MLRF Lecture 01

J. Chazalon, LRDE/EPITA, 2019

Introduction to *Twin it!*

Lecture 01 part 03

Twin it! overview

A poster game

- X bubbles, all different but
- Y bubbles, which have 1 (and only 1) twin

Your goals

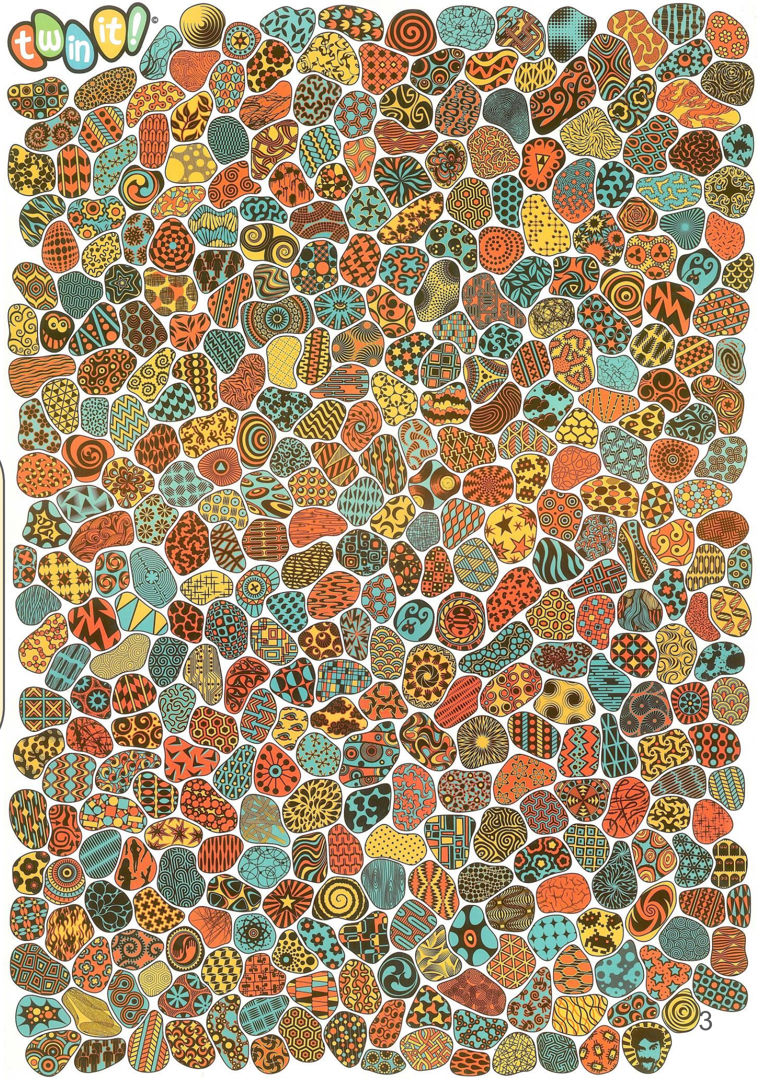
- …
- Find the pairs
- …

Discussion (3 minutes):

1. **How can we <u>decompose</u> the problem?**
2. **How can we make <u>sure</u> our solution works?**
3. **What should we <u>focus</u> on?**

Already done

- Scan the poster
- Stitch the tiles
- Normalize the contrast



Twin it! overview

A poster game

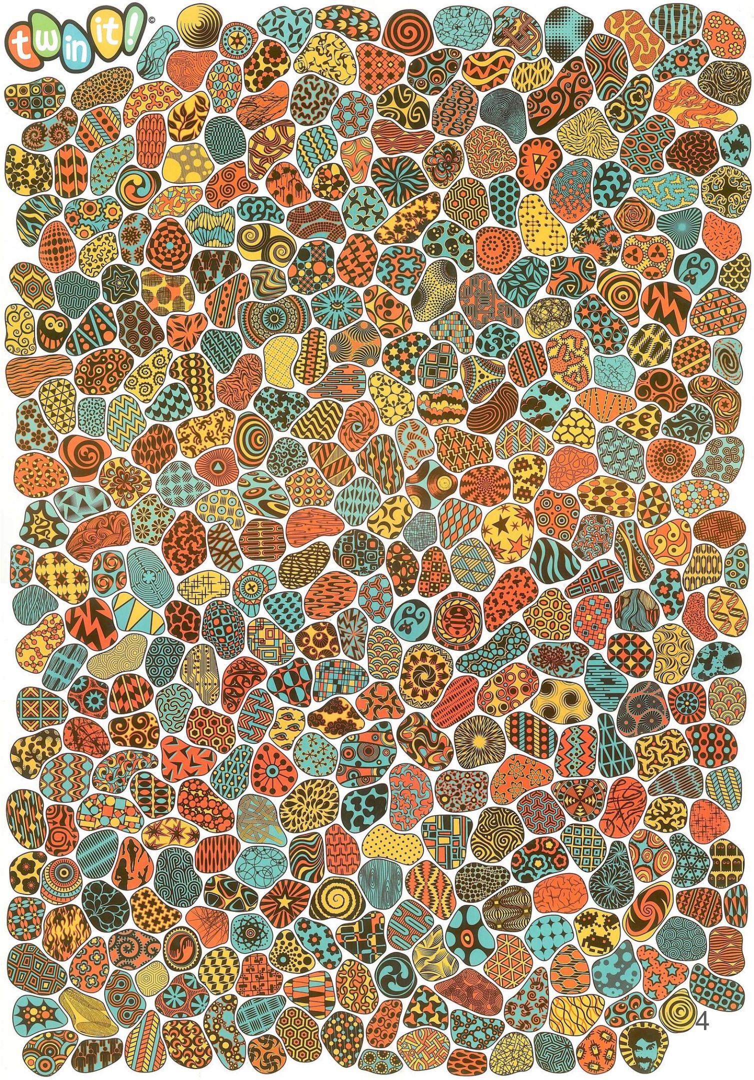
- X bubbles, all different but
- Y bubbles, which have 1 (and only 1) twin

Your goals

- **Isolate each bubble**
- Find the pairs
- **Check it works**

Already done

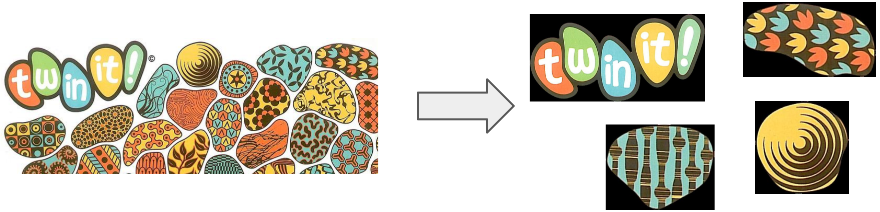
- Scan the poster
- Stitch the tiles
- Normalize the contrast



Twin it! underlying problems

1. Isolate each bubble ⇒ **Segmentation**

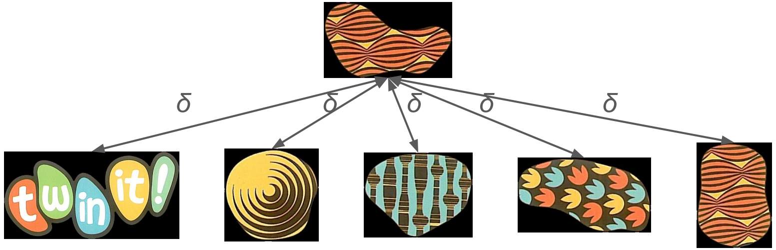
    We provide pre-computed results for this step.



2. Find the pair ⇒ **Matching**

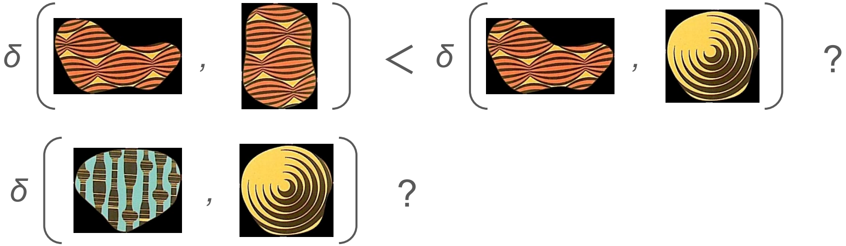
    We will focus on this one.

    *We will use **Template Matching.***



3. Check it works ⇒ **Evaluation**

    We will understand the challenges of this one.



Template Matching

Lecture 01 part 04

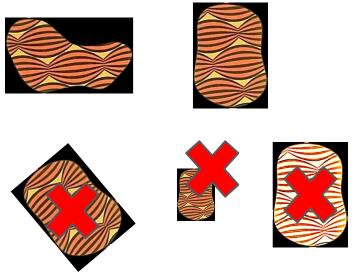
Why template matching?

A simple method which will be useful to understand

- Evaluation challenges
- The ideas behind keypoint detection (next lecture)

It can work in the Twin it! case

- Twice the same texture (in two bubbles of different shape)
- Textures at the **same scale**,
  <u>without</u> rotation
  nor **intensity change**
- Only need to cope with **translation** (and some **small noise**)



Step by step: Compare two images

Two arrays of intensities

128	128	10
126	126	9
126	126	9

I_1

135	130	12
127	128	8
126	128	9

I_2

Step by step: Compare two images

Two arrays of intensities

Take the difference

128	128	10
126	126	9
126	126	9

I_1

135	130	12
127	128	8
126	128	9

I_2

-7	-2	-2
-1	-2	1
0	-2	0

R

$$R(x,y) = I_1(x,y) - I_2(x,y)$$

Step by step: Compare two images

Two arrays of intensities

Take the **absolute** difference

128	128	10
126	126	9
126	126	9

I_1

135	130	12
127	128	8
126	128	9

I_2

7	2	2
1	2	1
0	2	0

R

$$R(x, y) = |I_1(x, y) - I_2(x, y)|$$

Step by step: Compare two images

Two arrays of intensities

Take the **squared** difference

128	128	10
126	126	9
126	126	9

I_1

135	130	12
127	128	8
126	128	9

I_2

49	4	4
1	4	1
0	4	0

$$R(x, y) = (I_1(x, y) - I_2(x, y))^2$$

R

Step by step: Compare two images

Two arrays of intensities

Take the **squared** difference

Sum the differences

128	128	10
126	126	9
126	126	9

I_1

135	130	12
127	128	8
126	128	9

I_2

49	4	4
1	4	1
0	4	0

R

$\Rightarrow$ S = 67

$$S = \sum_{x,y} (I_1(x,y) - I_2(x,y))^2$$

Step by step: Compare two images

Two arrays of intensities

Take the **squared** difference

Sum the differences

(Opt.) Normalize so the results belongs to [0, 1].

0: closest / match
1: farthest / no match

128	128	10
126	126	9
126	126	9

I_1

135	130	12
127	128	8
126	128	9

I_2

49	4	4
1	4	1
0	4	0

R

$\Rightarrow$ S = $6.8*10^{-4}$

$$S = \frac{\sum_{x,y}(I_1(x,y) - I_2(x,y))^2}{\sqrt{\sum_{x,y} I_1(x,y)^2 \cdot \sum_{x,y} I_2(x,y)^2}}$$

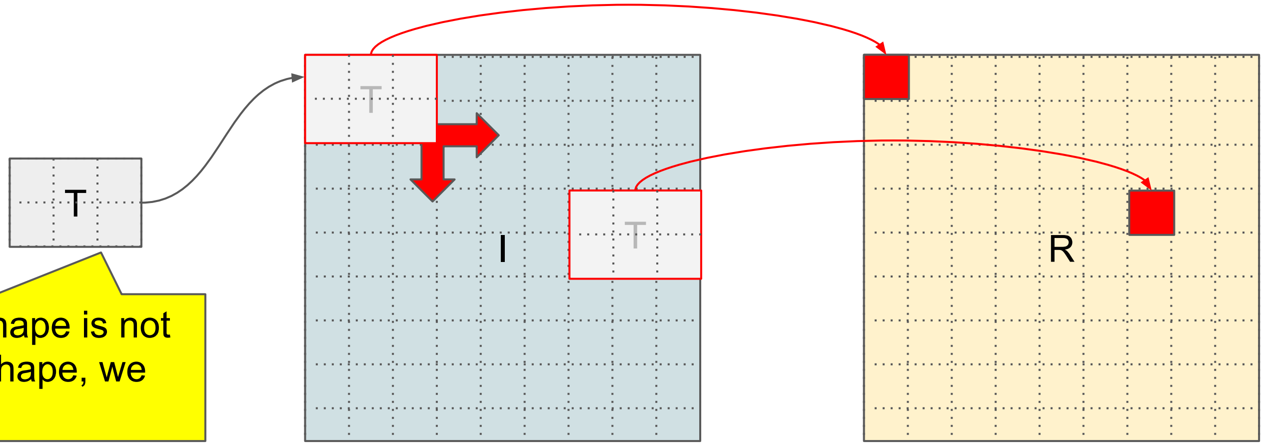
"Sum of squared differences" or "SSD"

Template Matching: Sliding comparison

I_1 is a small template T to match against I_2 (just I after).

We rewrite the preceding formula to compute a map R of the shape of I.

Each pixel of R will have the value of the SSD when the top-left pixel of T in on the pixel (x,y) of I.



Warning: If I's shape is not bigger than T's shape, we need **padding**.

Several approaches ⇒ Practice session

Sum of squared differences

$$R(r,c) = \sum_{r',c'} (T(r',c') - I(r+r',c+c')^2$$

Cross correlation

$$R(r,c) = \sum_{r',c'} (T(r',c') \cdot I(r+r',c+c'))$$

Correlation coefficient

$$R(r,c) = \sum_{r',c'} (T'(r',c') \cdot I'(r+r',c+c'))$$

where:

$$T'(r',c') = T(r',c') - 1/(w \cdot h) \cdot \textstyle\sum_{r'',c''} T(r'',c'')$$

$$I'(r+r',c+c') = I(r+r',c+c') - 1/(w \cdot h) \cdot \textstyle\sum_{r'',c''} I(r+r'',c+c'')$$

Simply divide by the mean of pixel values

Normed SSD

$$R(r,c) = \frac{\sum_{r',c'} (T(r',c') - I(r+r',c+c')^2}{\sqrt{\sum_{r',c'} (T(r',c'))^2 \cdot \sum_{r',c'} I(r+r',c+c')^2}}$$

Normed CCORR

$$R(r,c) = \frac{\sum_{r',c'} (T(r',c') \cdot I(r+r',c+c'))}{\sqrt{\sum_{r',c'} T(r',c')^2 \cdot \sum_{r',c'} I(r+r',c+c')^2}}$$

Normed CCOEFF

$$R(r,c) = \frac{\sum_{r',c'} (T'(r',c') \cdot I'(r+r',c+c'))}{\sqrt{\sum_{r',c'} T'(r',c')^2 \cdot \sum_{r',c'} I'(r+x',c+c')^2}}$$

Always the same normalization

Several approaches ⇒ Practice session

Sum of squared differences

$$R(r,c) = \sum_{r',c'} (T(r',c') - I(r+r', c$$

Normed SSD

$$, c) = \frac{\sum_{r',c'} (T(r',c') - I(r+r',c+c')^2}{\sqrt{\sum_{r',c'} (T(r',c'))^2 \cdot \sum_{r',c'} I(r+r',c+c')^2}}$$

Both very similar: just a local normalization

Cross correlation

$$R(r,c) = \sum_{r',c'} (T(r',c') \cdot I(r+r', c+c'))$$

Normed CCORR

$$R(r,c) = \frac{\sum_{r',c'} (T(r',c') \cdot I(r+r',c+c'))}{\sqrt{\sum_{r',c'} T(r',c')^2 \cdot \sum_{r',c'} I(r+r',c+c')^2}}$$

Correlation coefficient

$$R(r,c) = \sum_{r',c'} (T'(r',c') \cdot I'(r+r', c+c'))$$

Normed CCOEFF

$$R(r,c) = \frac{\sum_{r',c'} (T'(r',c') \cdot I'(r+r',c+c'))}{\sqrt{\sum_{r',c'} T'(r',c')^2 \cdot \sum_{r',c'} I'(r+x',c+c')^2}}$$

where:

$$T'(r',c') = T(r',c') - 1/(w \cdot h) \cdot \sum_{r'',c''} T(r'',c'')$$

$$I'(r+r',c+c') = I(r+r',c+c') - 1/(w \cdot h) \cdot \sum_{r'',c''} I(r+r'',c+c'')$$

Simply divide by the mean of pixel values

Always the same normalization

Several approaches ⇒ Practice session

Sum of squared differences

The smaller (close to 0),
the more similar

$$R(r,c) = \sum_{r',c'} (T(r',c') - I(r+r',c+c')^2$$

rmed SSD

$$R(r,c) = \frac{\sum_{r',c'} (T(r',c') - I(r+r',c+c')^2}{\sqrt{\sum_{r',c'} (T(r',c'))^2 \cdot \sum_{r',c'} I(r+r',c+c')^2}}$$

Cross correlation

The larger,
the more similar

$$R(r,c) = \sum_{r',c'} (T(r',c') \cdot I(r+r',c+c'))$$

rmed CCORR

$$R(r,c) = \frac{\sum_{r',c'} (T(r',c') \cdot I(r+r',c+c'))}{\sqrt{\sum_{r',c'} T(r',c')^2 \cdot \sum_{r',c'} I(r+r',c+c')^2}}$$

Correlation coefficient

The larger,
the more similar

$$R(r,c) = \sum_{r',c'} (T'(r',c') \cdot I'(r+r',c+c'))$$

rmed CCOEFF

$$R(r,c) = \frac{\sum_{r',c'} (T'(r',c') \cdot I'(r+r',c+c'))}{\sqrt{\sum_{r',c'} T'(r',c')^2 \cdot \sum_{r',c'} I'(r+x',c+c')^2}}$$

where:

$$T'(r',c') = T(r',c') - 1/(w \cdot h) \cdot \textstyle\sum_{r'',c''} T(r'',c'')$$

$$I'(r+r',c+c') = I(r+r',c+c') - 1/(w \cdot h) \cdot \textstyle\sum_{r'',c''} I(r+r'',c+c'')$$

Simply divide by the
mean of pixel values

Always the same
normalization

Cross correlation: 2 things to know

$$R(r,c) = \sum_{r',c'} (T(r',c') \cdot I(r+r',c+c'))$$

More robust to intensity shifts (as long as gradients “agree”) than SSD

SSD: X+offset - X = offset   CCORR: (X+offset) · X ≅ X^2

Base version requires to **normalize T by its mean**

Otherwise large image values always produce better matches

Not necessary for CCOEFF