Relations as Images

Mathieu Alain

Jules Desharnais

Université Laval, Québec, Canada

Plan

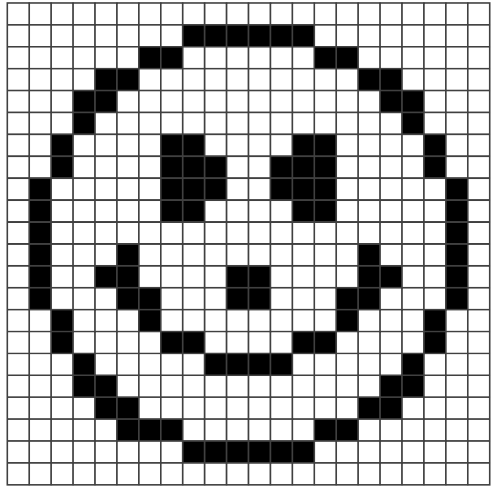
1. Introduction : relations, black and white images, mathematical morphology
2. Notation
3. Representing an image in the plane by a relation
4. Dilation and erosion
5. Graph morphology
6. Conclusion

1 Introduction : relations, black and white images, mathematical morphology

$$
\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&0&1&1&0&0&0&0&1&1&0&0&0&0&1&0&0\\
0&0&1&0&0&0&0&1&1&1&0&0&1&1&1&0&0&0&0&1&0&0\\
0&1&0&0&0&0&0&1&1&1&0&0&1&1&1&0&0&0&0&0&1&0\\
0&1&0&0&0&0&0&1&1&0&0&0&0&1&1&0&0&0&0&0&1&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0\\
0&1&0&0&1&1&0&0&0&0&1&1&0&0&0&0&1&1&0&0&1&0\\
0&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&1&0&0&0&0&1&1&0&0&0&0&1&1&0&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&1&1&1&1&0&0&0&0&0&1&0&0&0\\
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}
$$

Relation as a Boolean matrix

$$
\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&0&1&1&0&0&0&0&1&1&0&0&0&0&1&0&0\\
0&0&1&0&0&0&0&1&1&1&0&0&1&1&1&0&0&0&0&1&0&0\\
0&1&0&0&0&0&0&1&1&1&0&0&1&1&1&0&0&0&0&0&1&0\\
0&1&0&0&0&0&0&1&1&0&0&0&0&1&1&0&0&0&0&0&1&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0\\
0&1&0&0&1&1&0&0&0&0&1&1&0&0&0&0&1&1&0&0&1&0\\
0&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&1&0&0&0&0&1&1&0&0&0&0&1&1&0&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&1&1&1&1&0&0&0&0&0&1&0&0&0\\
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}
$$



Relation as a Boolean matrix

RelView display of the same relation

- **RelView convention (the one we use) :** $\begin{cases} 0\text{: white} \\ 1\text{: black} \end{cases}$

Mathematical morphology

- 1960s, 1970s: mostly developped for the analysis and transformation of binary (i.e., black and white) digital images (Matheron, Serra, . . .).
- Later extended in various directions : complete lattices, grey-level images, graphs, hypergraphs (Heijmans, Ronse, Nacken, Toet, Vincent, Stell, . . .).
- Basic operations: dilation, erosion, opening, closing.
- In this talk:
  - Implementation of dilation and erosion under RelView for binary images, and performance comparison with Mathematica and Matlab, where these operations are primitive.
  - Implementation of dilation and erosion under RelView for graphs.

2 Notation

I	identity relation
O	empty relation
L	universal relation
$\cup$	union
$\cap$	intersection
$;$	composition
$^{\mathsf{T}}$	transposition/conversion
$\overline{}$	complementation
*	reflexive transitive closure
$/$	left residuation
$\backslash$	right residuation

1. For $R_1 : T \leftrightarrow T_1$ and $R_2 : T \leftrightarrow T_2$

   Tupling $\langle R_1, R_2] : T \leftrightarrow T_1 \times T_2$

2. For $R_1 : T_1 \leftrightarrow T$ and $R_2 : T_2 \leftrightarrow T$

   Cotupling $[R_1, R_2\rangle = T_1 \times T_2 \leftrightarrow T$

3. For $R_1 : S_1 \leftrightarrow T_1$ and $R_2 : S_2 \leftrightarrow T_2$

   Parallel product $[R_1, R_2] : S_1 \times S_2 \leftrightarrow T_1 \times T_2$

3 Representing an image in the plane by a relation

Convention We use the standard indexing orientation for matrices rather than the standard Cartesian coordinates orientation.

$$R \;:\; \begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Associated relations

$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
S_{d}	S_{c}	o_{d}	o_{c}	O
successor	successor	origin	origin	origin
domain side	codomain side	domain side	codomain side	
y axis	x axis	y axis	x axis	
`succ(R)`	`succ(R^)`	`init(On1(R))`	`init(On1(R^))`	

Symbolic expression for a concrete relation

$$R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} = \begin{array}{rrrr} & & & O \,;\, S_{\mathsf{c}}^{2} \\ \cup & S_{\mathsf{d}}^{\mathsf{T}} \,;\, O \,;\, S_{\mathsf{c}} & \cup & S_{\mathsf{d}}^{\mathsf{T}} \,;\, O \,;\, S_{\mathsf{c}}^{2} \\ \cup & S_{\mathsf{d}}^{2\mathsf{T}} \,;\, O \,;\, S_{\mathsf{c}} & \cup & S_{\mathsf{d}}^{2\mathsf{T}} \,;\, O \,;\, S_{\mathsf{c}}^{2} \end{array}$$

General form:

$$R = (\bigcup i, j \mid iRj : S_{\mathsf{d}}^{i\mathsf{T}} \,;\, O \,;\, S_{\mathsf{c}}^{j})$$

(Partial) addition relation A

Let

$$\begin{array}{rcll} T_1 & = & \{0,1,2,3\} & \\ T_2 & = & \{0,1,2\} & \\ o_2 & = & \{0\} & \text{origin in } T_2 \\ S_1 & = & \{(0,1),(1,2),(2,3)\} & \text{successor on } T_1 \\ S_2 & = & \{(0,1),(1,2)\} & \text{successor on } T_2 \end{array}$$

Define $A\colon T_1 \times T_2 \leftrightarrow T_1$

$$A = \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 2 & 3 \\ 2 & 2 & 3 & \\ 3 & 3 & & \end{array} = \begin{array}{l} \{((0,0),0),((0,1),1),((0,2),2), \\ ((1,0),1),((1,1),2),((1,2),3), \\ ((2,0),2),((2,1),3), \\ ((3,0),3)\} \end{array} = [S_1, S_2^{\mathsf{T}}]^* \mathbin{;} [\mathsf{I}_1, o_2 \mathbin{;} \mathsf{L}\rangle$$

Going through $[S_1, S_2^{\mathsf{T}}]^* \mathbin{;} [\mathsf{I}_1, o_2 \mathbin{;} \mathsf{L}\rangle$

$$(1,2) \;\to\; [S_1, S_2^{\mathsf{T}}] \;\to\; (2,1) \;\to\; [S_1, S_2^{\mathsf{T}}] \;\to\; (3,0) \;\to\; [\mathsf{I}_1, o_2 \mathbin{;} \mathsf{L}\rangle \;\to\; 3$$

Size of $[S_1, S_2^{\mathsf{T}}]^*$**:** $|T_1| \times |T_2| \times |T_1| \times |T_2| = |T_1|^2 \times |T_2|^2$

Size of A**:** $|T_1| \times |T_2| \times |T_1| = |T_1|^2 \times |T_2|$

4 Dilation and erosion

$$\text{Let}\quad \begin{array}{ll} R : \mathbb{Z} \leftrightarrow \mathbb{Z} & \text{(the image)}, \\ P : \mathbb{Z} \leftrightarrow \mathbb{Z} & \text{(the pattern)}, \\ A : \mathbb{Z} \times \mathbb{Z} \leftrightarrow \mathbb{Z} & \text{(addition)}. \end{array}$$

The dilation $R \oplus P$ of R by P is the pointwise addition (also called the Minkowski addition) of R and P:

$$R \oplus P = \{(x_R + x_P, y_R + y_P) \mid (x_R, y_R) \in R \wedge (x_P, y_P) \in P\}.$$

Using the addition function A, this can be expressed as

$$R \oplus P = A^{\mathsf{T}} ; [R, P] ; A.$$

Dilation in the integer grid: example

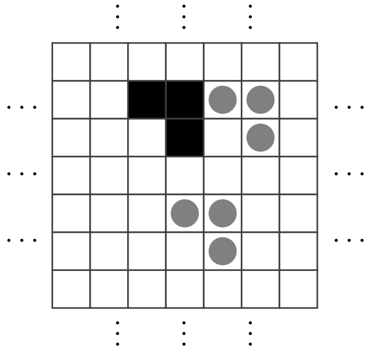
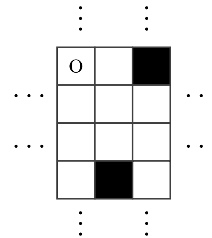


Image R (black) and dilation $R \oplus P$ (grey)



Structuring element P
(Pattern)

In the integer grid,

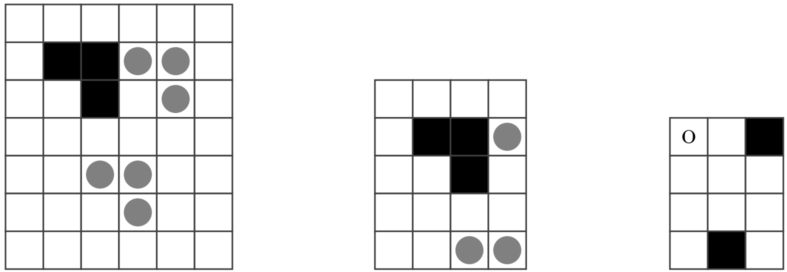
$$R \oplus P = P \oplus R,$$

hence the symmetric symbol for dilation. However, this is not the case for finite grids:

- The size of the image grid is normally much larger than that of the structuring element and the size of the result is that of the image;
- There are border effects.

This is why we write the dilation of R by P as $R \triangleright P$.

Dilation in finite grids: border effects



P

The partial addition gives the correct result in both cases

$$R \triangleright P \;=\; A_{\mathsf{d}}^{\mathsf{T}} \,;\, [R, P] \,;\, A_{\mathsf{c}}$$

Implementation under RelView

- Straigthforward: for fixed size grids, compute A_{d} and A_{c} once, and apply to various R and P.
- But: For large R and P, relations A_{d}, A_{c} and especially $[R, P]$ are huge.

Size (type) of $R \triangleright P$ = size of R

Computing dilation using less space

$$
\begin{array}{ll}
& R_1 \triangleright R_2 \\
= & \qquad \langle \text{ Definition of } \triangleright \text{ } \rangle \\
& A_{\mathsf{d1}}^{\mathsf{T}} \,;\, [R_1, R_2] \,;\, A_{\mathsf{c1}} \\
= & \qquad \langle \text{ Definition of } A_{\mathsf{d1}} \text{ and } A_{\mathsf{c1}} \text{ } \rangle \\
& \langle \mathsf{I}_{\mathsf{d1}}, \mathsf{L} \,;\, o_{\mathsf{d2}}^{\mathsf{T}}] \,;\, [S_{\mathsf{d1}}^{\mathsf{T}}, S_{\mathsf{d2}}]^* \,;\, [R_1, R_2] \,;\, [S_{\mathsf{c1}}, S_{\mathsf{c2}}^{\mathsf{T}}]^* \,;\, [\mathsf{I}_{\mathsf{c1}}, o_{\mathsf{c2}} \,;\, \mathsf{L}\rangle \\
= & \qquad \langle \text{ Proof in the paper } \rangle \\
& (\bigcup i, j : \mathbb{N} \mid i R_2 j : S_{\mathsf{d1}}^{i\mathsf{T}} \,;\, R_1 \,;\, S_{\mathsf{c1}}^{j}) \qquad (*) \\
= & \qquad \langle \text{ Proof in the paper } \rangle \\
& (\bigcup i, j : \mathbb{N} \mid o_{\mathsf{d2}}^{\mathsf{T}} \,;\, S_{\mathsf{d2}}^{i} \,;\, R_2 \,;\, S_{\mathsf{c2}}^{j\mathsf{T}} \,;\, o_{\mathsf{c2}} \neq \mathsf{O} : S_{\mathsf{d1}}^{i\mathsf{T}} \,;\, R_1 \,;\, S_{\mathsf{c1}}^{j}) \qquad (**)
\end{array}
$$

- (**) is readily implemented under RelView. The expression $o_{\mathsf{d2}}^{\mathsf{T}};S_{\mathsf{d2}}^{i};R_2;S_{\mathsf{c2}}^{j\mathsf{T}};o_{\mathsf{c2}}$ is a 1×1 matrix and can be evaluated efficiently. No parallel product or tupling is involved.
- (*) is compact and easily interpreted.

RelView program for dilation

$$R_1 \triangleright R_2 \;=\; (\bigcup i,j : \mathbb{N} \mid o_{\mathsf{d2}}^{\mathsf{T}} ; S_{\mathsf{d2}}^{i} ; R_2 ; S_{\mathsf{c2}}^{j\mathsf{T}} ; o_{\mathsf{c2}} \neq \mathsf{O} : S_{\mathsf{d1}}^{i\mathsf{T}} ; R_1 ; S_{\mathsf{c1}}^{j})$$

```
Dilation(R1, R2)
DECL Sd1, Sc1, Sd2, Sc2, od2, oc2,
     i1, j1, i2, j2, res, cond

BEG
Sd1 = succ(R1);
Sc1 = succ(R1^);
Sd2 = succ(R2);
Sc2 = succ(R2^);
od2 = init(On1(R2));
oc2 = init(On1(R2^));
i1 = I(Sd1);
j1 = I(Sc1);
i2 = I(Sd2);
j2 = I(Sc2);
res = O(R1);
WHILE -empty(i2) DO
  WHILE -empty(j2) DO
    cond = od2^*i2*R2*j2^*oc2
    IF -empty(cond) THEN
      res = res|i1^*R1*j1 FI
    j1 = j1*Sc1;
    j2 = j2*Sc2 OD
  j1 = I(Sc1);
  j2 = I(Sc2);
  i1 = i1*Sd1;
  i2 = i2*Sd2 OD
RETURN res
END.
```

Left erosion $\not\triangleright$ (the standard erosion of mathematical morphology)

Left erosion is related to dilation by a Galois connection

$$R_1 \triangleright R_2 \subseteq R_3 \;\Leftrightarrow\; R_1 \subseteq R_3 \not\triangleright R_2$$

Shown in the paper

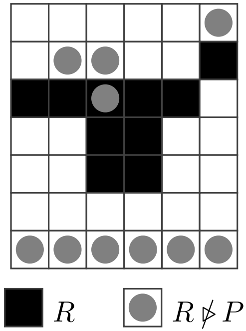
$$\begin{array}{rcl} R_1 \not\triangleright R_2 & = & (\bigcap i, j : \mathbb{N} \mid i R_2 j : S_{\mathsf{d1}}^{i\mathsf{T}} \backslash R_1 / S_{\mathsf{c1}}^{j}) \\ & = & (\bigcap i, j : \mathbb{N} \mid o_{\mathsf{d2}}^{\mathsf{T}} \,;\, S_{\mathsf{d2}}^{i} \,;\, R_2 \,;\, S_{\mathsf{c2}}^{j\mathsf{T}} \,;\, o_{\mathsf{c2}} \neq \mathsf{O} : S_{\mathsf{d1}}^{i\mathsf{T}} \backslash R_1 / S_{\mathsf{c1}}^{j}) \end{array}$$

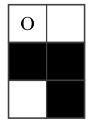
Compare with dilation

$$\begin{array}{rcl} R_1 \triangleright R_2 & = & (\bigcup i, j : \mathbb{N} \mid i R_2 j : S_{\mathsf{d1}}^{i\mathsf{T}} \,;\, R_1 \,;\, S_{\mathsf{c1}}^{j}) \\ & = & (\bigcup i, j : \mathbb{N} \mid o_{\mathsf{d2}}^{\mathsf{T}} \,;\, S_{\mathsf{d2}}^{i} \,;\, R_2 \,;\, S_{\mathsf{c2}}^{j\mathsf{T}} \,;\, o_{\mathsf{c2}} \neq \mathsf{O} : S_{\mathsf{d1}}^{i\mathsf{T}} \,;\, R_1 \,;\, S_{\mathsf{c1}}^{j}) \end{array}$$

Size (type) of $R_1 \not\triangleright R_2$ = size of R_1

Erosion in finite grids: example





P

$$R_1 \ntriangleright R_2 \;=\; (\bigcap i, j : \mathbb{N} \mid i R_2 j : S_{\mathsf{d1}}^{i\mathsf{T}} \backslash R_1 / S_{\mathsf{c1}}^{j})$$

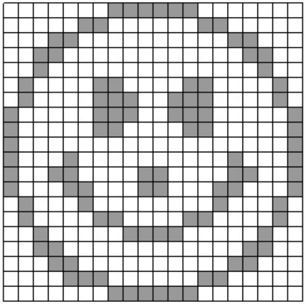
RelView program for erosion

$$R_1 \not\triangleright R_2 \;=\; \left(\bigcap i,j:\mathbb{N} \mid o_{\mathsf{d2}}^{\mathsf{T}} ; S_{\mathsf{d2}}^{i} ; R_2 ; S_{\mathsf{c2}}^{j\mathsf{T}} ; o_{\mathsf{c2}} \neq \mathsf{O} : S_{\mathsf{d1}}^{i\mathsf{T}} \backslash R_1 / S_{\mathsf{c1}}^{j}\right)$$

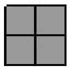
```
Erosion(R1, R2)
DECL Sd1, Sc1, Sd2, Sc2, od2, oc2,
     i1, j1, i2, j2, res, cond

BEG
Sd1 = succ(R1);
Sc1 = succ(R1^);
Sd2 = succ(R2);
Sc2 = succ(R2^);
od2 = init(On1(R2));
oc2 = init(On1(R2^));
i1 = I(Sd1);
j1 = I(Sc1);
i2 = I(Sd2);
j2 = I(Sc2);
res = L(R1);
WHILE -empty(i2) DO
  WHILE -empty(j2) DO
    cond = od2^*i2*R2*j2^*oc2
    IF -empty(cond) THEN
      res = res & (i1^\R1/j1) FI
    j1 = j1*Sc1;
    j2 = j2*Sc2 OD
  j1 = I(Sc1);
  j2 = I(Sc2);
  i1 = i1*Sd1;
  i2 = i2*Sd2 OD
RETURN res
END.
```

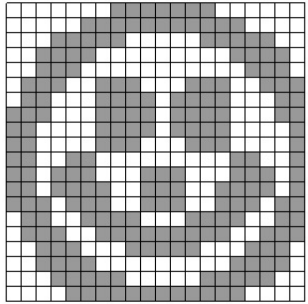
Screen captures of RelView (with a rather typical structuring element)



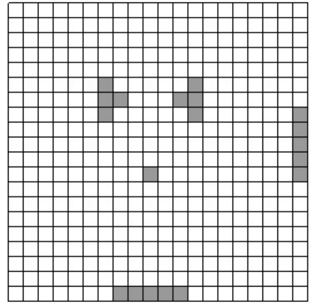
R



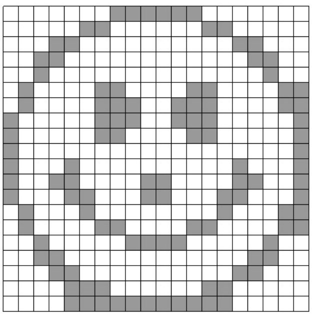
P



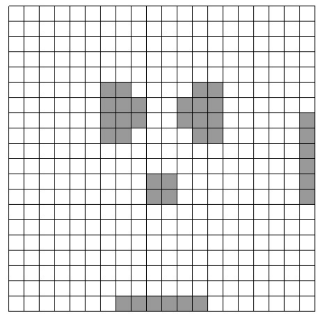
Dilation $R \rhd P$



Erosion $R \not\rhd P$



Closing $(R \rhd P) \not\rhd P$



Opening $(R \not\rhd P) \rhd P$

Matlab and Mathematica have primitive operations for computing dilation and erosion.
Matlab: `imdilate(R,P)`, `imerode(R,P)`
Mathematica: `Dilation(R,P)`, `Erosion(R,P)`

Performance comparisons: CPU times, small structuring element P

Size of R: $n \times n$. Time in seconds.

	Size of P: 3×3					
	Dilation $R \triangleright P$			Erosion $R \not\triangleright P$		
n	RelView	Matlab	Mathematica	RelView	Matlab	Mathematica
1000	2	.01	21	5	.01	20
2000	11	.02	81	25	.02	81
3000	26	.05	183	59	.05	182
4000	50	.09	327	119	.09	332
5000	85	.12	509	201	.12	509
6000	134	.20	731	315	.20	733
7000	197	.25	1007	418	.26	990
8000	246	.34	1310	545	.34	1314
9000	322	.42	1739	558	.44	1765
10000	469	.49	2180	1024	.53	2177

Performance comparisons: CPU times, larger structuring element P

Size of R: $n \times n$. Time in seconds.

	Size of P: 100×100					
	Dilation $R \triangleright P$			Erosion $R \not\triangleright P$		
n	RelView	Matlab	Mathematica	RelView	Matlab	Mathematica
100	3	.10	6	2	.10	6
150	5	.10	8	4	.10	8
200	10	.10	10	6	.10	10
250	19	.14	13	11	.13	13
300	35	.18	16	19	.17	16
350	58	.23	19	36	.23	19
400	84	.22	23	60	.22	23
450	107	.27	26	112	.27	26
500	150	.34	30	199	.34	30
550	213	.38	34	365	.38	34

5 Graph morphology

Our starting point

- Heijmans, H. J. A. M., Nacken, P., Toet, A., Vincent, L. Graph morphology. Journal of Visual Communication and Image Representation 3(1), 24–38 (1992).

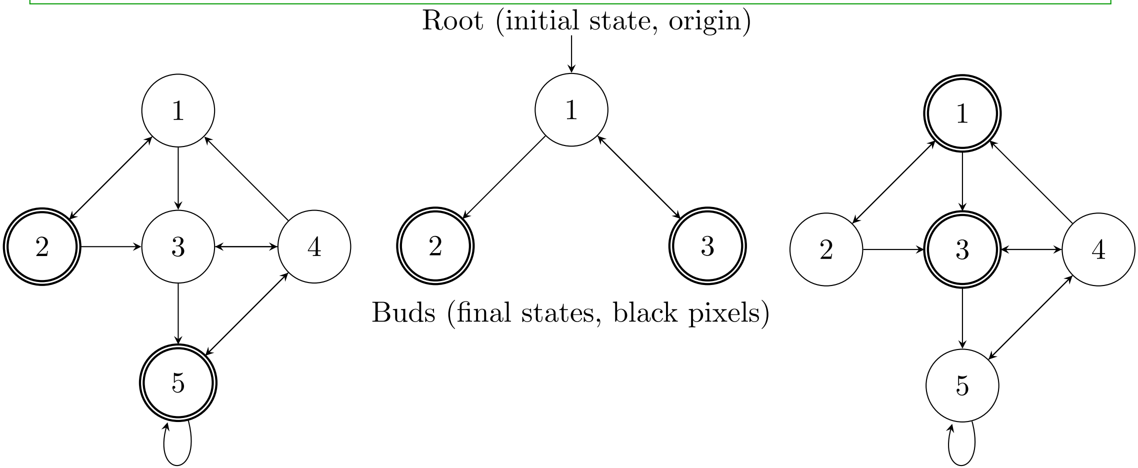
- Heijmans, H. J. A. M., Vincent, L. Graph morphology in image analysis. In Mathematical Morphology in Image Processing, Dougherty, E. (ed.), Marcel-Dekker, 171–203 (1992).

They deal with nondirected graphs with nodes weighted by grey-level values.

Our RelView implementation

Directed binary graphs (i.e., relations).

Graph dilation: example



Source graph R Structuring graph P (Pattern) Dilated graph $R \triangleright P$

Same procedure as for images

- Embed P in R, with a root of P on a bud of R.
- The buds of $R \triangleright P$ are the embeddings of the buds of P.

(**Root, bud:** Heijmans et al. terminology)

Differences with the case of images

1. **Origin/roots**
   - **Images:** the structuring image has a single origin.
   - **Graphs:** the structuring graph may have many roots.
2. **Embeddings**
   - **Images:** there is exactly one embedding (due to the regular grid and the fixed orientation of axes), except possibly none close to a border.
   - **Graphs:** there may be zero to many anywhere in the source graph.

Definitions

- **Predicate ih:**

$$\mathsf{ih}(\sigma, Q, R) \;\Leftrightarrow\; \sigma\,;\sigma^{\mathsf{T}} = \mathsf{I} \wedge \sigma^{\mathsf{T}};\sigma \subseteq \mathsf{I} \wedge \sigma^{\mathsf{T}};Q\,;\sigma \subseteq R$$

  i.e., $\mathsf{ih}(\sigma, Q, R)$ means that σ is an injective mapping that homomorphically maps Q inside R.

- **Graph G:** 4-tuple (V, R, r, b), where
  - V is the set of vertices,
  - $R : V \leftrightarrow V$ is a homogenous relation,
  - $r : V$ is a vector representing the roots of G (irrelevant for the source graph),
  - $b : V$ is a vector representing the buds of G.

Dilation $G_1 \triangleright G_2$ of graph G_1 by graph G_2

$$G_1 \triangleright G_2 \; = \; (V_1,\; R_1,\; r_1,\; (\bigcup \sigma \mid \mathsf{ih}(\sigma, R_2, R_1) \wedge \sigma^{\mathsf{T}} ; r_2 \cap b_1 \neq \mathsf{O} : \sigma^{\mathsf{T}} ; b_2))$$

Thus

- $G_1 \triangleright G_2$ is the same graph as G_1, except for the buds;
- $\mathsf{ih}(\sigma, R_2, R_1)$ means that the relation of the structuring graph G_2 is embedded in the relation of the source graph G_2 by the injective homomorphism σ;
- $\sigma^{\mathsf{T}} ; r_2 \cap b_1 \neq \mathsf{O}$ means that at least one root of G_2 is mapped to at least one bud of G_1;
- the buds of the dilated graph is the union of sets $\sigma^{\mathsf{T}} ; b_2$ which are the mappings by σ of the buds of G_2.

RelView program for graph dilation

```
Prev(p)
{
Returns the predecessor of point p.
}
BEG
  RETURN succ(p)*p
END.



Simulation(P,rP,R,rR)
{
Returns the largest simulation such that R simulates P
  and R^T simulates P^T. rP and rR are vectors giving
  the roots of P and R.
}
DECL sim, temp
BEG
  sim = L(Ln1(P)*L1n(R));
  temp = O(sim);
  WHILE -eq(sim,temp)
    DO
      temp = sim;
      sim = (rP^ \ rR^) & (P \ (sim * R)) & (P^ \ (sim * R^))
    OD
  RETURN sim
END.
```

```
DilGr(P,rP,bP,R,bR)
{Returns the dilation of graph R by graph P.
  Column vectors:
    rP : roots of P,
    bP : buds of P,
    bR : buds of R.
}
DECL inj, sim, simcop, i, Ln1R, LP, Lsim, res, pbR,
  plignesim, lignesim, bRcop, at, rPcop, prP
BEG
  LP = L(P);
  Ln1R = Ln1(R);
  Lsim = L(Ln1(P)*L1n(R));
  rPcop = rP;
  res = O(bR);
  WHILE -empty(rPcop) { Loop on the roots of P }
    DO
    prP = point(rPcop);
    rPcop = rPcop & -prP;
    bRcop = bR;
    WHILE -empty(bRcop) { Loop on the buds of R }
      DO
      pbR = point(bRcop);
      bRcop = bRcop & -pbR;
      sim = Simulation(P,prP,R,pbR);
      simcop = sim;
```

```
inj = O(sim);
IF eq(sim*Ln1R,Ln1(P))
  THEN {sim total; if not total, no homomorphism
        can be extracted from it}
    plignesim = init(Lsim);
    {next loop: combinatorial extraction of injective
     homomorphisms from sim}
    WHILE -empty(plignesim)
      DO
        lignesim = simcop & plignesim & -(LP * inj);
        IF empty(lignesim)
          THEN
            simcop = simcop | (sim & plignesim);
            plignesim = Prev(plignesim);
            inj = inj & -plignesim
          ELSE
            at = atom(lignesim);
            simcop = simcop & -at;
            inj = inj | at;
            IF empty(next(plignesim))
              THEN
                IF incl(inj^ * P * inj, R)
                  THEN res = res | inj^*bP FI;
                inj = inj & -plignesim
              ELSE plignesim = next(plignesim)
            FI
        FI
      OD
```

```
      FI;
      OD
    OD
  RETURN res
END.
```

Dilation of G_1 by G_2: CPU times with RelView

	Size of $G_2 = |V_2| = 3$							
Size of $R_1 = |V_1|$	50	100	150	200	250	300	350	400
Time	0.3	2.7	12	31	66	105	195	244

	Size of $G_2 = |V_2| = 4$							
Size of $R_1 = |V_1|$	10	20	30	40	50	100	110	120
Time	0.02	0.3	1.4	2.5	10	242	439	656

	Size of $G_2 = |V_2| = 5$							
Size of $R_1 = |V_1|$	10	20	30	35	40	45	50	55
Time	0.03	4.1	44	71	106	306	504	1085

Left erosion $G_1 \not\triangleright G_2$ of graph G_1 by graph G_2

As for images, left erosion is related to dilation by a Galois connection

$$R_1 \triangleright R_2 \subseteq R_3 \Leftrightarrow R_1 \subseteq R_3 \not\triangleright R_2$$

Shown in the paper

$$G_1 \not\triangleright G_2 \;=\; (G_1^{-} \triangleright G_2^{\leftrightarrow})^{-} \tag{1}$$

where for $G = (V, R, r, b)$,

- $G^{-} = (V, R, r, \overline{b})$ complements the bud vector b;
- $G^{\leftrightarrow} = (V, R, b, r)$ exchanges the roots and the buds.

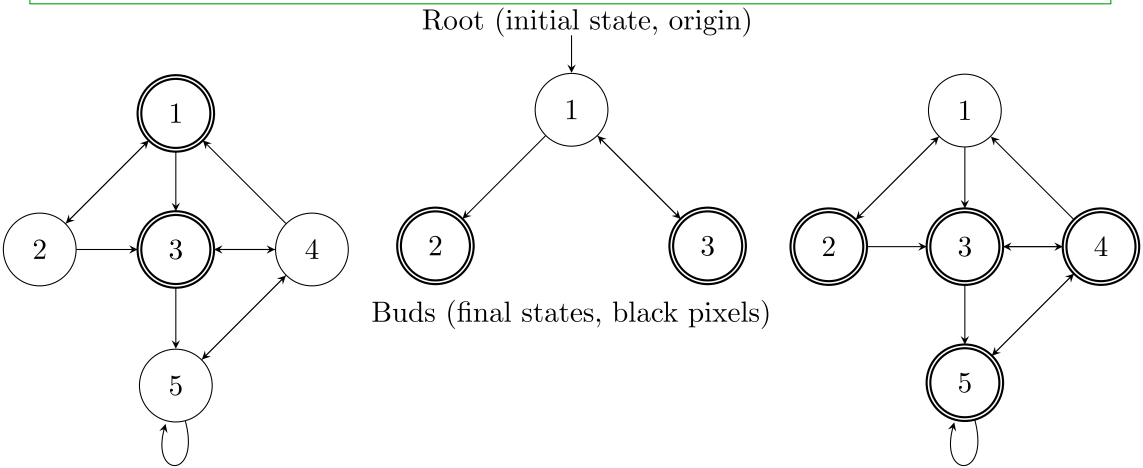
Notice the similarity of (1) with the relational law of the left residual

$$Q/R = \overline{\overline{Q}\,;R^{\mathsf{T}}}.$$

Also

$$G_3 \not\triangleright G_2 \;=\; (V_3, R_3, r_3, (\bigcap \sigma \mid \mathsf{ih}(\sigma, R_2, R_1) : \overline{\sigma^{\mathsf{T}} ; r_2 ; b_2^{\mathsf{T}} ; \sigma ; \overline{b_3}})).$$

Graph erosion: example



Source graph R Structuring graph P Eroded graph $R \ntriangleright P$

Intuitive way to get the result when there is a single root (as here)

- Embed P in R, with buds of P included in buds of R.
- The node where the root of P is mapped becomes a bud of the eroded graph.
- If there is no embedding mapping the root to a given node (e.g., node 5), that node is a bud of the eroded graph.

6 Conclusion

Exploiting the flexibility of RelView: other geometries

First quadrant of Cartesian coordinates

$$\underset{S_{\mathsf{d}}}{\begin{bmatrix}0&0&0\\1&0&0\\0&1&0\end{bmatrix}}\quad \underset{S_{\mathsf{c}}}{\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\0&0&0&0\end{bmatrix}}\quad \underset{o_{\mathsf{d}}}{\begin{bmatrix}0\\0\\1\end{bmatrix}}\quad \underset{o_{\mathsf{c}}}{\begin{bmatrix}1\\0\\0\\0\end{bmatrix}}\quad \underset{O}{\begin{bmatrix}0&0&0&0\\0&0&0&0\\1&0&0&0\end{bmatrix}}$$

S_{d}	S_{c}	o_{d}	o_{c}	O
successor domain side	successor codomain side	origin domain side	origin codomain side	origin

Vertical cylinder

$$\underset{S_{\mathsf{d}}}{\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix}}\quad \underset{S_{\mathsf{c}}}{\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}}$$

Horizontal cylinder

$$\underset{S_{\mathsf{d}}}{\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}}\quad \underset{S_{\mathsf{c}}}{\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\0&0&0&0\end{bmatrix}}$$

Torus

$$\underset{S_{\mathsf{d}}}{\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}}\quad \underset{S_{\mathsf{c}}}{\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}}$$

Exploiting the flexibility of RelView: higher dimensions

Recall the general form for a symbolic expression for a concrete relation

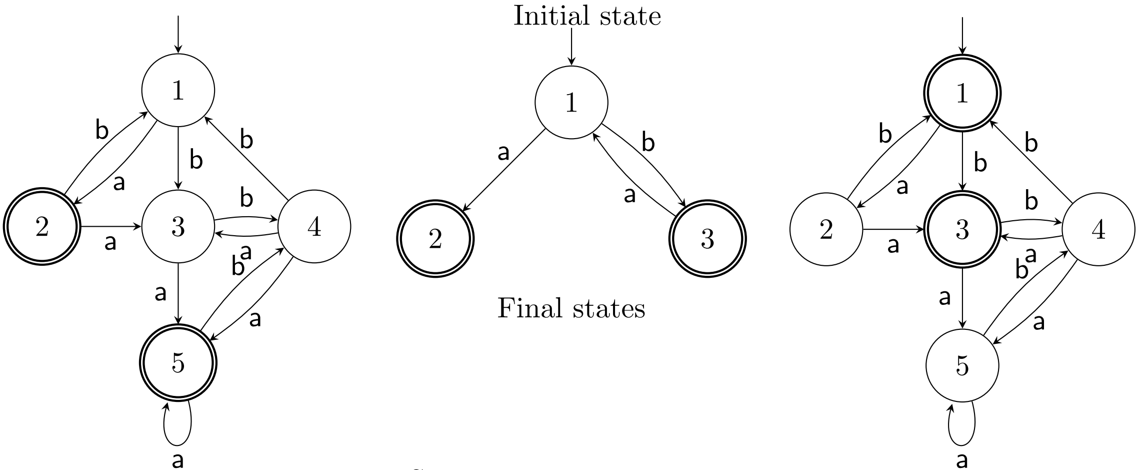
$$R = (\bigcup i,j \mid iRj : S_{\mathsf{d}}^{i\mathsf{T}} \,;\, O \,;\, S_{\mathsf{c}}^{j})$$

There is also a vectorised form

$$\mathsf{vec}(R) \quad = \quad (\bigcup i,j \mid iRj : [S_{\mathsf{d}}^{i\mathsf{T}},\, S_{\mathsf{c}}^{j\mathsf{T}}]) \;;\; [o_{\mathsf{d}}, o_{\mathsf{c}}\rangle$$

This is a nice form that is easily extended to n-ary relations, to which morphological operations can be applied.

Future research: morphological operations for automata



Source automaton R — Structuring automaton P — Dilated automaton $R \triangleright P$

Same procedure as for graphs (dilation illustrated above), except that the embedding must also preserve labels

- Given the languages of R and P, what is the language of $R \triangleright P$.
- This is easy when R and P are total and have no unreachable nodes. What about the general case?