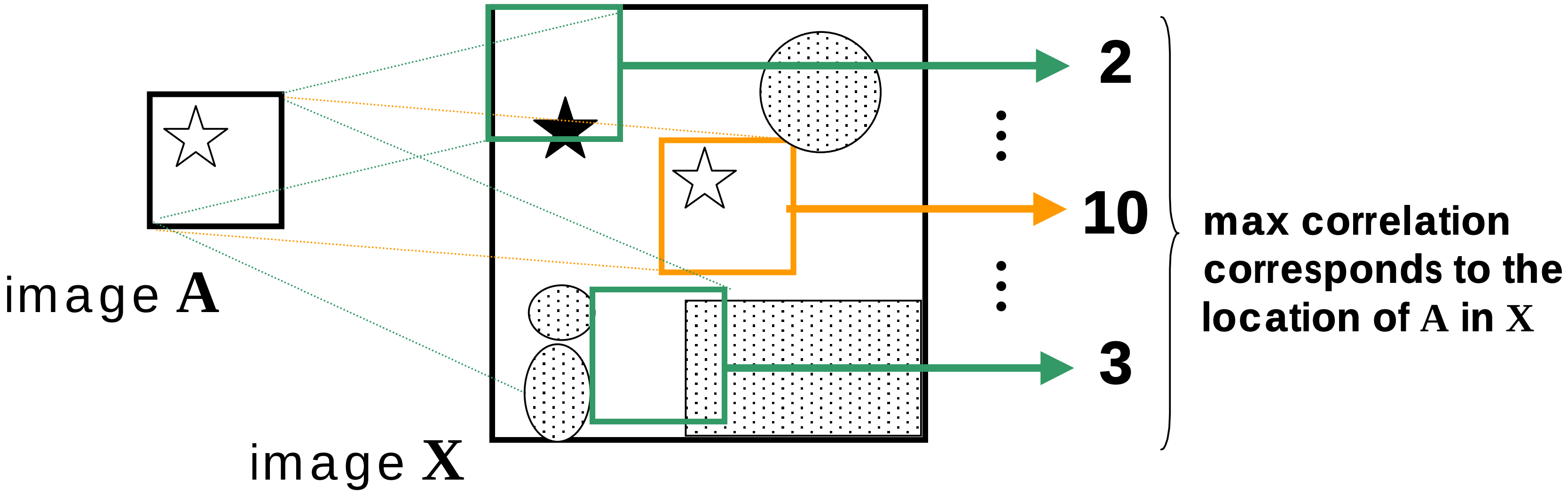
An Efficient Correlation Computation Method for Binary Images Based on Matrix Factorisation

R.Bogush, S.Maltsev (Polotsk State University, Belarus)
S.Ablameyko (Institute of Engineering Cybernetics, Belarus)
S.Uchida, S.Kamata (Kyushu University, Japan)

Overview

- Provide an efficient algorithm for the computation of the correlation values between binary images



- Utilize a matrix factorisation technique

Correlation computation based on direct method (1)

$$\mathbf{A} = \begin{bmatrix} 1&1&1&1&1&1 \\ 0&1&1&1&1&0 \\ 0&0&1&1&0&0 \\ 0&0&1&1&0&0 \\ 0&1&1&1&1&0 \\ 1&1&1&1&1&1 \end{bmatrix}$$

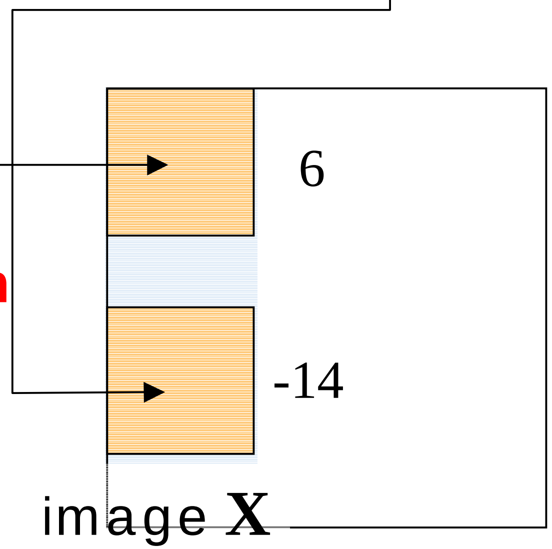
$$\mathbf{X} = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&0&0&0&0&0&1&1&1&1&1&1 \\ 0&0&0&1&1&0&0&0&0&0&0&1&1&1&1&0 \\ 0&0&1&1&1&1&0&0&0&0&0&0&1&1&0&0 \\ 0&1&1&1&1&1&1&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&1&1&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&1&1&1&1&0&0&0&0&0 \end{bmatrix}$$

$\mathbf{X}_{sub}$

note: “0” plays as “-1” in the followings

Correlation computation based on direct method (2)

$$
\mathbf{AX}_{\mathrm{sub}}^{T} =
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}^{T}
=
\begin{bmatrix}
-\mathbf{6} & 4 & 2 & -2 & -2 & 2 & 4 & -6 & -\mathbf{6} & -6 & -4 & -6 & -6 & -6 & -6 & -4 \\
-2 & \mathbf{4} & 2 & 2 & 2 & 2 & 4 & -2 & -2 & -\mathbf{2} & -4 & -2 & -2 & -2 & -2 & -4 \\
2 & 0 & \mathbf{2} & 2 & 2 & 2 & 0 & 2 & 2 & 2 & \mathbf{0} & 2 & 2 & 2 & 2 & 0 \\
2 & 0 & 2 & \mathbf{2} & 2 & 2 & 0 & 2 & 2 & 2 & 0 & \mathbf{2} & 2 & 2 & 2 & 0 \\
-2 & 4 & 2 & 2 & \mathbf{2} & 2 & 4 & -2 & -2 & -2 & -4 & -2 & -\mathbf{2} & -2 & -2 & -4 \\
-6 & 4 & 2 & -2 & -2 & \mathbf{2} & 4 & -6 & -6 & -6 & -4 & -6 & -6 & -\mathbf{6} & -6 & -4
\end{bmatrix}
$$



each diagonal sum corresponds to a correlation value

Binary matrix factorisation (1)

Lemma: Any binary matrix A can represented in the form of multiplication:

$$A=DB$$

where B is a matrix obtained from matrix A after deleting repeated and inverse strings, and D is a matrix with 1 (repeat) and 0 (inverse).

Example:

$$\underset{\mathbf{A}}{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}} = \underset{\mathbf{D}}{\begin{bmatrix} 1 & \\ & 1 \\ & 0 \\ & 1 \\ & 0 \end{bmatrix}} \underset{\mathbf{B}}{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}}$$

repetitive use of 1st stroke of **B**

inverse use of 2nd stroke of **B**

Binary matrix factorisation (2)

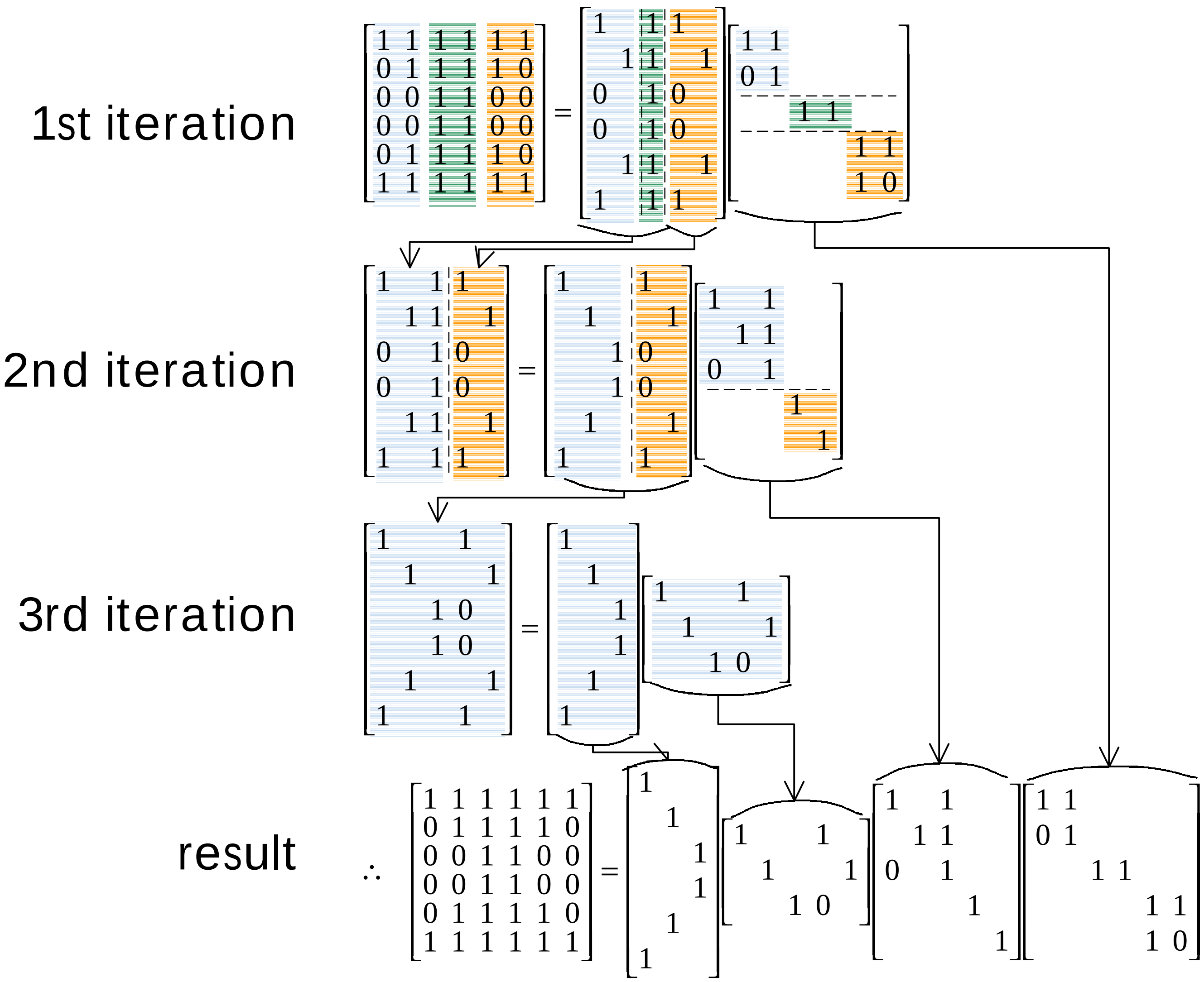
Theorem: Arbitrary binary matrix matrix $\mathbf{A}$ can be represented as multiplication of sparse block-diagonal submatrixes with no more then two info symbols in every stroke.

Example:

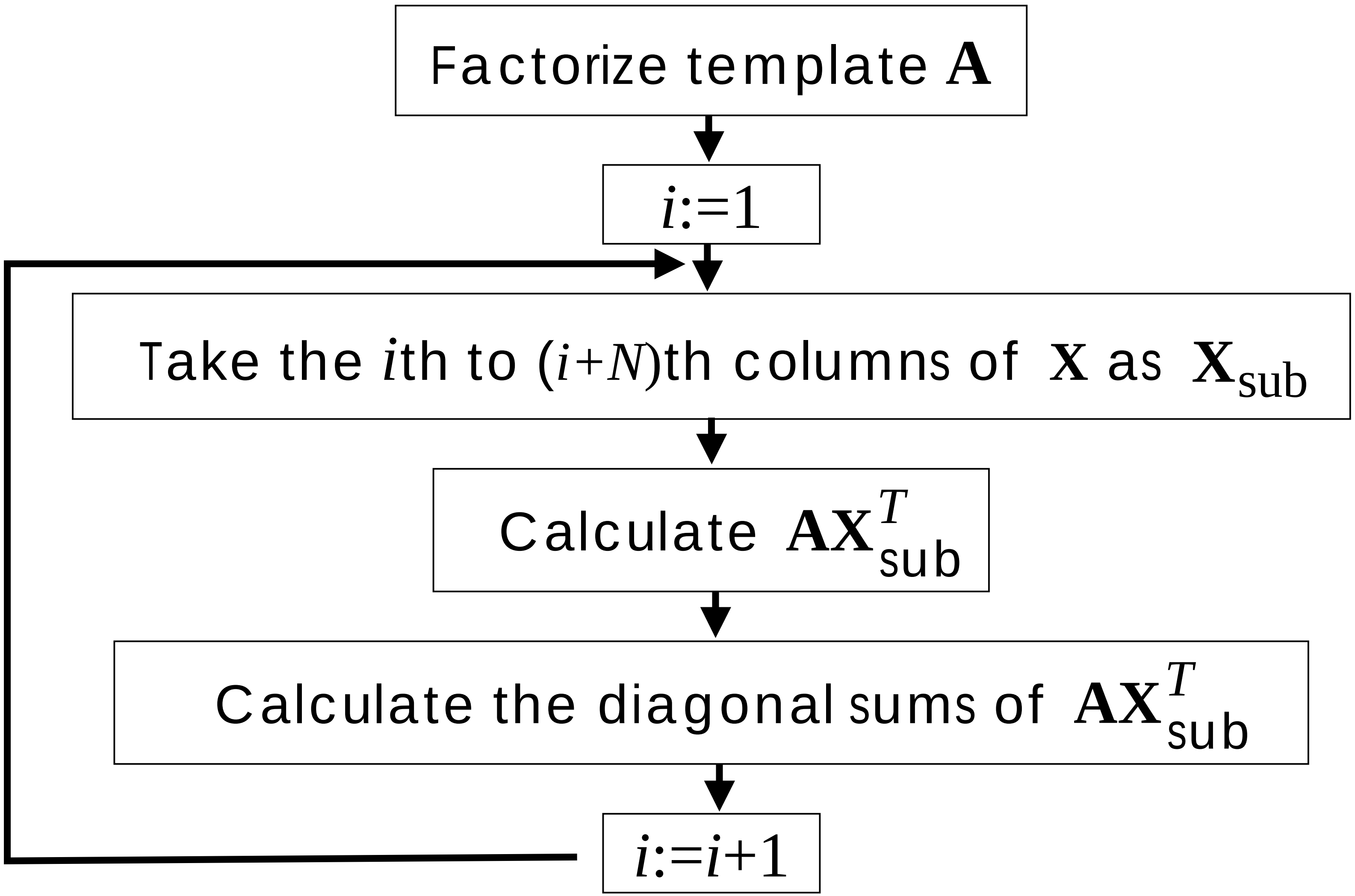
$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \\ & & 1 \\ & 1 & \\ 1 & & \end{bmatrix} \begin{bmatrix} 1 & & & 1 & \\ & 1 & &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

Outline of a proof : Local and iterative application of Lemma (as shown in the next slide).

Algorithm for matrix factorisation : An example



Correlation computation based on matrix factorisation



Comparison of the number of operations (1)

direct method

$$
\mathbf{AX}^{T}_{\mathrm{sub}} = \begin{bmatrix} 1&1&1&1&1&1 \\ 0&1&1&1&1&0 \\ 0&0&1&1&0&0 \\ 0&0&1&1&0&0 \\ 0&1&1&1&1&0 \\ 1&1&1&1&1&1 \end{bmatrix} \begin{bmatrix} 0&0&0&0&0&0 \\ 0&1&1&1&1&1 \\ 0&0&1&1&1&1 \\ 0&0&0&1&1&0 \\ 0&0&0&1&1&0 \\ 0&0&1&1&1&1 \\ 0&1&1&1&1&1 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&1 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&1 \end{bmatrix}^{T}
$$

total:
330(=30x11) operations

present method

$$
\mathbf{AX}^{T}_{\mathrm{sub}} = \begin{bmatrix} 1&&\\ &1&\\ &&1\\ &&1\\ &1&\\ 1&& \end{bmatrix} \begin{bmatrix} 1&&&1&\\ &1&&&1\\ &&1&0& \end{bmatrix} \begin{bmatrix} 1&&1&&\\ &1&1&&\\ 0&&1&&\\ &&&1&\\ &&&&1 \end{bmatrix} \begin{bmatrix} 1&1&&&&\\ 0&1&&&&\\ &&1&1&&\\ &&&&1&1\\ &&&&1&0 \end{bmatrix} \begin{bmatrix} 0&0&0&0&0&0 \\ 0&1&1&1&1&1 \\ 0&0&1&1&1&1 \\ 0&0&0&1&1&0 \\ 0&0&0&1&1&0 \\ 0&0&1&1&1&1 \\ 0&1&1&1&1&1 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&1 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&1 \end{bmatrix}^{T}
$$

5x16 op.

3x16 op.

3x16 op.

0x16 op.

total:
176(=16x11) operations

Comparison of the number of operations (2)

Our algorithm based on matrix factorisation	10 240	31 795	68 812	126 720	379 200	463 667	2.6E+6	3.1E+6
Direct method	31 744	108 288	258 040	505 600	1.7E+6	2.1E+6	1.4E+7	1.7E+7
Nussbaumer polynomial algorithm and split-radix FFT	35 516	-	174 780	-	-	830 140	-	3.8E+6
Nussbaumer algorithm	3 580	-	194 820	-	-	948 100	-	4.7E+6
Polynomial transform algorithm	35 828	-	196 212	-	-	955 342	-	-
Winograd algorithm	-	80 832	-	-	642 024	-	3.3E+6	-
Rader-Brenner algorithm	46 336	-	241 152	-	-	1.2E+6	-	5.6E+6
n(=N=M) (matrix size)	32	48	64	80	120	128	240	256

Note : It is assumed that the complexity of the one multiplication operation is equivalent to three plus/minus operations complexity.