Efficient Hardware Architectures and Algorithms for Embedded Vision Systems

Eva Dokladalova [1,2]

[1] ESIEE Paris, Computer Science Department, France

[2] Paris-Est University, LIGM (Gaspard Monge Computer Science Laboratory), A3SI team, CNRS UMR 8049, France

October 27, 2015

Who I am?

FACULTY OF ELECTRICAL ENGINEERING UNIVERSITY OF WEST BOHEMIA

Ing.

INP Grenoble

DEA

ESIEE PARIS

Head of CS spec.

MINES ParisTech

PhD

DE LA RECHERCHE À L'INDUSTRIE cea

PostDoc/Perm.

Gaspard - Monge Laboratoire d'informatique

Research

Efficient Hardware Architectures and Algorithms for Embedded Vision Systems

Where and why?

Applicative context

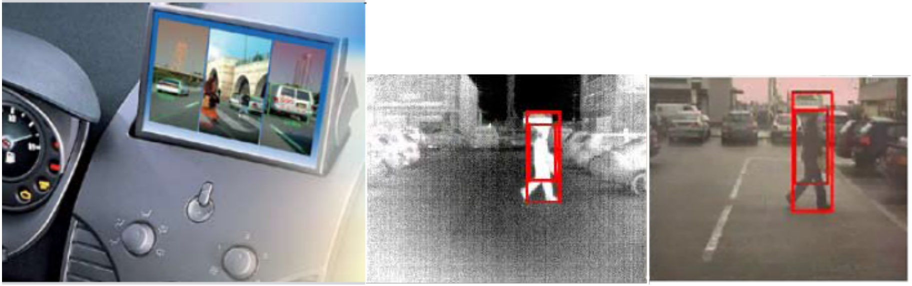
1. Very specific sensing systems
2. Multiple technologically different vision sensors
3. High perfomance computing ability
4. Low processing latency requirements
5. Low energy consumption constraints

Where and why?


a

[a] credits: Sagem Defense

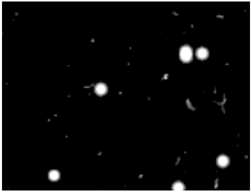
Where and why?



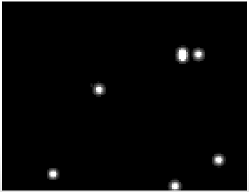
[a]

[a]credits: CEA LIST, MEDEA+ CarVision project

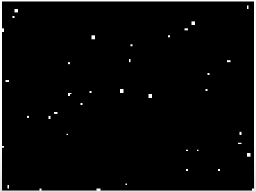
Where and why?



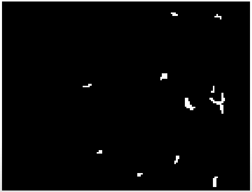
Alpha + gamma + electron



alpha



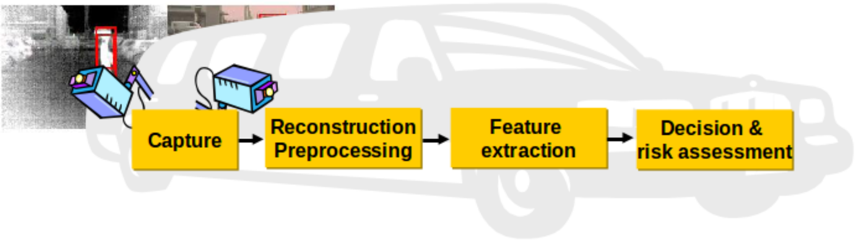
gamma



electron [a]

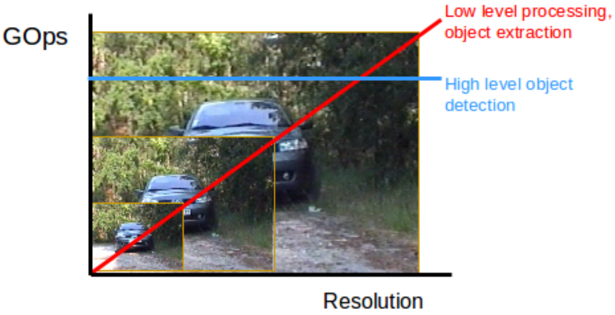
[a] credits: UTEF Praha, ZCU Plzen

Efficiency bottlenecks



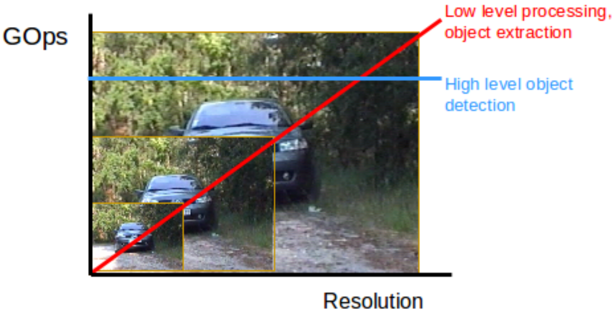
- Intensive memory accesses

Efficiency bottlenecks



- Intensive memory accesses
- High performance computing

Efficiency bottlenecks



- Intensive memory accesses
- High performance computing
- ! Low working frequency

Propositions

Axioms on algorithm limitations

- Respect sequential data reading $\rightarrow$ sensor pixel stream
- Enable on-the-fly processing $\rightarrow$ eliminate intermediate storage
- Reduce algorithm complexity $\rightarrow$ $O(1)$ per pixel
- Consider low and object extraction

Remark: properties interesting for all type of computing platforms

Image processing approaches

	Geometric space	Abstract space
Linear	**Linear** • Convolution • Fourrier • Wavelets	**Statistical** • Multivariate analysis • Neural networks
Nonlinear	**Morphologic** • Filtering • Measures • Segmentation	**Syntactic** • Grammars • Indexation • Structural pattern

Contents

Mathematical morphology

- Born in 1964 at Ecole des Mines de Paris, France.

- Mathematical theory studing interactions between image and set called structuring element (SE)
- Various image processing techniques implemented by combining only a few simple operations: erosion/dilation, closing/opening

Binary dilation and erosion

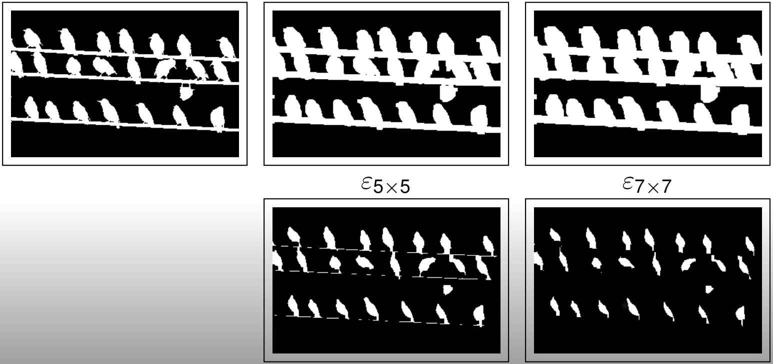
Definition

Let F be a binary image and B be a set called structuring element (SE).

$$\delta_B(F) = \{z : B(z) \cap F\} \qquad \varepsilon_B(F) = \{z : B(z) \subseteq F\}$$

Input image F $\qquad$ $\delta_{5\times5}$ $\qquad$ $\delta_{7\times7}$

$\varepsilon_{5\times5}$ $\qquad$ $\varepsilon_{7\times7}$

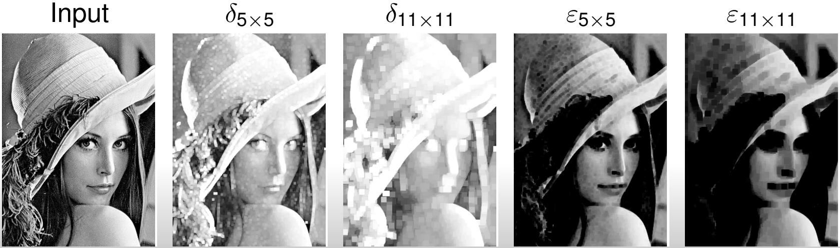


Grayscale dilation and erosion

Definition

Let B be a flat structuring element (SE)

$$[\delta_B(f)](x) = \max_{b \in B} [f(x+b)] \qquad [\varepsilon_B(f)](x) = \min_{b \in B} [f(x-b)]$$

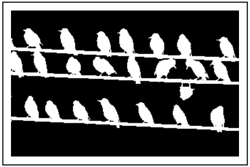


Compound operators

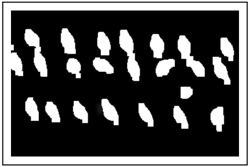
- Opening; Closing:

$$\gamma_B(f) = \delta_B[\varepsilon_B(f)] \; ; \qquad \varphi_B(f) = \varepsilon_B[\delta_B(f)]$$

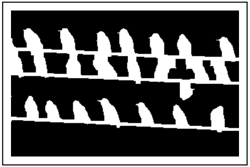
Original



Opening



Closing



- Gradient:

$$g(f) = \delta_B(f) - \varepsilon_B(f)$$

- Top hat:

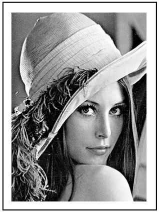
$$th^{\gamma}(f) = f - \gamma_B(f) \; ; \qquad th^{\varphi}(f) = \varphi_B(f) - f$$

Compound operators

- Opening; Closing:

$$\gamma_B(f) = \delta_B[\varepsilon_B(f)] \; ; \qquad \varphi_B(f) = \varepsilon_B[\delta_B(f)]$$

Original



Opening

Closing

- Gradient:

$$g(f) = \delta_B(f) - \varepsilon_B(f)$$

- Top hat:

$$th^{\gamma}(f) = f - \gamma_B(f) \; ; \qquad th^{\varphi}(f) = \varphi_B(f) - f$$

Practical use cases

1. Image enhancing and nonlinear filtering

$$\mathrm{ASF}^{\lambda} = \varphi^{\lambda}\gamma^{\lambda}\varphi^{\lambda-1}\gamma^{\lambda-1}\dots\varphi^{1}\gamma^{1}$$

Figure: Alternate sequential filters

Practical use cases

1. Image enhancing and nonlinear filtering

$$\mathrm{ASF}^{\lambda} = \varphi^{\lambda}\gamma^{\lambda}\varphi^{\lambda-1}\gamma^{\lambda-1}\ldots\varphi^{1}\gamma^{1}$$

2. Directional size distributions (texture analysis)

$$PS_{\lambda_j B}(f) = \sum_{D}\left(\gamma_{\lambda_i B} f - \gamma_{\lambda_j B} f\right)$$

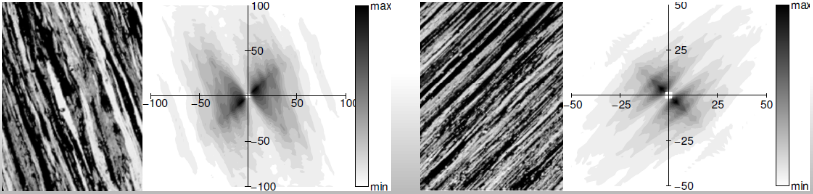
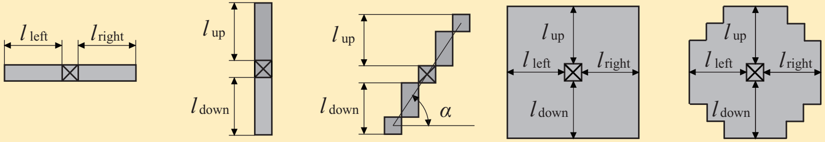


Figure: Texture analysis

Application Challenges

Synthesis of application requirements

- Variety of sizes, shapes, orientation of Structuring Elements



Horizontal Vertical Inclined Rectangle Octagon

Computational complexity

- Naively: $n \times n$ square SE has complexity $\mathcal{O}(n^2)$
- E.g., $B = 11 \times 11$ needs 120 max() per pixel

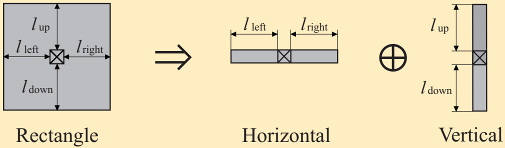
Optimizing Computation of Dilation

Structuring element decomposition

- Decomposition by means of the Minkowski set addition

$$\delta_{B_1 \oplus B_2}(f) = \delta_{B_1}\delta_{B_2}(f)$$

- Reduce $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$
- Dilation by square SE is equal to horizontal dilation followed by vertical dilation



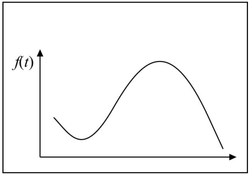
1-D Efficient algorithms with $\mathcal{O}(1)$

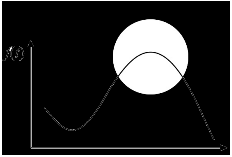
- Reduce $\mathcal{O}(n)$ to $\mathcal{O}(1)$

Morphological operators in constant time

Principle

- Data read from left to right
- Limited field of view
- Results obtained only from visible data





1-D Dilation in constant time

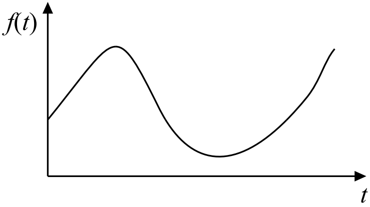


Figure: 1-D stream dilation

Principle

- Field of view limited on SE
- Find maximum within scope of the SE
- Erase the most recent pixel within SE if its value is smaller than the current pixel

1-D Dilation in constant time

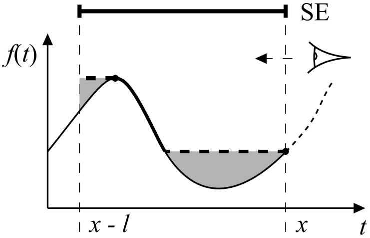


Figure: 1-D stream dilation

Principle

- Field of view limited on SE
- Find maximum within scope of the SE
- Erase the most recent pixel within SE if its value is smaller than the current pixel

1-D Dilation in constant time

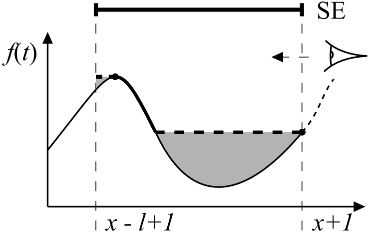


Figure: 1-D stream dilation

Principle

- Field of view limited on SE
- Find maximum within scope of the SE
- Erase the most recent pixel within SE if its value is smaller than the current pixel

1-D Dilation in constant time

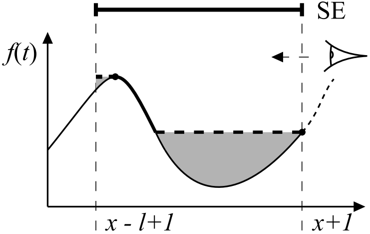


Figure: 1-D stream dilation

Principle

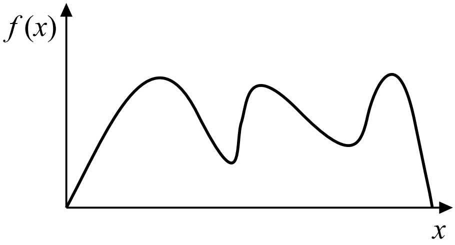
- Field of view limited on SE
- Find maximum within scope of the SE
- Erase the most recent pixel within SE if its value is smaller than the current pixel

Interesting feature

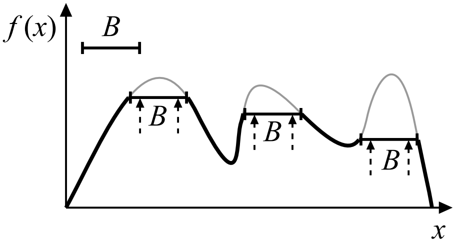
- Size of SE can be changed on the fly !
- SE implemented as a simple queue (FIFO)

1-D Opening in general

- Let B be a flat SE of length l
- Gray-scale opening is defined as the union of all SEs that fit under the graph of a function f



(a) Input signal $f(x)$



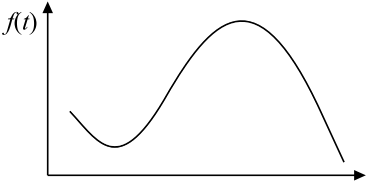
(b) Output signal $\gamma_B f(x)$

Intuition

- Erase all signal peaks narrower than l pixels.

1-D opening in constant time (J. Bartovsky)

Peak elimination principle

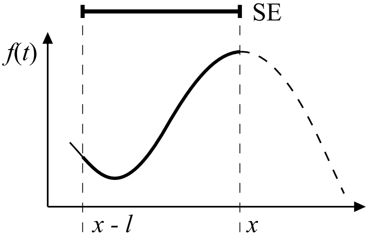


Principle

- Input signal $f(t)$
- SE position
- All peaks located under SE are erased
- Erase the most recent pixel within SE if it is a peak

1-D opening in constant time (J. Bartovsky)

Peak elimination principle

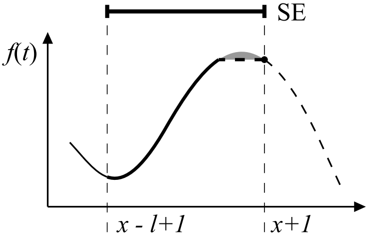


Principle

- Input signal $f(t)$
- SE position
- All peaks located under SE are erased
- Erase the most recent pixel within SE if it is a peak

1-D opening in constant time (J. Bartovsky)

Peak elimination principle

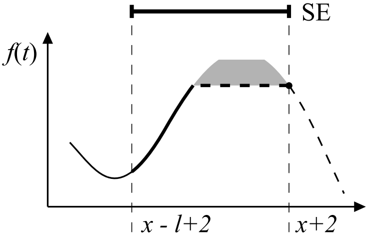


Principle

- Input signal $f(t)$
- SE position
- All peaks located under SE are erased
- Erase the most recent pixel within SE if it is a peak

1-D opening in constant time (J. Bartovsky)

Peak elimination principle

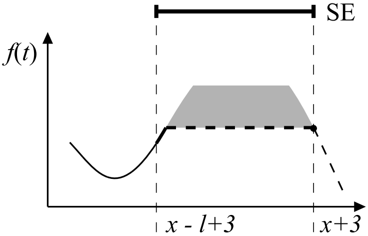


Principle

- Input signal $f(t)$
- SE position
- All peaks located under SE are erased
- Erase the most recent pixel within SE if it is a peak

1-D opening in constant time (J. Bartovsky)

Peak elimination principle



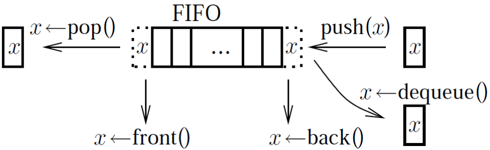
Principle

- Input signal $f(t)$
- SE position
- All peaks located under SE are erased
- Erase the most recent pixel within SE if it is a peak

Interesting features

- Pattern spectrum obtained directly !

Algorithm formulation



1-D Dilation

while Increasing slope **do**
- Erase invisible pixel

Push current pixel
Pop outdated pixel (given by *l*)
Output result pixel

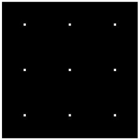
1-D Opening

while Peak detected **do**
- Erase the most recent pixel

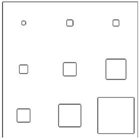
Push current pixel
Pop outdated pixel
Output result pixel

Extension to spatially variant morphology

- SE variation has to be continous, fast approximation



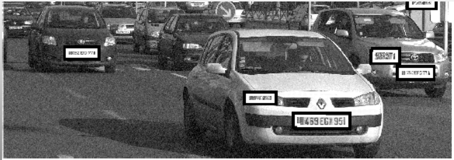
Input



Variant SE

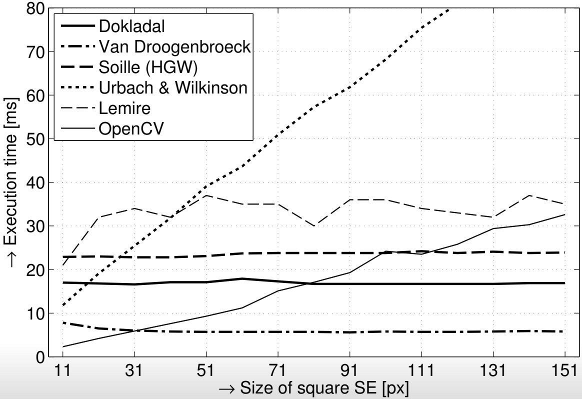
1-D space variant dilation

while Increasing slope **do**
    Erase invisible pixel
Push current pixel
Pop outdated pixel (given by *l*)
Adjust next reading position
Adjust writing position
Output result pixel



Comparison of Dilation Algorithms

- Dilation by square SE of real-world photo
- General-purpose 64-bit Intel processor, linux, gcc



M. Van Droogenbroeck, and M. J. Buckley. Morphological erosions and openings: Fast algorithms based on anchors. J. Math. Imaging Vis., vol. 22, no. 2-3, pages 121–142, 2005.

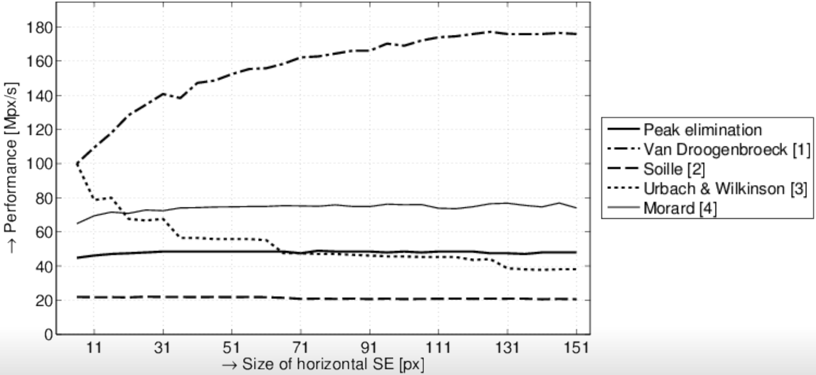
P. Soille, E. J. Breen, and R. Jones. Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. IEEE Trans. Pattern Anal. Mach. Intell., vol. 18, no. 5, pages 562–567, 1996.

E. R. Urbach, and M. H. F. Wilkinson. Efficient 2-D grayscale morphological transformations with arbitrary flat structuring elements. IEEE Trans. Image Processing, vol. 17, no. 1, pages 1 –8, jan. 2008.

D. Lemire. Streaming maximum-minimum filter using no more than three comparisons per element. CoRR, 2006.

Comparaison of Opening algorithms

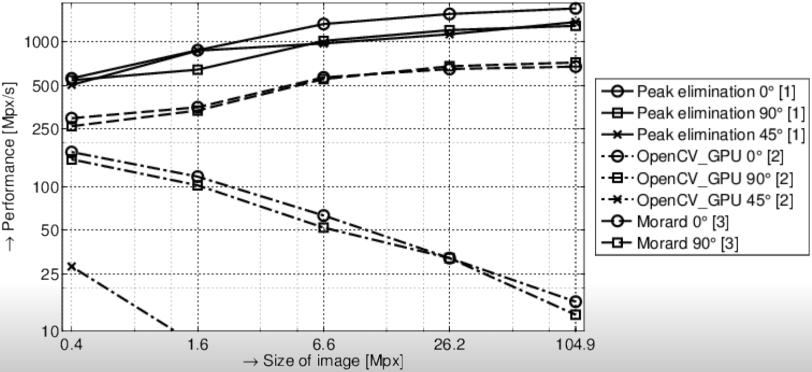
- Opening and size distribution (natural image)
- General-purpose 64-bit Intel Xeon processor, linux, gcc



[1] M. Van Droogenbroeck, and M. J. Buckley. Morphological erosions and openings: Fast algorithms based on anchors. J. Math. Imaging Vis., vol. 22, no. 2-3, pages 121-142, 2005.
[2] P. Soille, E. J. Breen, and R. Jones. Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. IEEE Trans. Pattern Anal. Mach. Intell., vol. 18, no. 5, pages 562-567, 1996.
[3] E. R. Urbach, and M. H. F. Wilkinson. Efficient 2-D grayscale morphological transformations with arbitrary flat structuring elements. IEEE Trans. Image Processing, vol. 17, no. 1, pages 1-8, jan. 2008.
[4] V. Morard, P. Dokladal, and E. Decenciere. Linear openings in arbitrary orientation in O(1) per pixel. In Acoustics, Speech and Signal Processing (ICASSP), pages 1457-1460, may 2011.

Comparaison of 1-D Opening algorithms (P. Karas, J. Bartovsky)

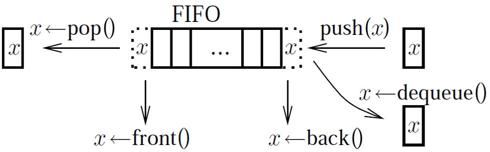
- Opening par SE with arbitrary angle (texture)
- Nvidia Tesla C2050 GPU, CUDA 3.1



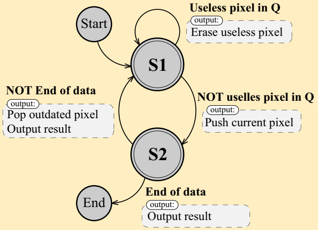
[1] P. Karas, V. Morard, J. Bartovsky, T. Grandpierre, E. Dokladalova, P. Matula, and P. Dokladal. GPU implementation of linear morphological openings with arbitrary angle. Journal of Real-Time Image Processing, 2012.
[2] OpenCV 2.0. http://opencv.willowgarage.com, 2012.
[3] V. Morard, P. Dokladal, and E. Decenciere. Linear openings in arbitrary orientation in O(1) per pixel. In Acoustics, Speech and Signal Processing (ICASSP), pages 1457-1460, may 2011.

Contents

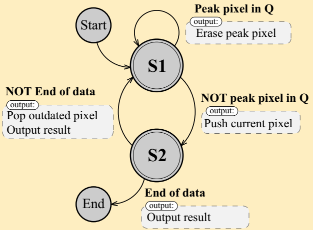
Hardware implementation



1-D Dilation

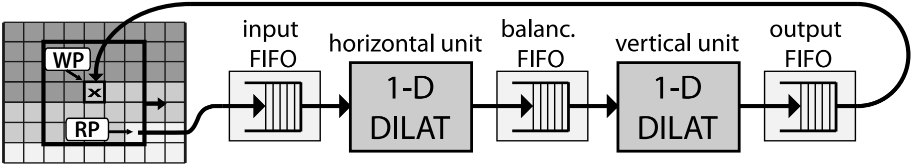


1-D Opening



2-D Rectangular Dilation Architecture

- Consists of one horizontal and one vertical 1-D unit
- Sequential access to data allows for concatenation of both units
- Units are coupled by FIFOs in order to optimize performance



Inter-operator parallelism

- Two operators run at the same time on time-delayed data
- Enable operator chaining without memorization

Example : concatanted operators - ASF

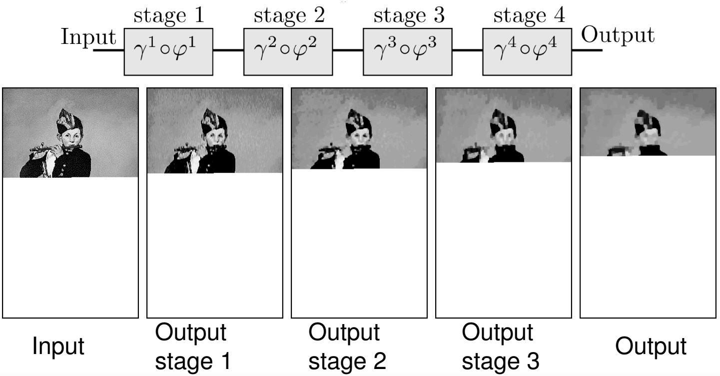


Figure: Computation of ASF^4.

Output available with only few lines latency !

High performance computing
Morphological Co-Processing Unit (J. Bartovsky)

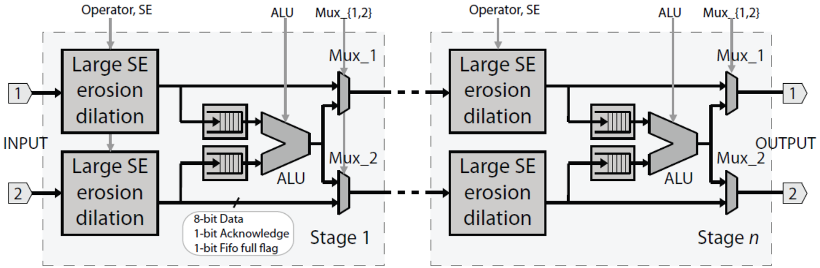
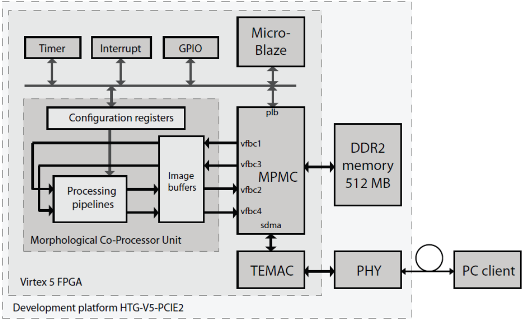


Figure: Large SE programmable pipeline

Important features

- *Programability* - shape, size and angle of SE
- *Scalability* - constant performances
- *Performances* - 200 Mpix/s Full HD TV 1080p 100Hz

High performance computing MCPU (J. Bartovsky)



System characteristics

- Tri-speed Ethernet interface using either TCP/IP or UDP/IP
- MicroBlaze processor using Peripheral Local Bus (PLB)
- Software interface (integrated in MorphM), C/C++ and Python

High performance computing
Performance evaluation

Table: Comparison of several FPGA and ASIC architectures concerning morphological dilation and erosion. N, M stand for the image width and height of respective architectures.

	Processing unit				Hardware System		Application Example ASF[6]		
	Technology	Supported SE	Throughput [Mpx/s]	f_{max} [MHz]	Number of units	Supported image	Image scans	FPS [frame/s]	Latency [px]
Clienti [1]	FPGA	arbitrary 3×3	403	100	16	1024×1024	6	66.7	$5NM + 84N$
Chien [2]	ASIC	disc 5×5	190	200	1	720×480	45	12.2	$44NM + 84N$
Déforges (a) [3]	FPGA	arbitrary 8-convex	50	50	1	512×512	13	14.7	$12NM + 84N$
Déforges (b) [3]	FPGA	arbitrary 8-convex	50	50	13	512×512	1	50	$84N$
our MCPU	FPGA	regular polygon	195	100	13	1024×1024	1	185	$84N$

[1]Ch. Clienti, M. Bilodeau, and S. Beucher. An efficient hardware architecture without line memories for morphological image processing. In ACIVS '08, pages 147–156, Berlin, Heidelberg, 2008. Springer-Verlag
[2] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen. Partial-result-reuse architecture and its design technique for morphological operations with flat structuring elements. Circuits and Systems for Video Technology, IEEE Transactions on, 15(9):1156 – 1169, sept. 2005.
[3] O. Deforges, N. Normand, and M. Babel. Fast recursive grayscale morphology operators: from the algorithm to the pipeline architecture. Journal of Real-Time Image Processing, pages 1–10, 2010.

Contents

Self Aware Vision System

Self Aware Vision System

Proposition

Morphological coprocessor unit for scene understanding (very low latency, low consumption, polyvalent).

Observe

Decide

Act

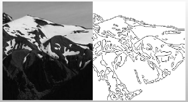
Feedback control loop

Morphological scene understanding

- Global approach
- Morphological descriptors: openings, closing, etc...
- SVM classification
- SceneClass13 data set from Stanford Vision Lab (3800 images)

Classification hierarchy

1. Indoor and urban scenes *x* Countryside
2. Indoor scenes *x* Urban scenes



Experimental results

Indoor and urban scenes *x* Countryside scenes

Opening by rectangular SE $>$ 90 %

Indoor scenes *x* Urban scenes

Opening by rectangular SE $>$ 86 %

Urban scenes *x* Countryside scenes

Opening by linear SE $>$ 94 %

Mountain scenes *x* Other countryside scenes

Closing by linear SE $>$ 80 %

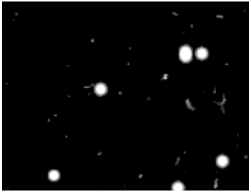
Implementation evaluation

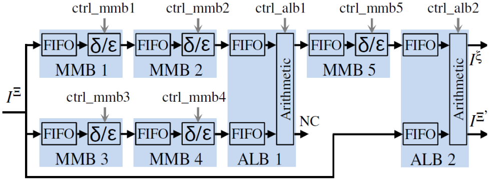
Table: Performance results of selected operators. Image is natural photo 1000×1000 px, time results are in milliseconds (unless seconds are specified).

Operator	Shape of SE	Size of SE or λ	MCPU	OpenCV at Sabre	Smil at Sabre	OpenCV at Xeon
Dilation	Rectangle	3×3	21.9	32.7	8.4	0.58
Opening	Rectangle	151×151	24.3	2450	1083	38.6
Opening	Octagon	151×151	41.9	246 s	2453	2301
Opening by recon.	Rectangle	151×151	544	47.6 s	22.1 s / 2110*	1940
Opening by recon.	Octagon	151×151	512	289 s	21.1 s	4356
ASF	Rectangle	$\lambda = 11$	64.2	4530	1987	57.1
ASF	Octagon	$\lambda = 11$	83.3	77 s	3872	814
Pattern spectrum	Rectangle	$\lambda = 11$	62.3	2570	1098	53.8
Pattern spectrum	Octagon	$\lambda = 11$	62.7	21.2 s	1782	249
by recon.	Rectangle	$\lambda = 11$	2530	190 s	85.3 s / 18.2 s*	8920
by recon.	Octagon	$\lambda = 11$	2410	201 s	81.5 s	8751

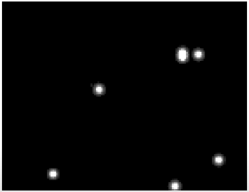
note *: The second result is obtained by an algorithm based on hierarchy queues.
λ is the maximal SE in the chain.

Classification of particles

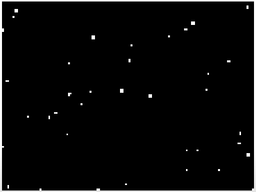




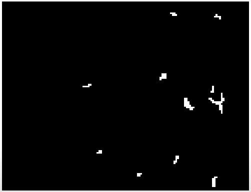
Alpha + gamma + electron



alpha



gamma



electron

⇒ up to 750 fps !

Contents

Conclusions

Algorithms

- New algorithm familly with $O(1)$ complexity for atomic morphological operations
- Very significant enhancement of implementation performances (GPU, GPP and FPGA)

Architecture

- Efficient hardware implementation
- Programmable co-processing unit

Extensions

- Spatially variant morphology
- Scene understanding preprocessing