Summary*

Peter Pazmany Catholic University, Budapest

Preliminaries:

- Cellular Automata are an alternative to (rather than an approximation of) differential equations in modeling physics /Toffoli, 1984/
- There are cellular structures described by ODEs for which a limiting PDE with the same qualitative properties does not exist /Keener,1987/
- The spectrum of the volume of any physical region is discrete /Rovelli-Smolin, 1995/
- PDEs are merely idealizations of cellular structures described by coupled ODEs or local rules /Chua, 1997/

Peter Pazmany Catholic University, Budapest

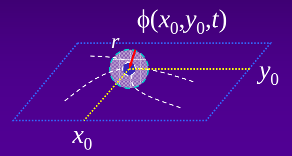
Local and Global PDEs

Local PDE:

- the evolution of the system at any location depends only on local properties

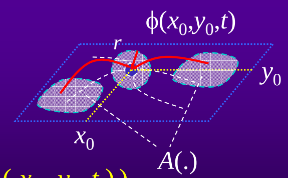
$$\dot{\phi}_r(x,y,t) = \Gamma(\phi_r(x,y,t))$$

$$\{\ \forall\, (x_0,y_0,t) \mid r \to 0 \ \Rightarrow \Gamma = \Gamma(x_0,y_0,t)\ \}$$



Global PDE (Sapiro et. al):

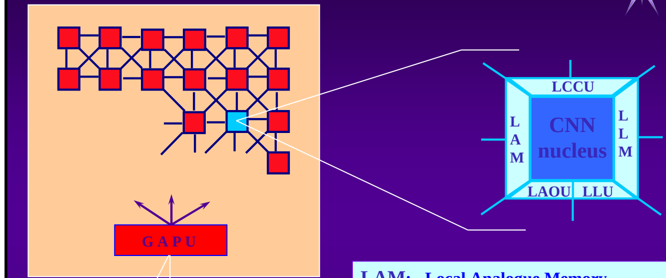
- the evolution of the system at some location depends also on global properties



$$\dot{\phi}_r(x,y,t) = \Gamma(\phi_r(x,y,t),\ A(x,y,t))$$

$$\{\ \exists\, (x_0,y_0,t) \mid r \to 0 \ \Rightarrow \Gamma = \Gamma(x_0,y_0,t,A(.)) \wedge A(.) \neq 0\ \}$$

CNN Universal Machine (CNN-UM)



GAPU: Global Analogic Programming Unit

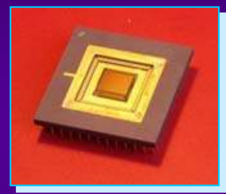
LAM: Local Analogue Memory
LLM: Local Logic Memory
LCCU: Local Communication and Control Unit
LAOU: Local Analogue Output Unit
LLU: Local Logic Unit

APR: Analog Instruction Register
LPR: Logic Program Register
SCR: Switch Configuration Register
GACU: Global Analogic Control Unit

$[A_1\ B_1\ z_1], [A_2\ B_2\ z_2], \ldots$

"Analogic (analog+logic) algorithm"

The workhorse of the experiments: a 64x64 cell CNN-UM visual microprocessor (ACE4K)



Chip design:
IMSE-CNM, Sevilla, Spain

Hardware-software environment:
Analogical and Neural Computing Systems Laboratory
MTA-SZTAKI, Hungary

PDE formulation of reaction-diffusion processes

Reaction-diffusion type nonlinear PDE:

$$\frac{\partial \phi(x,y,t)}{\partial t} - div(c(\phi(x,y,t))\, grad(\phi(x,y,t))) = \phi_0(x,y,t_0) + \Im(\phi(x,y,t))$$

Sub-classes ($c(.) = c_0$):

I. $\phi_0(.) = 0 \ \wedge\ \Im(.) = 0$ ⟹ *linear diffusion equation*

II. $\phi_0(.) \neq 0 \ \wedge\ \Im(.) = 0$ ⟹ *constrained linear diffusion equation*

III. $\phi_0(.) = 0 \ \wedge\ \Im(.) \neq 0$ ⟹ *nonlinear trigger-wave equation*

IV. $\phi_0(.) \neq 0 \ \wedge\ \Im(.) \neq 0$ ⟹ *constrained nonlinear trigger-wave equation*

Deriving coupled ODEs from PDEs I.

Reaction-diffusion type nonlinear ODE:

$$\frac{d\phi_{ij}(t)}{dt} = g(\phi_{ij}(t)) - \phi_{ij}(t) + \frac{c_1}{4}(\phi_{i-1j}(t) + \phi_{i+1j}(t) + \phi_{ij-1}(t) + \phi_{ij+1}(t)) + z_{ij}$$

$$\phi_{ij}(t) = f(x_{ij}(t)) \ \wedge\ g(.) = c_0 f(.) \ \wedge\ z_{ij} = z_0 + \sum_{kl \in N_1} b_{kl}\phi_{kl}(t_0)$$

Sub-classes:

I. $z_{ij} = 0 \ \wedge\ f(\phi) = \phi$ ⟹ *linear diffusion equation*

II. $z_{ij} \neq 0 \ \wedge\ f(\phi) = \phi$ ⟹ *constrained linear diffusion equation*

III. $z_{ij} = 0 \ \wedge\ f(\phi) = sigm(\phi)$ ⟹ *nonlinear trigger-wave equation*

IV. $z_{ij} \neq 0 \ \wedge\ f(\phi) = sigm(\phi)$ ⟹ *constrained nonlinear trigger-wave equation*

All implementable on the ACE4K 64x64 CNN-UM chip !

Deriving coupled ODEs from PDEs II.

Reaction-diffusion type nonlinear ODE:

$$\frac{d\phi_{ij}(t)}{dt} = g(\phi_{ij}(t)) - \phi_{ij}(t) + \frac{c_1}{4}(\phi_{i-1j}(t) + \phi_{i+1j}(t) + \phi_{ij-1}(t) + \phi_{ij+1}(t)) + z_{ij}$$

$$\phi_{ij}(t) = f(x_{ij}(t)) \;\wedge\; g(.) = c_0 f(.) \;\wedge\; z_{ij} = z_0 + \sum_{kl \in N_1} b_{kl}\phi_{kl}(t_0)$$

Templates (symmetric-isotropic class):

$$A = \begin{bmatrix} 0 & c_1 & 0 \\ c_1 & c_0 & c_1 \\ 0 & c_1 & 0 \end{bmatrix}; \qquad B = \begin{bmatrix} b_2 & b_1 & b_2 \\ b_1 & b_0 & b_1 \\ b_2 & b_1 & b_2 \end{bmatrix}; \qquad z_0$$

$c_0 = 0 \wedge c_1 > 0 \rightarrow diffusion;$ $\qquad c_0 > c_1 > 0 \rightarrow trigger - waves$

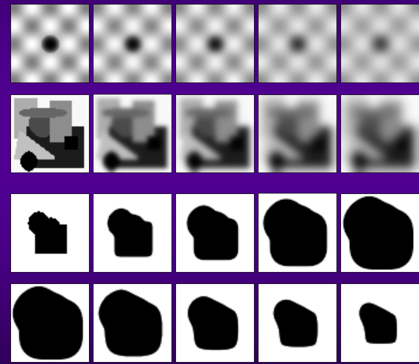
Computing with diffusion and waves - I.

/ derived from reaction-diffusion systems /

Examples:

Linear diffusion

Trigger-waves



Linear Diffusion Scale Space - I.

Diffusion (heat equation): $\frac{d}{dt}x_t(v_1, v_2) = \nabla^2 x_t(v_1, v_2)$

2D spatial Fourier-tr: $\frac{d}{dt}\tilde{X}_t(\Omega_1, \Omega_2) = -(\Omega_1^2 + \Omega_2^2)\tilde{X}_t(\Omega_1, \Omega_2)$

since: $\tilde{\nabla}^2(\Omega_1, \Omega_2) = -(\Omega_1^2 + \Omega_2^2)$

Solution in freq. domain: $\tilde{X}_t(\Omega_1, \Omega_2) = e^{-t(\Omega_1^2 + \Omega_2^2)}\tilde{X}_0(\Omega_1, \Omega_2)$

Solution in time domain: $x_t(v_1, v_2) = e^{-(v_1^2 + v_2^2)/2t} * x_0(v_1, v_2)$

The output at time t is the convolution of the input by a Gaussian kernel function (the width of G is parametrized by time) $\qquad \sigma = \sqrt{t}$

Linear Diffusion Scale Space - II.

Autonomous CNN: $\frac{d}{dt}x_t(n_1, n_2) = \Lambda(n_1, n_2) * x_t(n_1, n_2)$

where: $A(n_1, n_2) = \begin{cases} A_{0,0} - 1 & n_1, n_2 \in (0,0) \\ A_{n_1,n_2} & n_1, n_2 \in S_r \\ 0 & otherwise \end{cases}$

Discrete Space FT: $\frac{d}{dt}\tilde{X}_t(\omega_1, \omega_2) = \tilde{\Lambda}(\omega_1, \omega_2)\tilde{X}_t(\omega_1, \omega_2)$

Solution: $\tilde{X}_t(\omega_1, \omega_2) = e^{t\tilde{\Lambda}(\omega_1, \omega_2)}\tilde{X}_0(\omega_1, \omega_2)$

$\tilde{\Lambda}(\omega_1, \omega_2)$ approximates the diffusion operator (constraints: discrete space and limited size of the operator)

Example: $(\tilde{\Lambda}(0,0) = 0)$

$$\Lambda = \begin{bmatrix} 0.5 & 1.0 & 0.5 \\ 1.0 & -6.0 & 1.0 \\ 0.5 & 1.0 & 0.5 \end{bmatrix} \Rightarrow \tilde{\Lambda} \approx -2(\omega_1^2 + \omega_2^2)$$

Laplace!

Scale: $s \propto \sqrt{t}$

Peter Pazmany Catholic University, Budapest

Multi-Scale Representations - I.

Gauss Decomposition

Base level (original image): $\tilde{G}^{(0)}(\omega_1,\omega_2)=\tilde{X}_0(\omega_1,\omega_2)$

p-th level (p=1,2,..P): $\tilde{G}^{(p)}(\omega_1,\omega_2)=\tilde{\Gamma}_{t_p}(\omega_1,\omega_2)\tilde{G}^{(0)}$

method:

$\forall\sqrt{\omega_1^2+\omega_2^2}\ge\pi : \tilde{\Gamma}_t(\omega_1,\omega_2)\le K \qquad \tilde{\Gamma}_t(\omega_1,\omega_2)=e^{t\tilde{\Lambda}(\omega_1,\omega_2)}$

$t_p=\frac{\ln K}{\tilde{\Lambda}(\pi/2^p,0)}$ **- to determine the stopping times for a given K and level p**

Peter Pazmany Catholic University, Budapest

Multi-Scale Representations - II.

Laplace Decomposition

- DoG decomposition (by recursion from G decomposition)

p-th level (p=1,2,..P): $L^{(p)}=\tilde{G}^{(p)}-\tilde{G}^{(p+1)}$

- LoG decomposition (by applying the appr. Laplacian operator to each level of G decomposition)

Peter Pazmany Catholic University, Budapest

Multi-Scale Representations - III.

Wavelet Representation

Wavelet transformation (1D): $\hat{f}(\xi,\vartheta)=|\xi|^{-1}\int f(v)\,h(\frac{v-\vartheta}{\xi})\,dv$

Wavelet basis: $h_{\xi,\vartheta}(v)=|\xi|^{-1}h(\frac{v-\vartheta}{\xi})$

- h() satisfies an admissible and regularity condition

CNN wavelets derived from a low-pass - high-pass approach:

$h_{t'}(\omega_1,\omega_2)=\tilde{\Gamma}_{t'}(\omega_1,\omega_2)(1-\tilde{\Gamma}_{t'}(\omega_1,\omega_2))$ **- band-pass!**

For a desired center frequency: (ω_1^c,ω_2^c) $t'=\frac{\ln(1/2)}{\tilde{\Lambda}(\omega_1^c,\omega_2^c)}$ **- transient length**

Peter Pazmany Catholic University, Budapest

Computing with diffusion and waves - II.

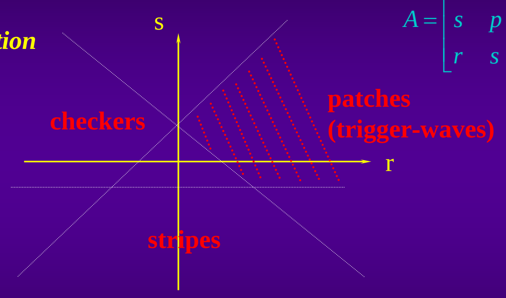
/ derived from reaction-diffusion systems /

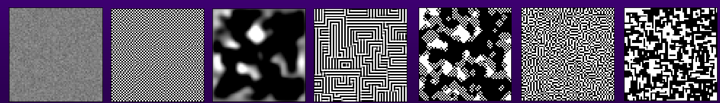
Examples:

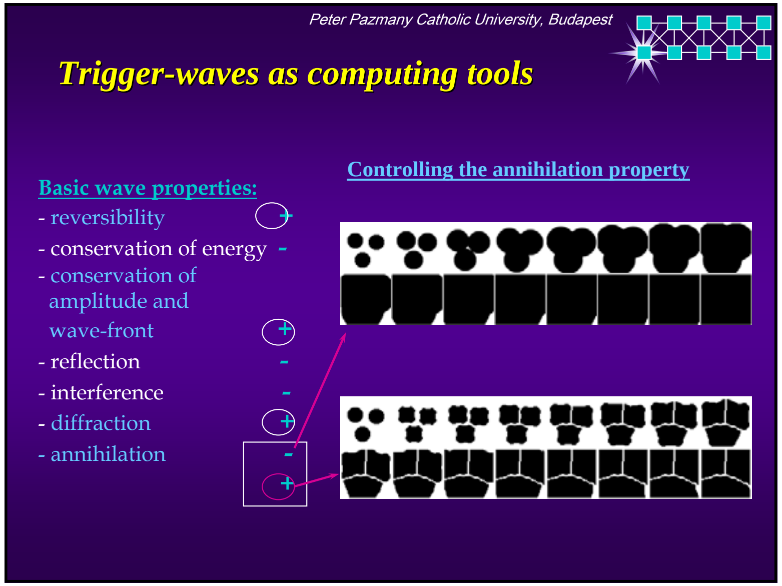
Pattern formation

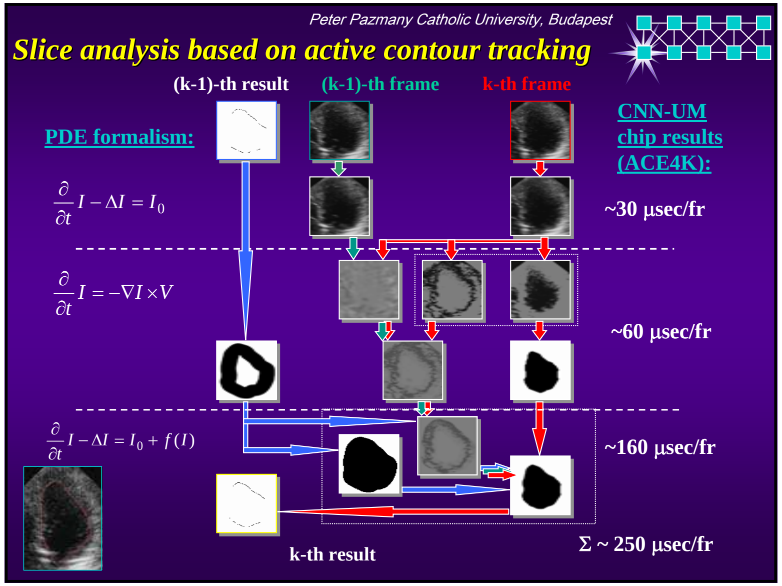
(p > 1)

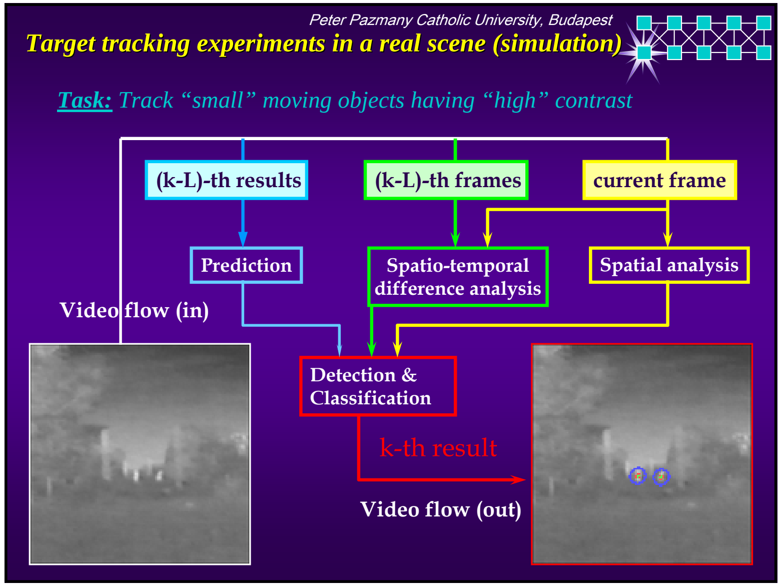
$A=\begin{bmatrix} r & s & r \\ s & p & s \\ r & s & r \end{bmatrix}$

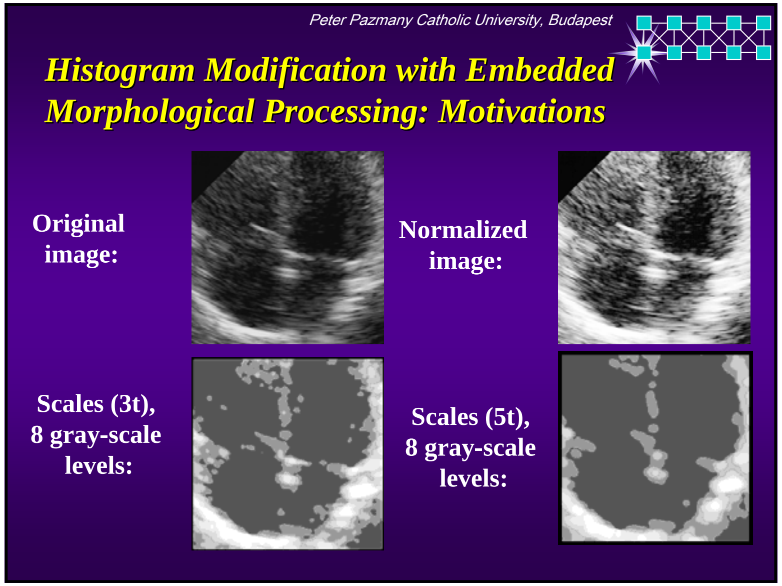





Peter Pazmany Catholic University, Budapest
Trigger-waves as computing tools
Basic wave properties:
- reversibility +
- conservation of energy -
- conservation of amplitude and wave-front +
- reflection -
- interference -
- diffraction +
- annihilation - +
Controlling the annihilation property


Peter Pazmany Catholic University, Budapest
Slice analysis based on active contour tracking
(k-1)-th result
(k-1)-th frame
k-th frame
PDE formalism:
CNN-UM chip results (ACE4K):
$\frac{\partial}{\partial t} I - \Delta I = I_0$
~30 µsec/fr
$\frac{\partial}{\partial t} I = -\nabla I \times V$
~60 µsec/fr
$\frac{\partial}{\partial t} I - \Delta I = I_0 + f(I)$
~160 µsec/fr
k-th result
Σ ~ 250 µsec/fr


Peter Pazmany Catholic University, Budapest
Target tracking experiments in a real scene (simulation)
Task: Track "small" moving objects having "high" contrast
(k-L)-th results
(k-L)-th frames
current frame
Prediction
Spatio-temporal difference analysis
Spatial analysis
Video flow (in)
Detection & Classification
k-th result
Video flow (out)


Peter Pazmany Catholic University, Budapest
Histogram Modification with Embedded Morphological Processing: Motivations
Original image:
Normalized image:
Scales (3t), 8 gray-scale levels:
Scales (5t), 8 gray-scale levels:

PDE Formulation of Histogram Modification (Contrast Enhancement) with Denoising - I.

Sapiro & Caselles:

$$\frac{\partial\phi(x,y,t)}{\partial t} = \alpha\kappa + (N^2 - H(\phi(x,y,t))) - A[(v,w):\phi(v,w,t) \geq \phi(x,y,t)]$$

where:

$$\kappa = div(grad(\|\phi\|))$$

Φ - *image intensity*
$\kappa(.)$ - *diffusion term*
H(.) – monotonic control function of histogram modification
A(.) - area (integral)
α,*N - constants*

$$A = \iint_\Omega (.)\, dxdy$$

PDE Formulation of Histogram Modification (Contrast Enhancement) with Denoising - II.

Steady-state solution (with α=0):

$$A[(v,w):\phi(v,w) \geq \phi(x,y)] = N^2 - H(\phi(x,y))$$

If $H = H_0\, \phi = (N^2/M)\phi$ then the result is the histogram equalized image!

Global asymptotic stability:

$$U(\Phi) = \frac{1}{2}\int(\Phi(X) - \frac{1}{2})\, dX - \frac{1}{4}\iint|\Phi(X) - \Phi(Z)|\, dX\, dZ$$

$U(\Phi)$ is a Lyapunov function!

PDE Formulation of Histogram Modification (Contrast Enhancement) with Denoising - III.

Derivation of nonlinear ODEs:

Case1 (global histogram equalization):

$$Let: \quad \alpha = 0; \quad H_0 = N^2/M; \quad N^2 = 1; \quad M = 1;$$

$$\Rightarrow \frac{d\phi_{ij}(t)}{dt} = -\phi_{ij}(t) + (1 - A_{ij})$$

Case2 (global histogram equalization with denoising):

$$Let: \quad \alpha = 1; \quad \kappa = div(grad(\|\phi\|)); \quad H_0 = N^2/M; \quad N^2 = 1; \quad M = 1;$$

$$\Rightarrow \frac{d\phi_{ij}(t)}{dt} = -\phi_{ij}(t) + \frac{1}{4}(\phi_{i-1j}(t) + \phi_{i+1j}(t) + \phi_{ij-1}(t) + \phi_{ij+1}(t)) + (1 - A_{ij})$$

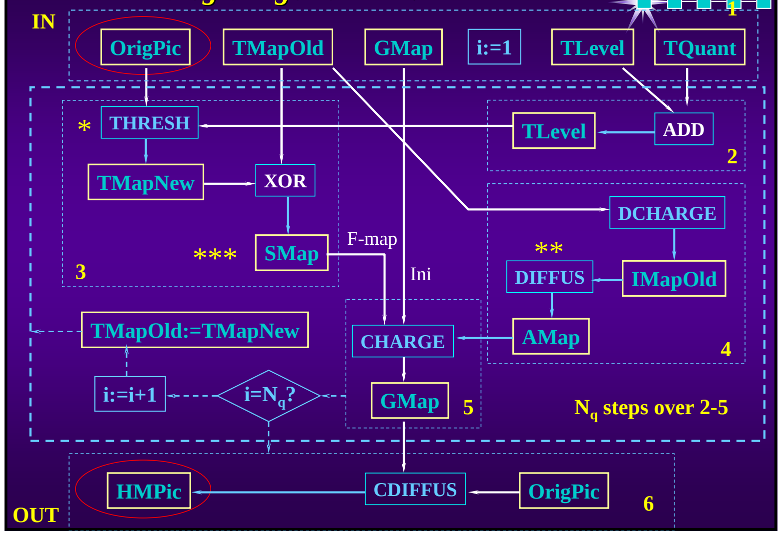
PDE Formulation of Histogram Modification (Contrast Enhancement) with Denoising - IV.

Computational aspects:

- The denoising term is a (nonlinear) diffusion equation formulation
- In both cases ***A_{ij} = const during the evolution***, therefore should be calculated only once!
- Though ***A_{ij}*** is the output of a global transformation it is possible to give an approximation based on purely local (analog & logic) operations. Exploiting the global dynamics of the local analog operations results in further desirable properties!

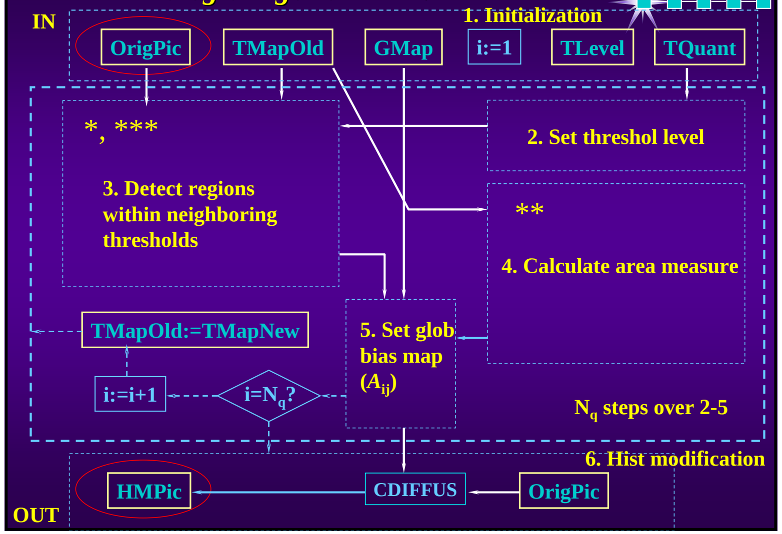
Peter Pazmany Catholic University, Budapest

HM Analogic Algorithm - I.



Peter Pazmany Catholic University, Budapest

HM Analogic Algorithm - II.



Peter Pazmany Catholic University, Budapest

HM Analogic Algorithm III.

Main processing stages:

1. Initialization (TLevel=1; TQuant=-2/N_q; TMapOld=-1; GMap=0;)
2. Set threshold level
3. Detect regions above threshold
4. Calculate area measure
5. Set Global Bias Map pixels
6. Modify image histogram with denoising

Alternative solution for block 2:

- Image read-out (TMapOld)
- Pixel number count (N_B)
- Image write-in (Amap := N_B /PicNum)

Adaptive processing (at *, ** and ***):

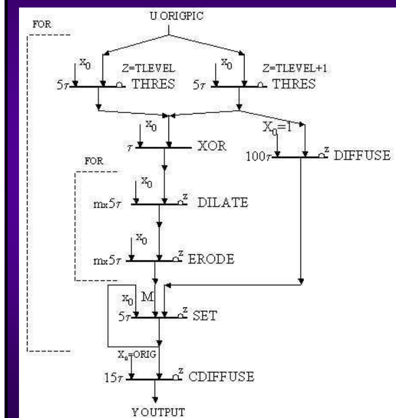
* Fuzzy decomposition instead of fixed-threshold decomposition (results in overlapping SMap images and sub-quantum level adaptation)

** Expand diffusion only to local areas instead of the entire image regions (over a circular area calculated from TMapNew and TMapOld)

*** Embedded morphology processing the level-set function SMap

Peter Pazmany Catholic University, Budapest

UMF description of the HM Algorithm



Algorithm 2:

$\Phi_{\lambda_0} = 0, \quad A = N^2$

$\Phi_{Norm} = Norm(\Phi)$

for $i = 1$ to q

$\Phi_{\lambda_i} = Thr(\Phi_{Norm}, \lambda_i)$

$\bar{\Phi}_{\lambda_i} = \Phi_{\lambda_i} \ xor \ \Phi_{\lambda_{i-1}}$

$\bar{\Phi}_{\lambda_i} = Erode(Dilate(\bar{\Phi}_{\lambda_i}, B)^{(n)}, B)^{(n)}$

$A_i = Area(\Phi_{\lambda_i})$

$v = A_i / A$

$\Phi_{Map} = SetLev(\bar{\Phi}_{\lambda_i}, v)$

$\Phi_{\lambda_{i-1}} = \Phi_{\lambda_i}$

end

$\Phi_{HM} = Diff(\Phi, \Phi_{Map})$

Implementation of the HM Algorithm I.

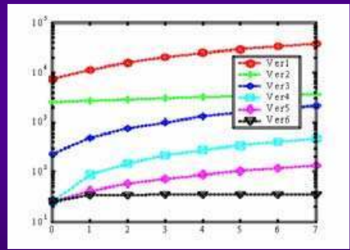
Different hardware-software configurations implemented:

- *Version 1:* MATCNN MATLAB Toolbox simulating all analog CNN dynamics µP: Pentium 1GHerz
- *Version 2:* Version 1 except for simulating through the CNN fixed-points with an optimized C-code
- *Version 3:* Version 2 except for the C-code contains the entire algorithm.
- *Version 4:* Aladdin Professional with an optimized C-code that contains the entire algorithm running on a DSP
- *Version 5:* Version 4 except for the morphology operation is optimized at the assembly level
- *Version 6:* Version 4 except for the morphology operation is optimized for the CNN-UM chip
- *Version 7*: in Aladdin Professional running the entire algorithm on a CNN-UM chip (ongoing). µP: Alcatel ACE4k

Implementation of the HM Algorithm II.

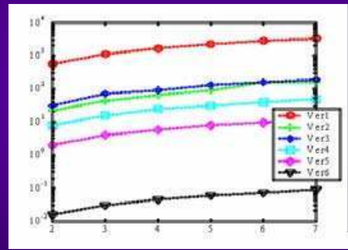
Execution time of Hmod algorithms (q=8)

Execution time [ms]



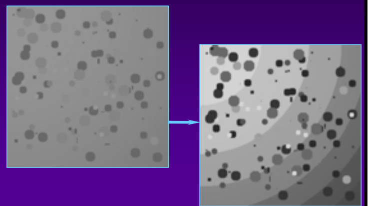
Morphological steps (num)

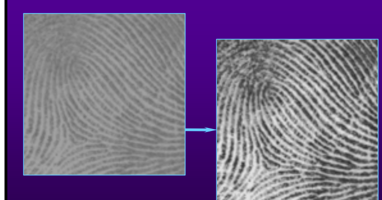
Execution time of binary morphology

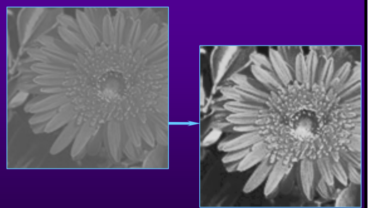


Morphological steps (num)

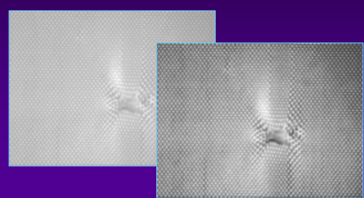
HM – examples I.

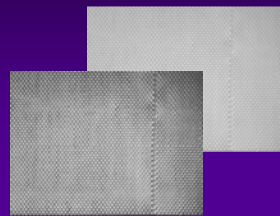


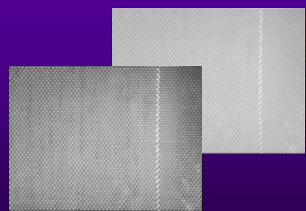


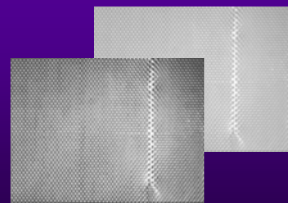


HM - examples II.









HM with Embedded Morphological Processing

Low contrast and noisy image (input):

Original image:

The number of morphological operation performed (0-4)

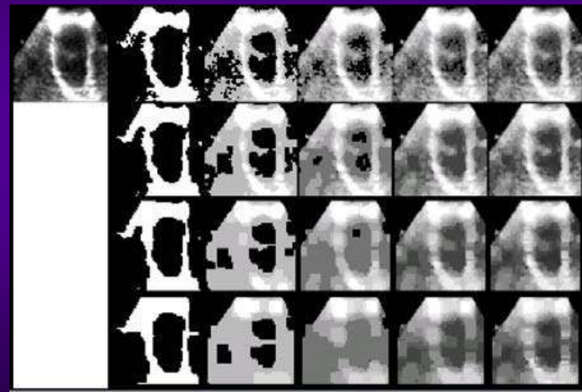
The number of equalization levels (32,16,8,4,2)

ACE4k chip experiments

The number of equalization levels (2,4,8,16,32)

Noisy input echo image:

The number of morphological operation performed (0-3)



Summary:

- *Important classes of local and global PDEs describing calculations via image flows can be well approximated by local processing (with global dynamics) on a locally connected nonlinear net described by nonlinear ODEs*

- *Diffusion and waves, generated and controlled on silicon, can be viewed as flexible computing tools in solving various image processing problems*

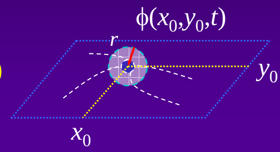
- *Analogic CNN algorithms - building on PDE formulation - have been designed for contour detection, tracking, object classification and global histogram equalization with simultaneous denoising not exceeding the complexity of the ACE4K CNN-UM chip*

This is not the end of the story!

Is the wave-equation realizable on silicon?

$$\frac{\partial^2 \phi_r}{\partial t^2}(x, y, t) = \Gamma(\phi_r(x, y, t))$$

$$\{\forall (x_0, y_0, t) \mid r \to 0 \Rightarrow \Gamma = \Gamma(x_0, y_0, t)\}$$



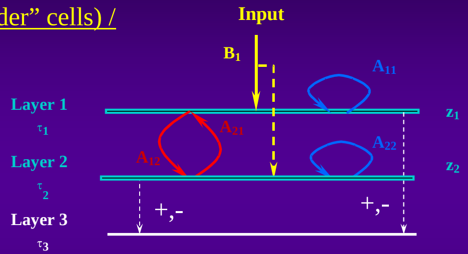
Things to come:

- *A 2nd order/two layer CNN described by nonlinear ODEs*

- *"Complex cell" CNN-UM chip* **(CICA1k - 2001)**

The architecture I.

Two-layer (or "2nd order" cells) / 3-layer model



Main requirements:

- core: two mutually coupled 1st order RC cells
- 21 linear synapses on 8-connected square-grid
- nearest neighbor interaction
- double time-scale property
- constant and zero flux boundary condition
- single input (1st L or 2nd L), separate initial states

Peter Pazmany Catholic University, Budapest

The architecture II.

System equations

$$C_1\dot{x}_{1,ij} = -x_{1,ij}/R_1 + \sum_{kl\in N_1} A_{11,kl}y_{1,kl} + a_{21}y_{2,ij} + b_0u_{ij} + z_1, \quad y_{1,ij} = f(x_{1,ij})$$

$$C_2\dot{x}_{2,ij} = -x_{2,ij}/R_2 + \sum_{kl\in N_1} A_{22,kl}y_{2,kl} + a_{12}y_{1,ij} + z_2, \qquad y_{2,ij} = f(x_{2,ij})$$

$$x_{3,ij} = f_a(x_{1,ij}, x_{2,ij})$$

$$\tau_1 = R_1C_1, \qquad \tau_2 = R_2C_2, 1\le i\le M, \quad 1\le j\le M, 0\le |u_{ij}|\le 1$$

$$Chua-Yang: 0\le |x_{ij}(0)|\le 1, \qquad Full-range: 0\le |x_{ij}(t\ge 0)|\le 1,$$

Template format

$$A_{11}-\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{1,4} & a_{1,5} & a_{1,6} \\ a_{1,7} & a_{1,8} & a_{1,9} \end{bmatrix}; \qquad A_{22}-\begin{bmatrix} a_{2,1} & a_{2,2} & a_{2,3} \\ a_{2,4} & a_{2,5} & a_{2,6} \\ a_{2,7} & a_{2,8} & a_{2,9} \end{bmatrix}; \qquad A_{12} = a_{12}; \qquad A_{21} = a_{21};$$

$$B_1 = b_0; \qquad z_1; \qquad z_2; \qquad \tau_1; \qquad \tau_2$$

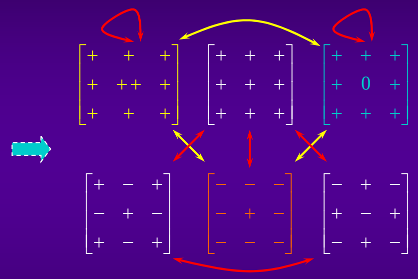
Peter Pazmany Catholic University, Budapest

Multi-layer templates

Configurations (determined by fixed A templates)

$$A_{11}-\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{1,4} & a_{1,5} & a_{1,6} \\ a_{1,7} & a_{1,8} & a_{1,9} \end{bmatrix}$$

$$A_{22}-\begin{bmatrix} a_{2,1} & a_{2,2} & a_{2,3} \\ a_{2,4} & a_{2,5} & a_{2,6} \\ a_{2,7} & a_{2,8} & a_{2,9} \end{bmatrix}$$

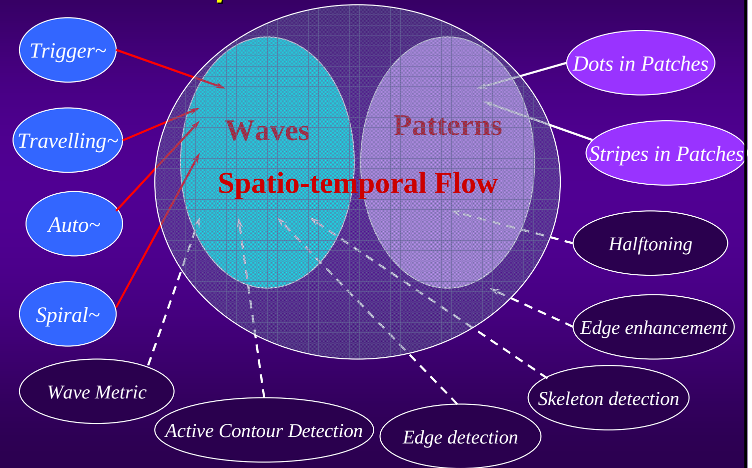


Tuned parameters

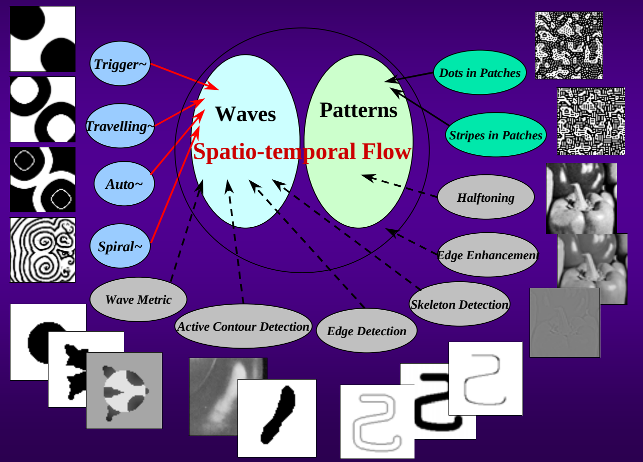
$$A_{12} = a_{12}; \qquad A_{21} = a_{21}; \qquad B_1 = b_0; \qquad z_1; \qquad z_2; \qquad \tau_1; \qquad \tau_2$$

Peter Pazmany Catholic University, Budapest

The Universe of Phenomena in a 2nd Order CNN



The Universe of Phenomena in a 2nd Order CNN



Wave Phenomena

/ combining trigger-waves travelling at different speed /

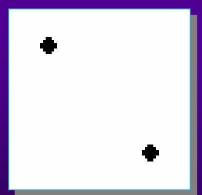
Examples:

- trigger-waves
- travelling-waves
- spiral-waves
- auto-waves

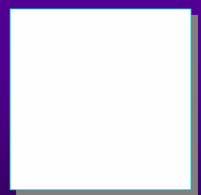
Trigger-waves

$$A_{11} = A_{22} = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3.00 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}; \qquad A_{12} = 1.0; \qquad A_{21} = 0;$$

$$B_1 = 0; \qquad z_1 = 3.75; \qquad z_2 = 3.75; \qquad \tau_1 / \tau_2 = 5$$



Layer 1

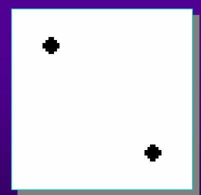


Layer 2

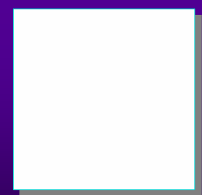
Travelling-waves

$$A_{11} = A_{22} = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3.00 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}; \qquad A_{12} = 3.0; \qquad A_{21} = -5;$$

$$B_1 = 0; \qquad z_1 = -1.25; \qquad z_2 = 2.25; \qquad \tau_1 / \tau_2 = 5$$



Layer 1

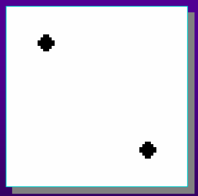


Layer 2

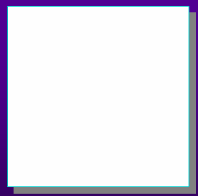
Spiral-waves

$$A_{11} = A_{22} = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3.00 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}; \qquad A_{12} = 5.5; \qquad A_{21} = -17.5;$$

$$B_1 = 0; \qquad z_1 = -1.25; \qquad z_2 = -1.25; \qquad \tau_1 / \tau_2 = 5$$



Layer 1



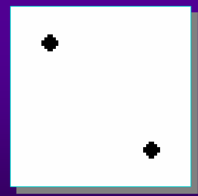
Layer 2

Peter Pazmany Catholic University, Budapest

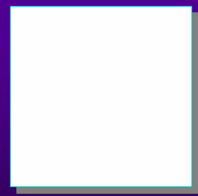
Auto-waves *(ChY-model)*

$$A_{11} = A_{22} = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3.00 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}; \qquad A_{12} = 5.5; \qquad A_{21} = -5;$$

$$B_1 = 0; \qquad z_1 = -1.25; \qquad z_2 = -1.25; \qquad \tau_1 / \tau_2 = 5$$



Layer 1



Layer 2

Peter Pazmany Catholic University, Budapest

Pattern Formation

/ combining low-pass, high-pass and band-pass type filters /

Examples:

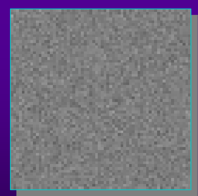
- patches and dots in patches
- patches and stripes in patches

Peter Pazmany Catholic University, Budapest

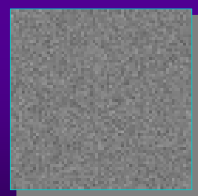
Patches and dots in patches

$$A_{11} = \begin{bmatrix} 0.5 & 1 & 0.5 \\ 1 & 2 & 1 \\ 0.5 & 1 & 0.5 \end{bmatrix}; \quad A_{22} = \begin{bmatrix} -0.5 & -1 & -0.5 \\ -1 & 2 & -1 \\ -0.5 & -1 & -0.5 \end{bmatrix}; \quad A_{12} = 4; \quad A_{21} = 0;$$

$$B_1 = 0; \qquad z_1 = 0; \qquad z_2 = 0; \qquad \tau_1 / \tau_2 = 0.1$$



Layer 1

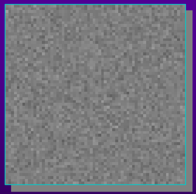


Layer 2

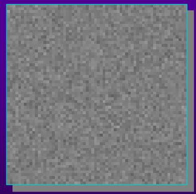
Patches and stripes in patches

$$A_{11}=\begin{bmatrix}0.5 & 1 & 0.5\\ 1 & 2 & 1\\ 0.5 & 1 & 0.5\end{bmatrix};\quad A_{22}=\begin{bmatrix}-0.5 & 0 & -0.5\\ 0 & 2 & 0\\ -0.5 & 0 & -0.5\end{bmatrix};\quad A_{12}=1;\quad A_{21}=0;$$

$$B_1=0;\quad z_1=0;\quad z_2=0;\quad \tau_1/\tau_2=0.1$$



Layer 1



Layer 2

Image Processing

/ combining diffusion, trigger-waves and various filtering effects /

Examples:

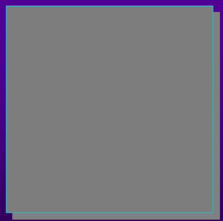
- edge enhancement
- halftoning
- active contour detection
- wave-metric
- edge detection
- skeletonization

Edge enhancement *(outer retina model)*

$$A_{11}=A_{22}=\begin{bmatrix}0.05 & 0.20 & 0.05\\ 0.20 & 0.00 & 0.20\\ 0.05 & 0.20 & 0.05\end{bmatrix};\quad A_{12}=0.1;\quad A_{21}=-5.0;$$

$$B_1=1.0;\quad z_1=0;\quad z_2=0;\quad \tau_1/\tau_2=10;$$

Input



Layer 1

Layer 2

Halftoning

$$A_{11}=\begin{bmatrix}-0.36 & -0.60 & -0.36\\ -0.60 & 1.05 & -0.60\\ -0.36 & -0.60 & -0.36\end{bmatrix};\quad A_{22}=\begin{bmatrix}0.05 & 0.20 & 0.05\\ 0.20 & 0.00 & 0.20\\ 0.05 & 0.20 & 0.05\end{bmatrix};\quad \tau_1/\tau_2=10;$$

$$A_{12}=1.0;\quad A_{21}=-1.0;\quad B_1=7.0;\quad z_1=0;\quad z_2=0;$$

Input

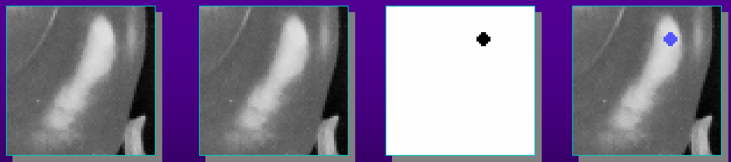
Layer 1

Layer 2

Active Contour Detection

$$A_{11} = 0.5\begin{bmatrix} 0.05 & 0.20 & 0.05 \\ 0.20 & 0.00 & 0.20 \\ 0.05 & 0.20 & 0.05 \end{bmatrix}; \quad A_{22} = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3.00 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}; \quad \tau_1 / \tau_2 = 1.0;$$

$$A_{12} = -2.5; \quad A_{21} = -0.1; \quad B_1 = 1.0; \quad z_1 = 0; \quad z_2 = 1.25$$



Input **Layer 1** **Layer 2** **Result & Inp**

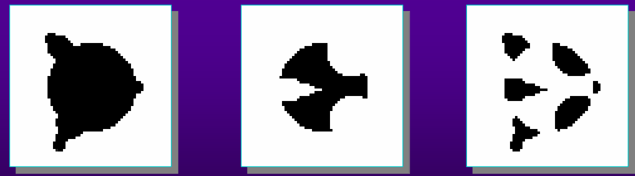
Peter Pazmany Catholic University, Budapest

Wave-metric

Obj1 **Obj2**

$$A_{11} = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3.00 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}; \quad A_{22} = 0.0; \quad \tau_1 / \tau_2 = 0.1;$$

$$A_{12} = -0.5; \quad A_{21} = 0.0; \quad B_1 = 2.0; \quad z_1 = 1.75; \quad z_2 = -1.0;$$



Input **Layer 1** **Layer 2**

Peter Pazmany Catholic University, Budapest

Edge detection

$$A_{11} = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3.00 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}; \quad A_{22} = \begin{bmatrix} -0.1 & -0.1 & -0.1 \\ -0.1 & 2.3 & -0.1 \\ -0.1 & -0.1 & -0.1 \end{bmatrix}; \quad \tau_1 / \tau_2 = 1.0;$$

$$A_{12} = -1.0; \quad A_{21} = 0.0; \quad B_1 = 0.0; \quad z_1 = 3.75; \quad z_2 = 0.0;$$

Input **Layer 1** **Layer 2**

Peter Pazmany Catholic University, Budapest

Skeletonization

$$A_{11} = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3.00 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}; \quad A_{22} = \begin{bmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & 5.4 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{bmatrix}; \quad \tau_1 / \tau_2 = 0.5;$$

$$A_{12} = -2.5; \quad A_{21} = 0.0; \quad B_1 = 0.0; \quad z_1 = 3.75; \quad z_2 = 0.0;$$

Input **Layer 1** **Layer 2**

Summary

A ***double time-scale, single-input 2nd order/3-layer CNN model (with two mutually coupled 1st order RC cells and max. 21 synapses)*** is capable to reproduce a number of complex wave and pattern formation phenomena, and can be used as the core architecture in various meaningful image processing operations.

Higher complexity cellular chips could also be built and used in PDE approximation, modeling and engineering applications.