

Tree-structured knowledge in a distributed intelligent MEMS application

¹⁾Atsushi Sato, ²⁾Eugen Dedu, ²⁾Julien Bourgeois, ¹⁾Runhe Huang
¹⁾Hosei University, ²⁾UFC/FEMTO-ST

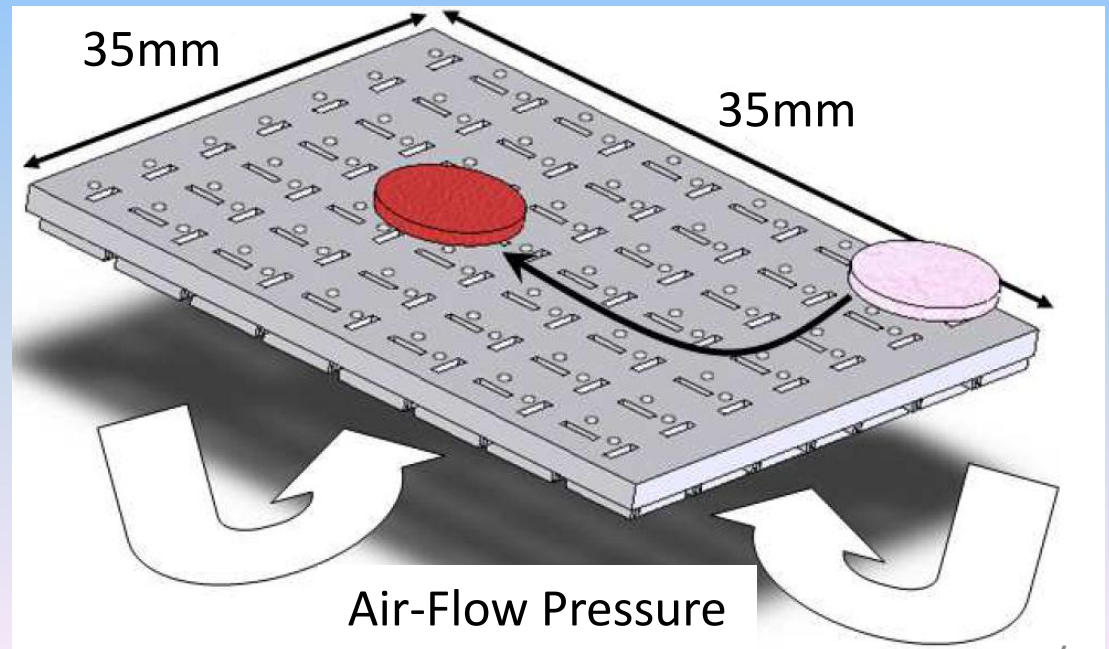
Table of Contents

- Introduction
 - Smart Surface
 - Main issues
- Theory
 - Tree-structured knowledge (TSK)
 - Reconstruction of the object with TSK
 - Differentiation of the object with TSK
- Analyses
- Conclusions and future work

Smart Surface

MEMS-arrayed manipulation surface

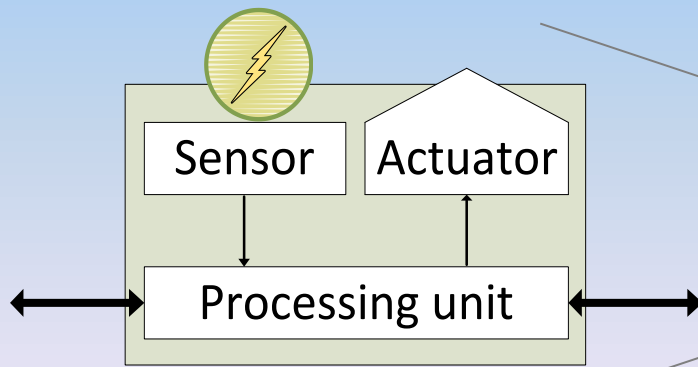
- Recognition
- Conveyance
- Positioning



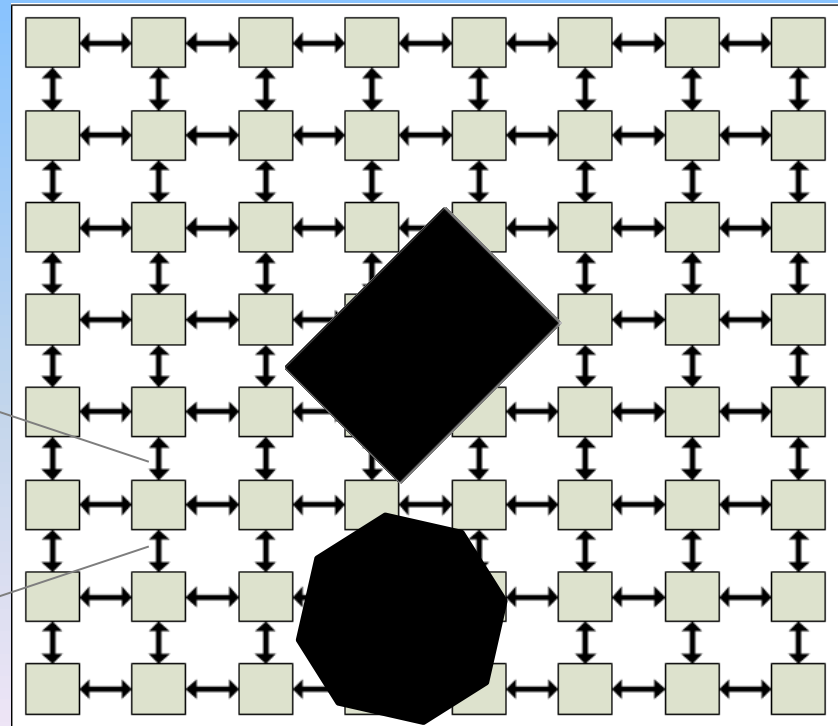
Distributed control

MEMS

- Sense
- Act
- Decide
- Communicate

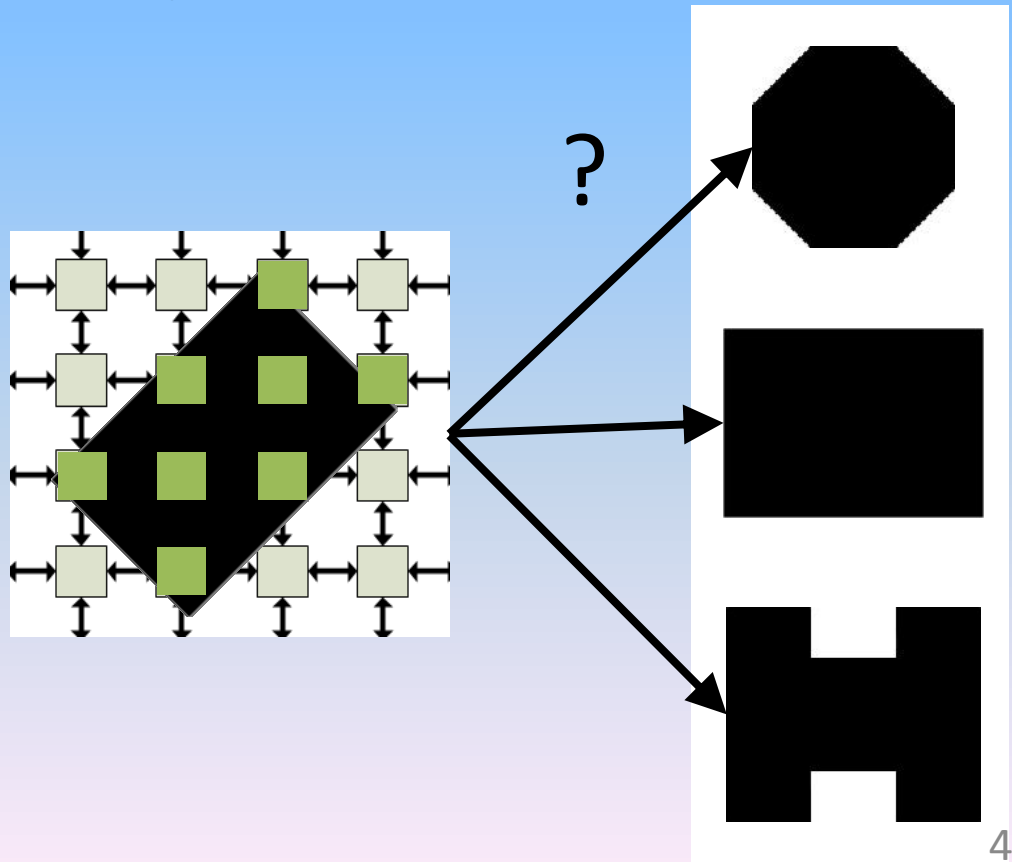


MEMS



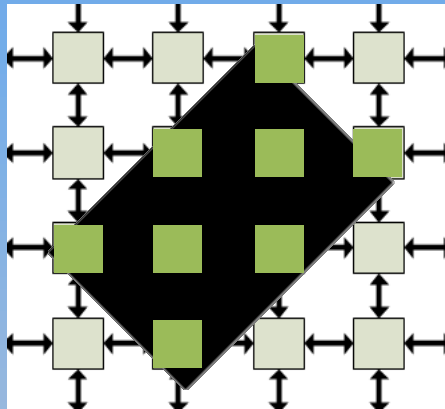
Recognition

- Offline stage
 - Create database of shapes of models
- Online stage
 - Reconstruction
 - Differentiation



Offline stage (the previous approach)

Repeat



Rotate and translate
the object on the Smart Surface

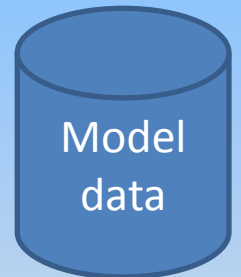
Matrix

0010
0111
1110
0100

Criteria

A: 10
P: 16
S: 8
⋮

Database

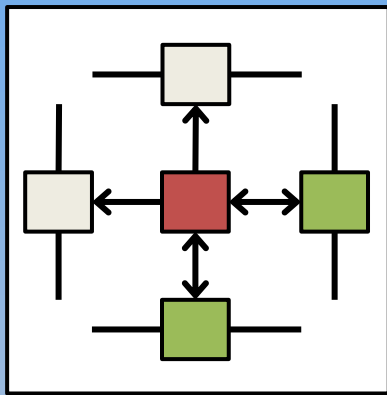


Database is uploaded to every cell

Online stage (the previous approach)

Repeat

- Reconstruction phase



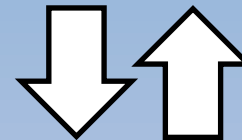
```
00000000  
00000000  
01000000  
00000000  
00000000  
00000000  
00000000
```



```
00000000  
00000000  
01100000  
01000000  
00000000  
00000000  
00000000
```



```
00000000  
00100000  
01110000  
11100000  
01000000  
00000000  
00000000
```



failure



success

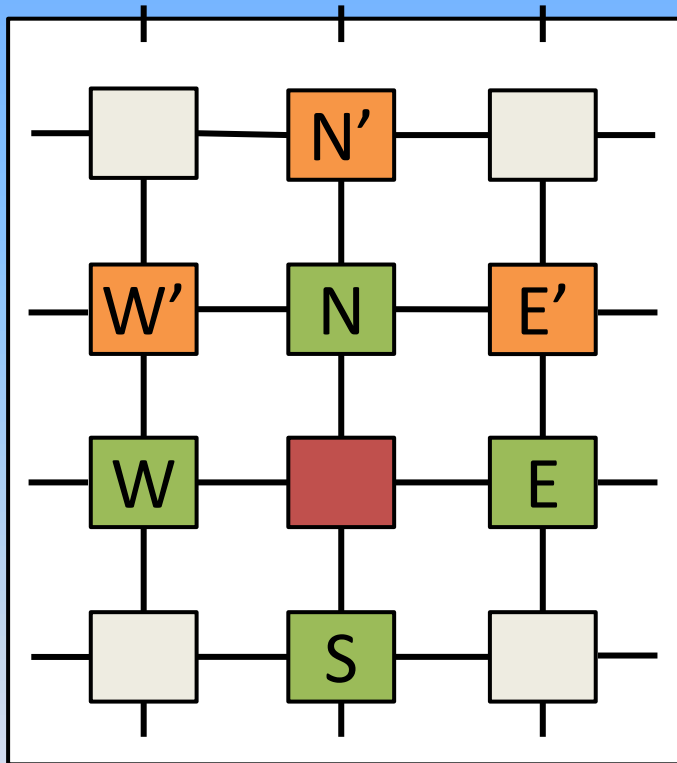
- Differentiation phase
 - Calculate criteria
 - Compare with database

Main issues

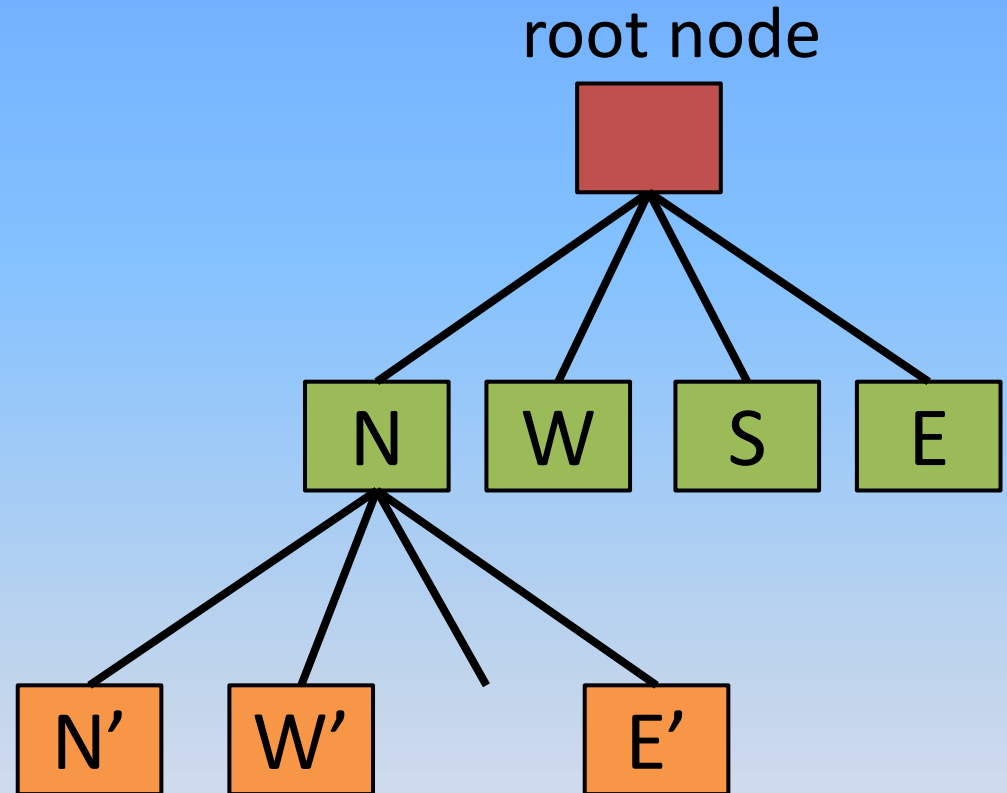
- Message size is the same as the Smart Surface
-> redundant
- excessively comparison
-> there is no trigger

→ Relative position based representation

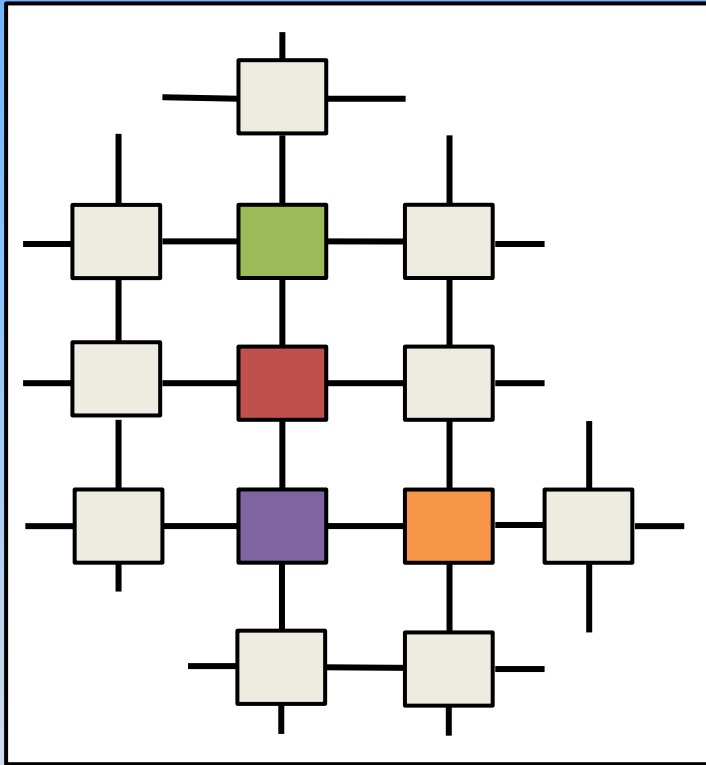
Tree-structured knowledge (our current approach)



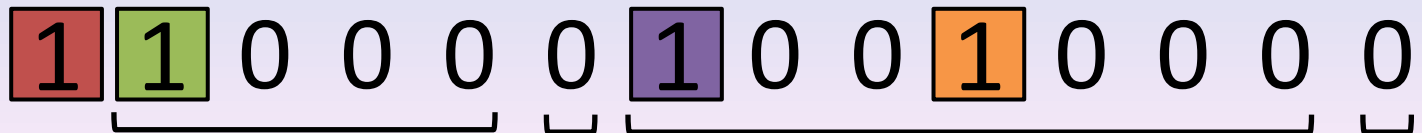
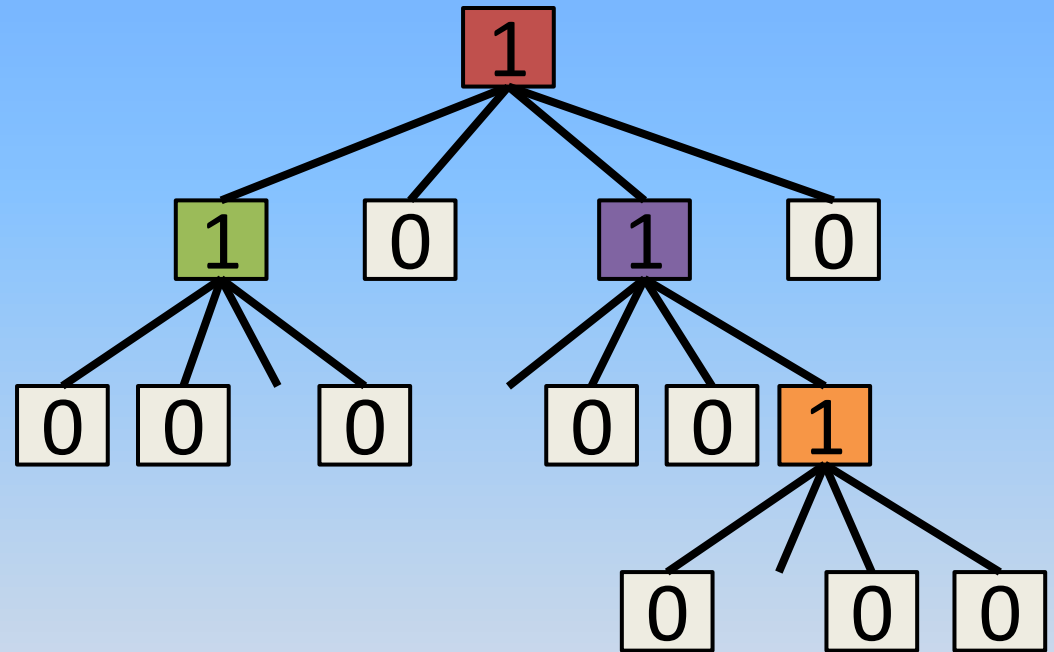
Smart Surface



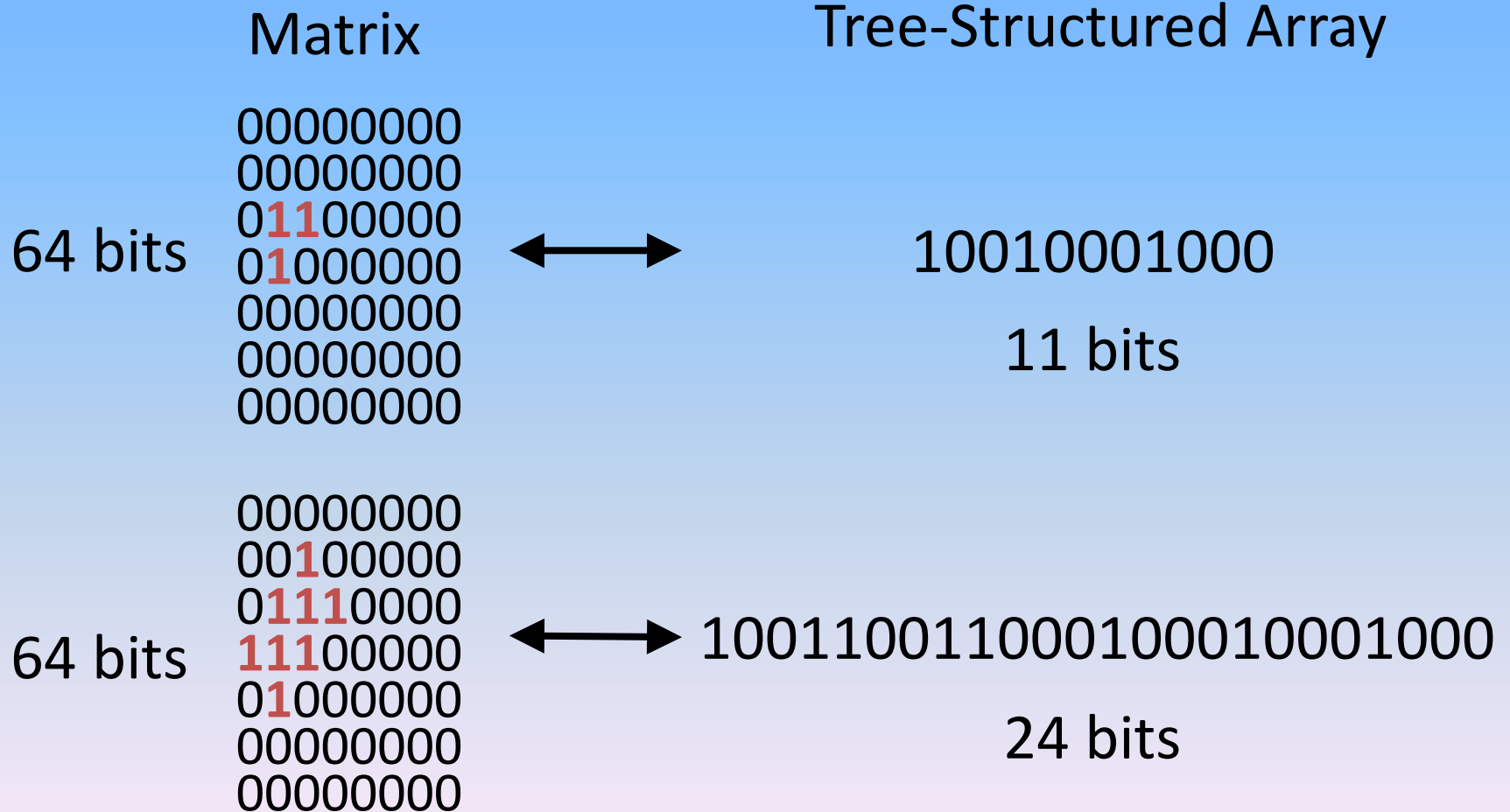
Tree-structured array



Smart Surface

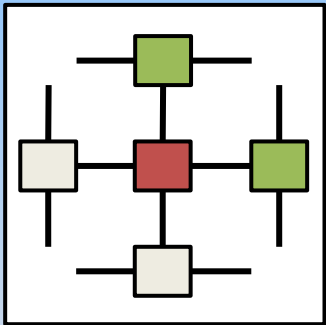


Matrix VS. TSK



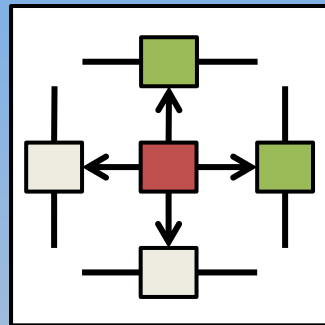
Reconstruction

① Initialize its array

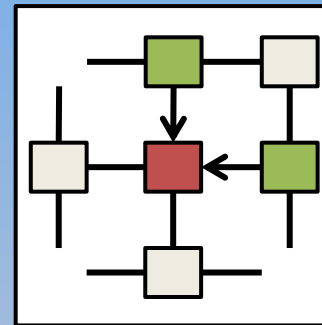


Repeat

② Generate and send messages



③ Receive and merge messages

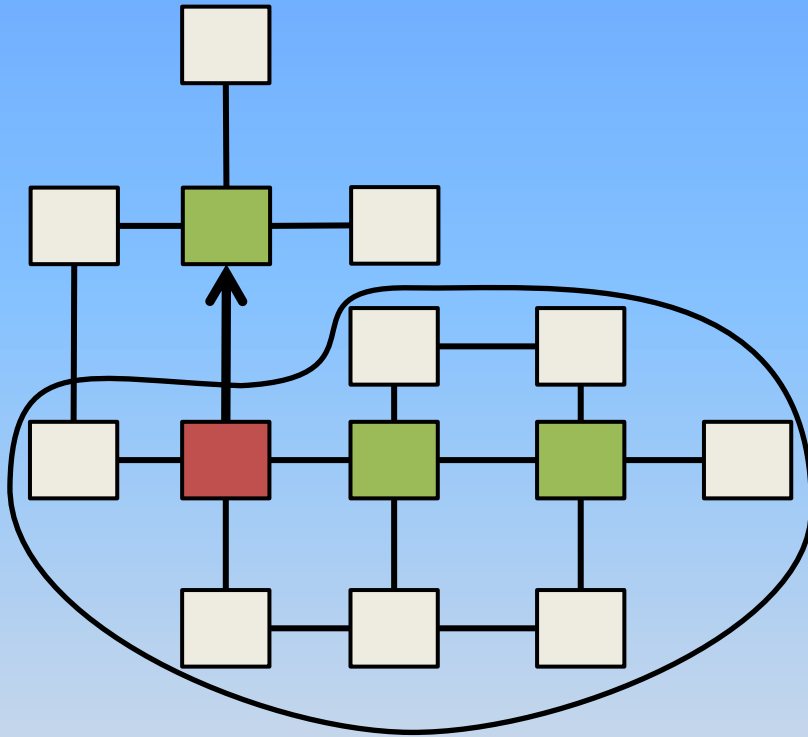


④ Check duplication

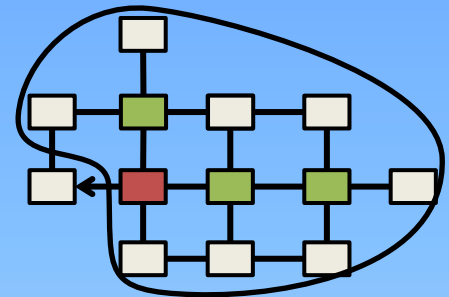


Differentiation phase

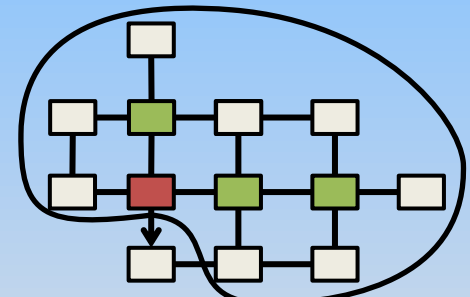
Generate messages



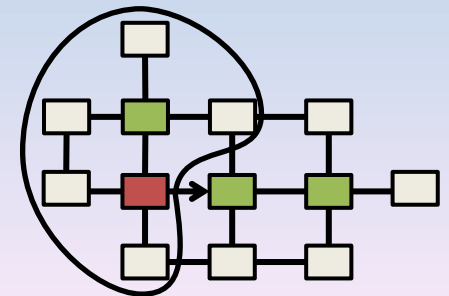
1 1 0 0 0 0 0 0 1 0 0 1 0 0 0



1 1 0 0 0 0 1 0 0 1 0 0 0

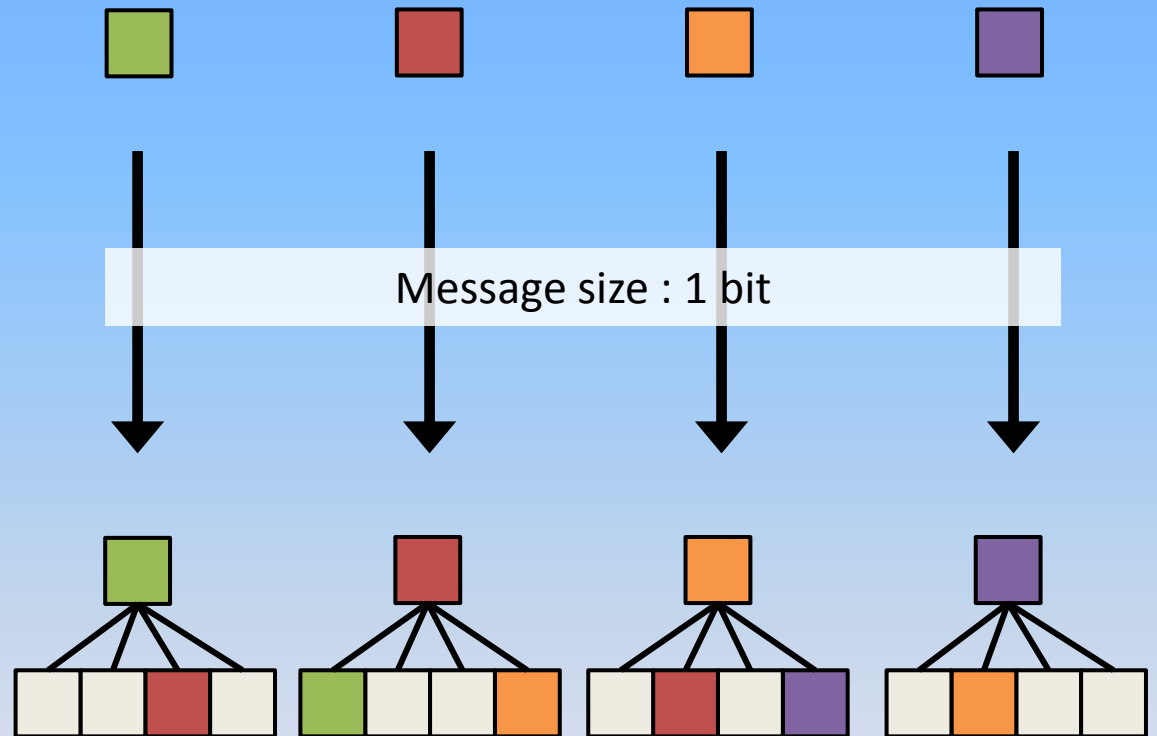
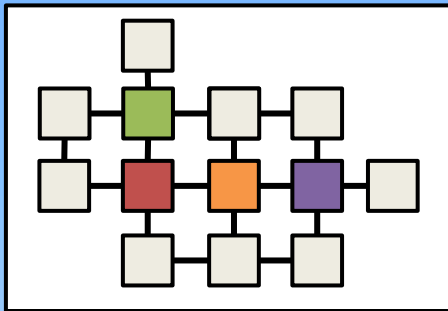


1 1 0 0 0 0 1 0 0 1 0 0 0

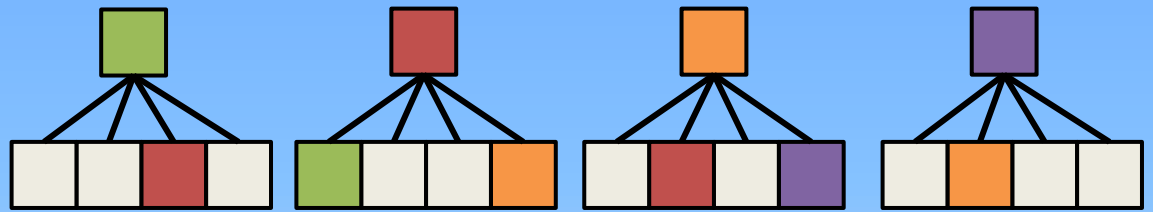
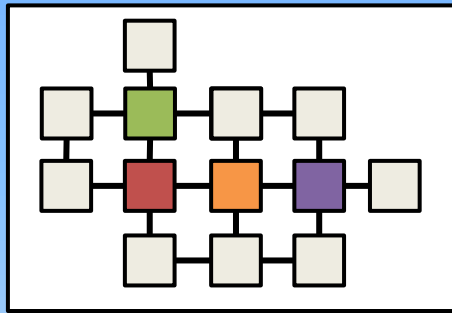


1 1 0 0 0 0 0 0

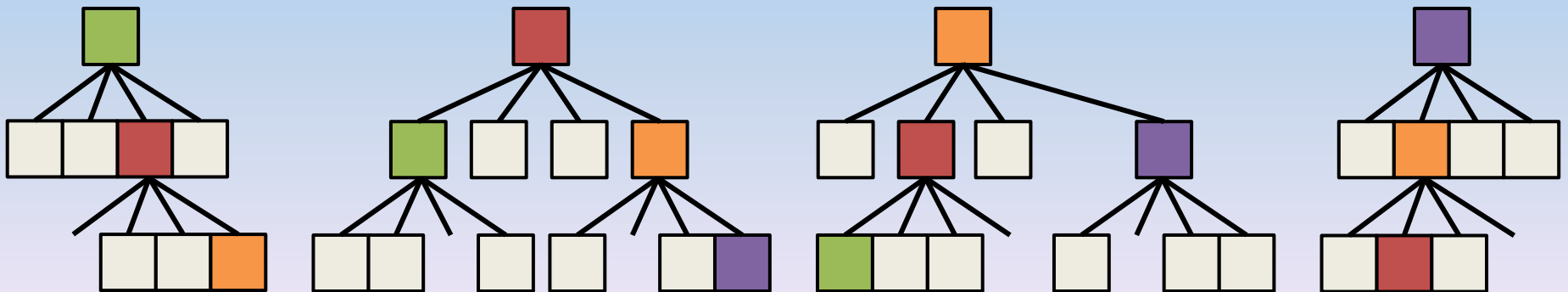
Merge messages(1/3)



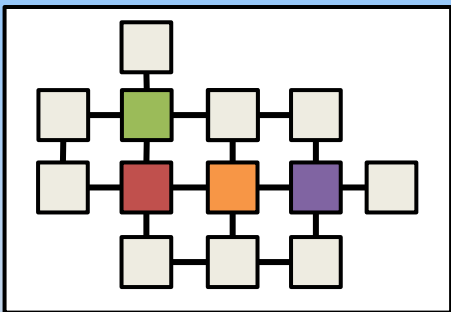
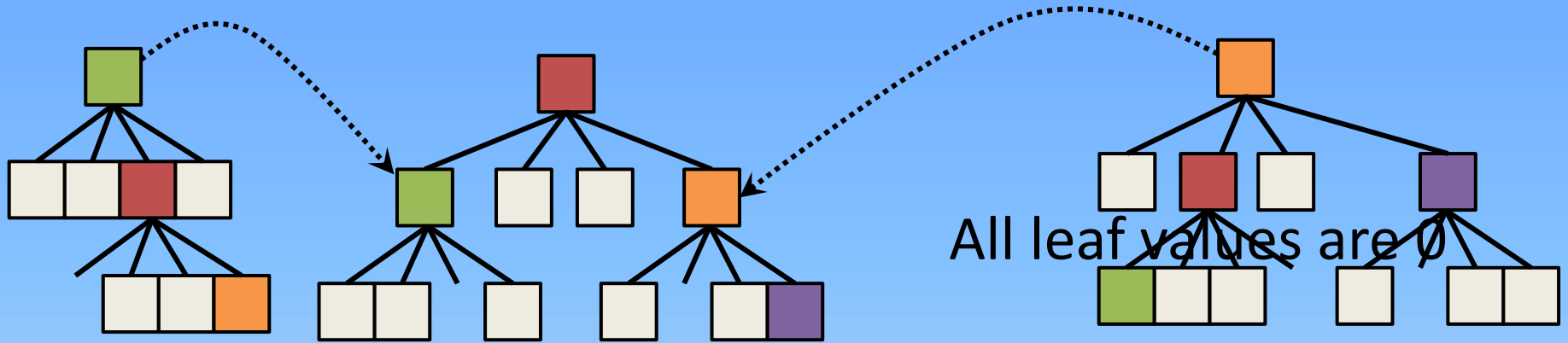
Merge messages(2/3)



Message size : 4 bits

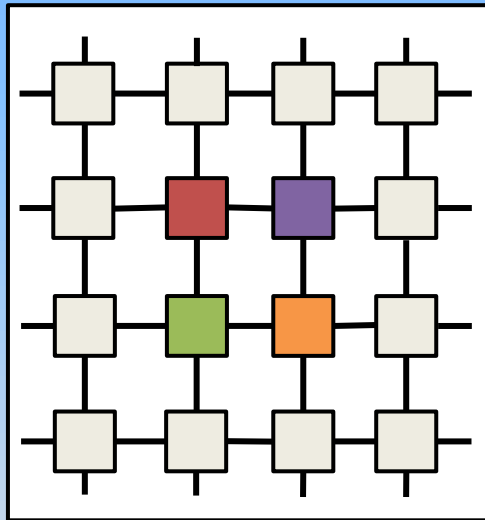


Merge messages(3/3)



Go to
differentiation phase

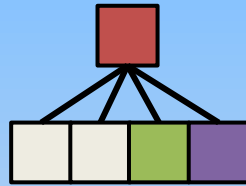
Duplication check



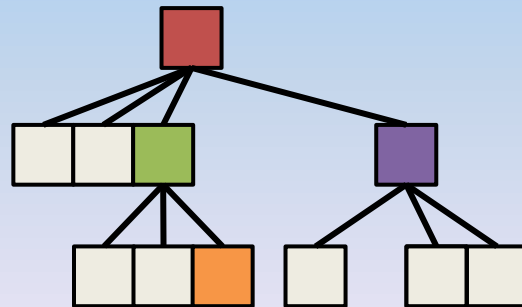
Smart Surface



1 0 0 0 0



1 0 0 1 1



1 0 0 1 0 0 1 1 0 1 0



1 0 0 1 0 0 1 1 0 0 0

Differentiation

① Transform its tree to the regular form

Repeat

- Change the root to the north
- Change the root to the west

Until the root is most northern and western

② Compare the array with all the shapes in database

if (discover the same array)

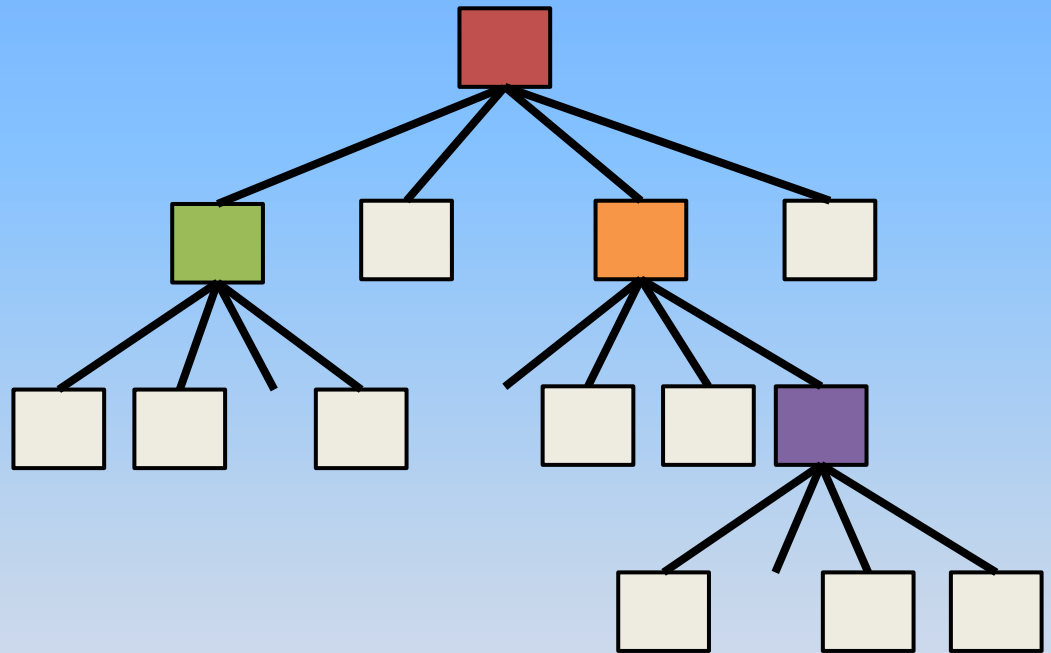
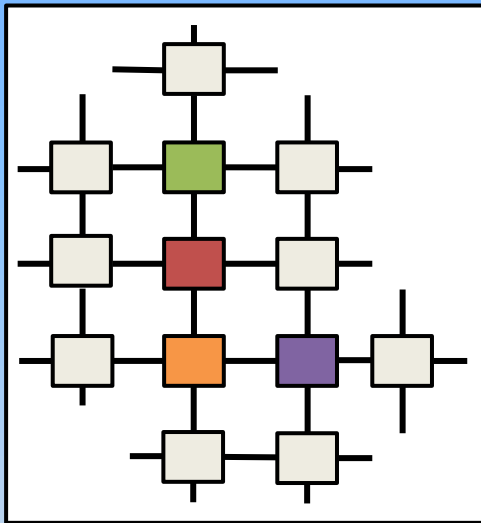
Send the result to the motion controller

else

Restart the online stage

Transformation (1 / 2)

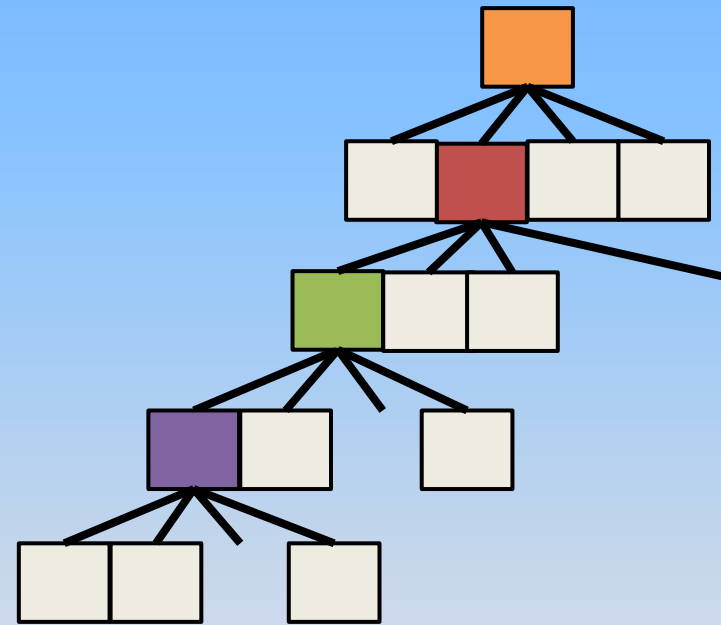
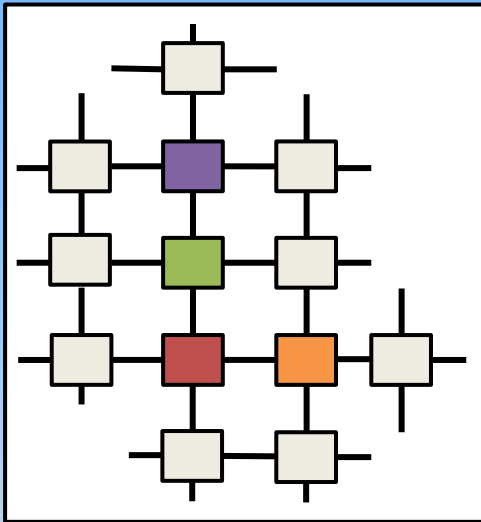
Change the root to the **north** cell



1 **1** 0 0 0 0 1 0 0 1 0 0 0 0

Transformation (2 / 2)

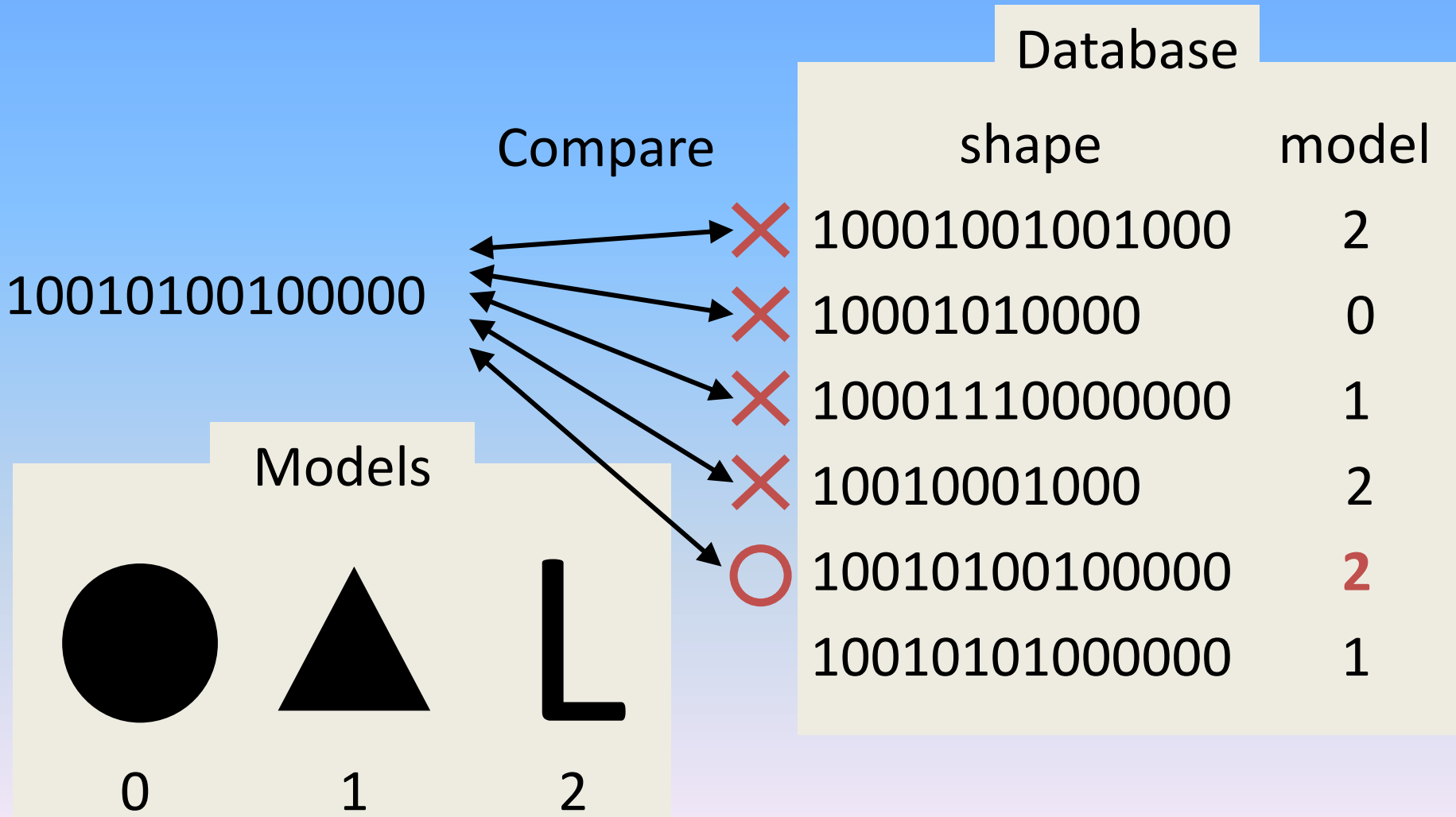
Change the root to the **west** cell



1 0 **1** 1 1 0 0 0 0 0 0 0 0 0 0



Comparison

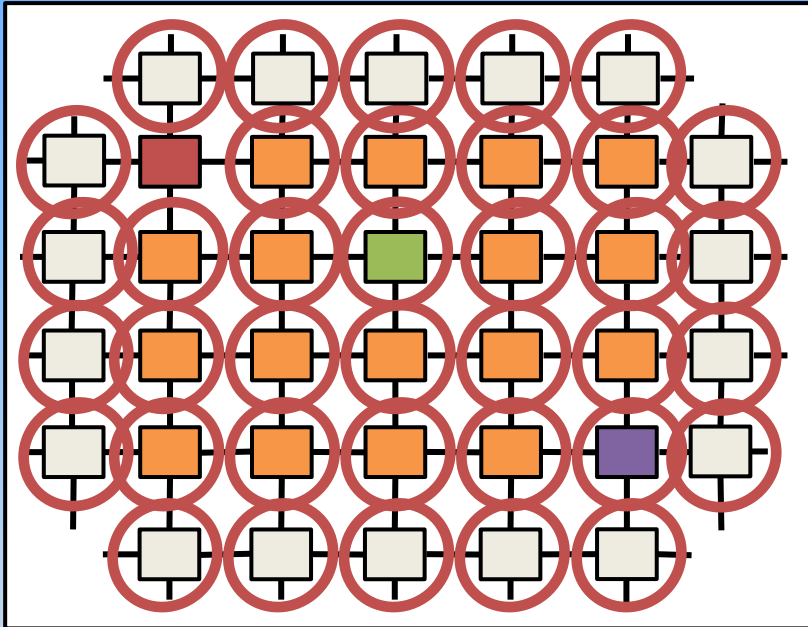


Performance analyses

- Number of communication iterations
- Communication traffic
- Computation time
- Memory footprint

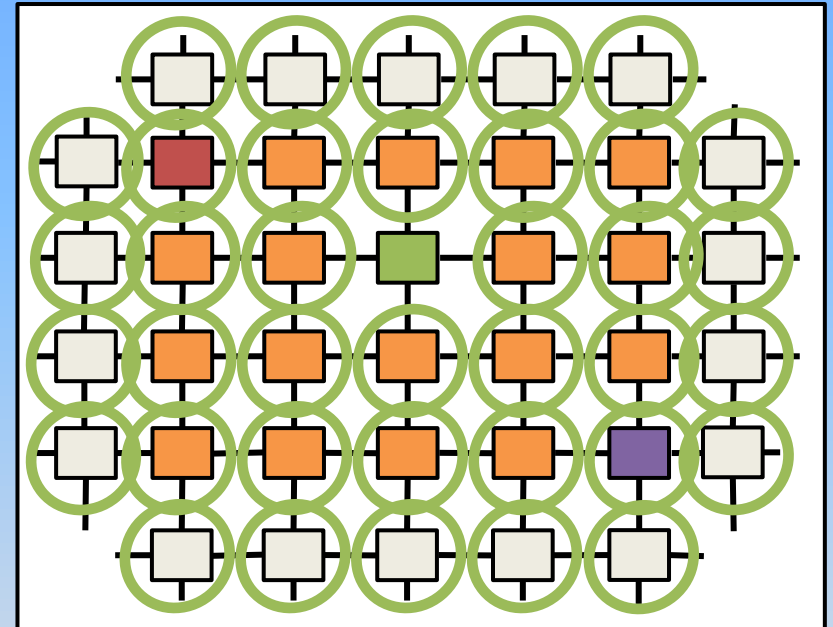
The number of communication iterations : N

Iteration 1



Edge cells need more communications

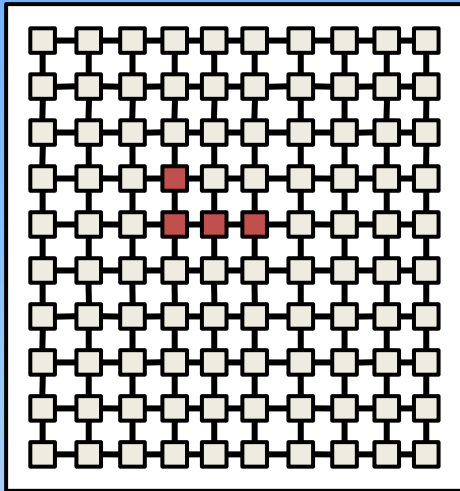
Iteration 2



Central cells need fewer communications

$$\frac{height + width}{2} + 1 \leq N \leq height + width - 1$$

Communication traffic



Smart Surface
10 x 10

Matrix

$$10 \times 10 \times 4 \times \boxed{4 \times 4} = 6400 \text{ bits}$$

↓
The number of
communication iterations

Tree-structured array

$$(1 + 4 + 7 + 10) \times \boxed{4 \times 4} = 352 \text{ bits}$$

↓
The number of
messages at a time

×

↓
The number of
active cells

× $\frac{1}{18}$

Computation time

Proposed approach

$$T = \sum_{i=1}^{N_{Reconstruction}} TR_i + \sum_{j=1}^{N_{Transform}} TT_j + \sum_{k=1}^{N_{Comparison}} TC_k$$

Previous approach

$$T' = \sum_{i=1}^{N'_{Reconstruction}} \left(TR'_i + TA_i + \sum_{j=1}^{N'_{Comparison}} TC'_{ij} \right)$$

- Reconstruction time (TR)
- Transformation time (TT)
- Comparison time (TC)
- Computation time (TA)

Memory footprint(1/2)

Previous approach

- One shape needs 29 bytes ($\cong \frac{15 \times 15}{8}$)

- $29 \times N_{Shape}$ bytes for matrices

- One criterion needs 4 bytes

- $4 \times N_{Criteria}$ bytes for criteria

- Total Memory for models is :

$$M = 29 \times N_{Shape} + 4 \times N_{Criteria}$$

Memory footprint(2/2)

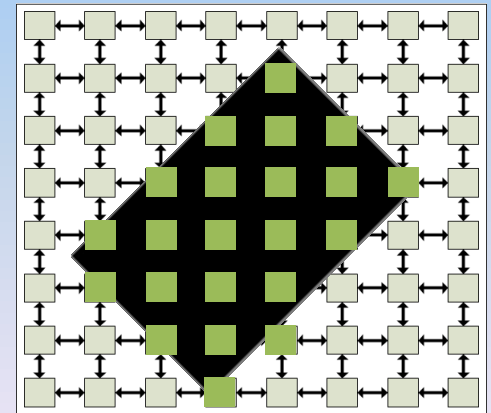
Proposed approach

- One shape needs:

$$MS_i = 3n + 2 \text{ bits}$$

- n is the number of cells covered by the object
- Total Memory for models is :

$$M = \sum_{i=1}^{N_{Shapes}} MS_i$$



n is 22, $MS_i = 68$ bits

Simulation of the offline stage(1/2)

Circle : 48 shapes

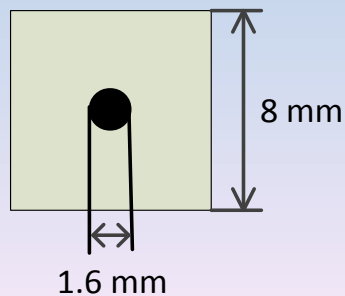
Rectangle : 248 shapes

H : 428 shapes

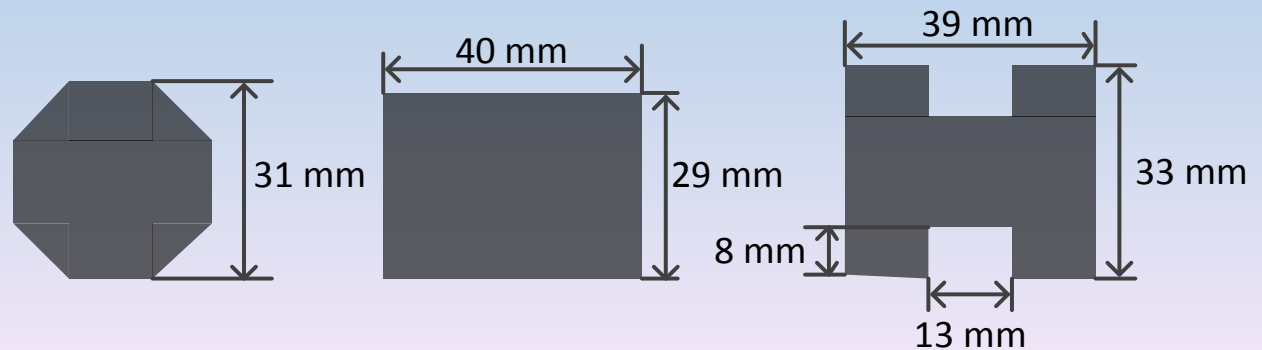
Criteria : 58

724 shapes

MEMS



Models



Every model covers less than 25 cells

Simulation of the offline stage(2/2)

Previous approach

$$\begin{aligned} M &= 29 \times N_{Shape} + 4 \times N_{Criteria} \\ &= 29 \times 724 + 4 \times 58 \\ &= \mathbf{21228} \text{ bytes} \end{aligned}$$

Proposed approach

$$\begin{aligned} M &= \sum_{i=1}^{N_{Shapes}} MS_i & MS_i &= 3 \times 25 + 2 \text{ bits} \\ & & &\cong 10 \text{ bytes} \\ &\cong \mathbf{10} \times 724 \\ &= \mathbf{7240} \text{ bytes} \end{aligned}$$

reduction of the memory footprint

724 shapes are too many to store in every cell



Include the shape
appearing rarely

Reduce the stored shapes

fewer



The number
of shapes



many

low



the probability
of matching



high

Conclusions and future work

- Representing the shapes as tree-structured array reduces their memory footprint and redundant information in messages.
- The number of shapes can be reduced, but it trades off with the probability of the successful differentiation.
- Reduction of the number of shapes to be stored in every cell.