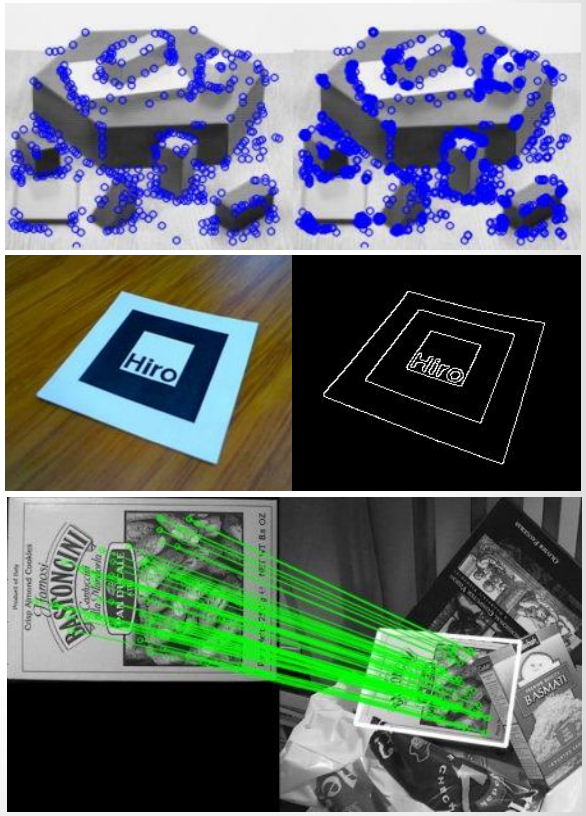
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Image Processing
Object Detection

Introduction

- **Feature detection and matching are an essential component of many CV applications**

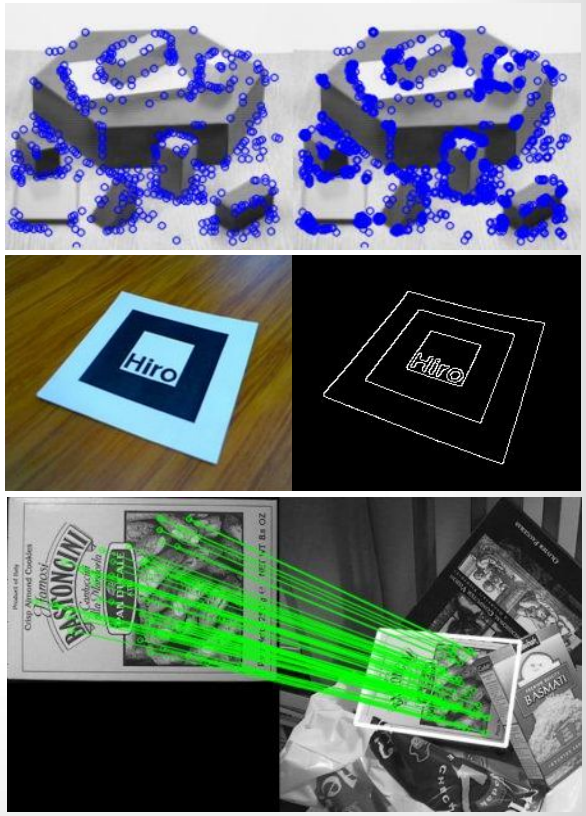


- **Important features on images**
  - **Key points/corners**
  - **Edges/Contours**
  - **Patterns/Templates**
  - **Whole objects**

Introduction

- So far, we have seen how to
  - Remove noise
  - Compute gradients
  - Detect Edges

- The last thing that will cover in this course is the simplest method to detect objects in images
  - Template Matching



Template Matching

- **Template matching is a method to find the location of a target object in a larger image**

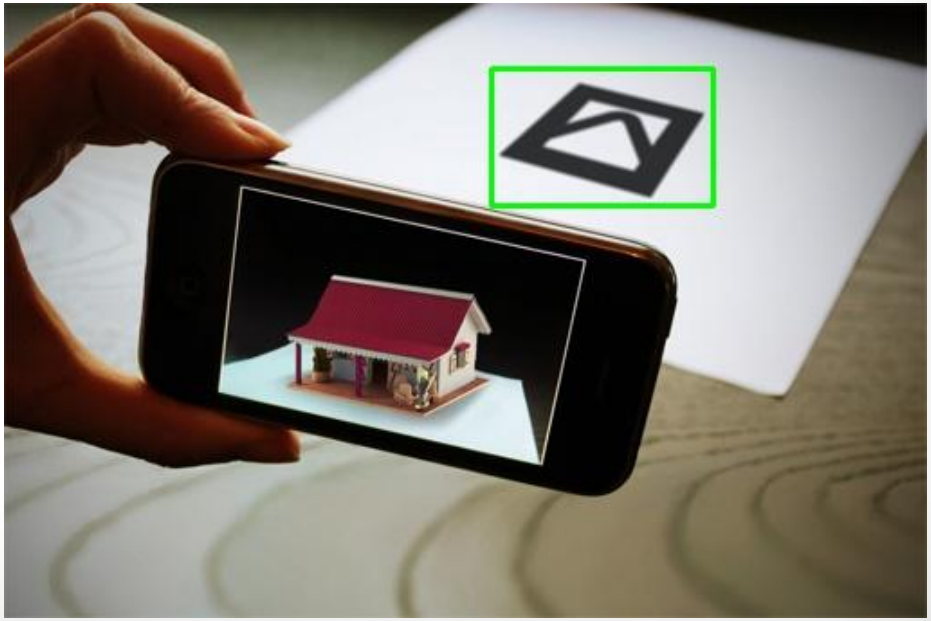
Target

Input image

Template Matching

- **Template matching is a method to find the location of a target object in a larger image**

Target



Template Matching Result

Template Matching

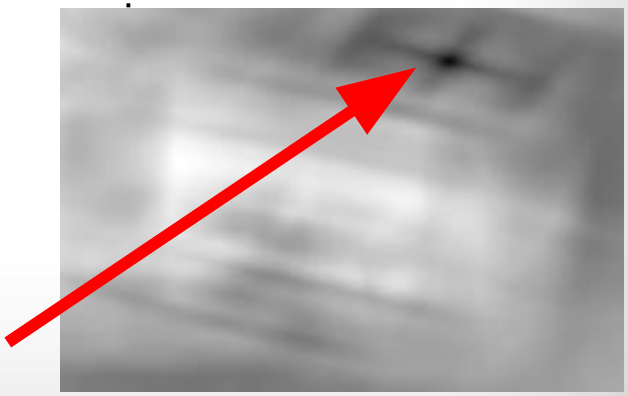
- **Based on sliding the target patch over the input image (convolution kernel)**

Target and input image

Template Matching

- Based on sliding the target patch over the input image (convolution kernel)
- At each pixel of the input image, a metric tells how good or bad the match at the pixel is
- This generates a "Matching map"
  - Minimum or maximum value (depending on the metric) represents the possible location of the target in the image

Target and input



Possible match
(min value)

Match map

Template Matching

- **Metrics used to compute the matching map**
  - Being I the Input image, T the template patch and R the matching map (result)
  - If I resolution is (W, H) and T is (w, h), R resolution is $(W - w + 1, H - h + 1)$

- **Squared Difference**

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

- **Normalized Squared Difference**

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

Possible match is the **minimum** value in both metrics

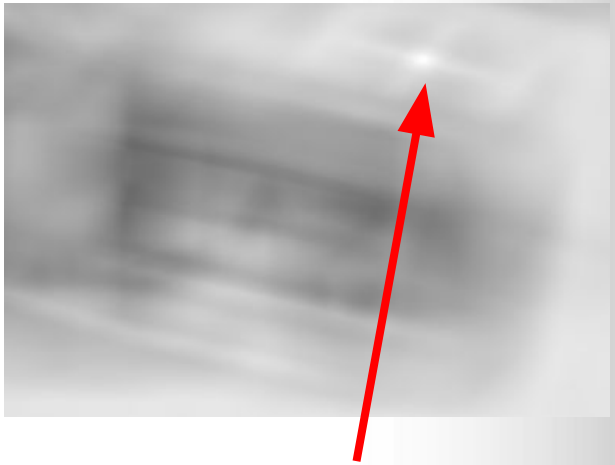
Template Matching

- **Cross Correlation**

$$R(x,y) = \sum_{x',y'} (T(x',y') \cdot I(x+x', y+y'))$$

- **Normalized Cross Correlation**

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



Possible match is the **maximum** value in both metrics

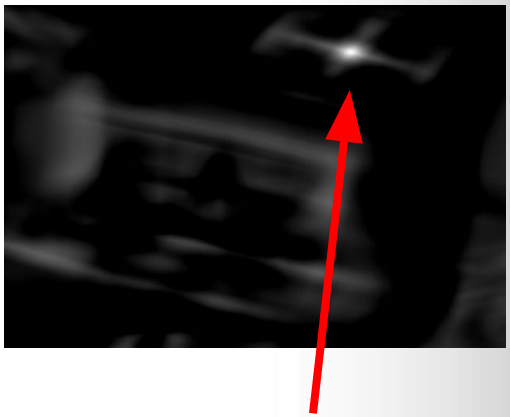
Template Matching

- **Correlation Coefficient**

$$R(x,y) = \sum_{x',y'} (T'(x',y') \cdot I'(x+x',y+y'))$$

where

$$T'(x',y') = T(x',y') - 1/(w \cdot h) \cdot \sum_{x'',y''} T(x'',y'')$$
$$I'(x+x',y+y') = I(x+x',y+y') - 1/(w \cdot h) \cdot \sum_{x'',y''} I(x+x'',y+y'')$$



Possible match is the **maximum** value in both metrics

- **Normalized Correlation Coefficient**

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

Template Matching

- **Template matching with OpenCV**

cv2.matchTemplate(img, targ, metric) → matchMap		
img	Input image	
targ	Target patch	
metric	cv2.TM_SQDIFF cv2.TM_SQDIFF_NORMED cv2.TM_CCORR cv2.TM_CCORR_NORMED cv2.TM_CCOEFF cv2.TM_CCOEFF_NORMED	0 1 2 3 4 5
matchMap	Returns the matching map computed with the given metric	


Template Matching

- **Results**

Target

Template Matching Result

Template Matching

- **Advantages**
  - Easy to compute (slide the target over the image)
- **Disadvantages**
  - The target must appear in the image as is
    - No rotations
    - No scaling
  - Depending on image and target resolution may be slow

Further Reading

- OpenCV offers a lot of well-know CV techniques that can be used in out AR applications
  - Key points and corner detection
  - Pattern recognition/Feature matching well-suited to rotations, scaling…
  - Segmentation of objects
  - Object detection using machine learning (faces, etc.)
  - …
- More information and examples related to these techniques and can be found in
  - http://opencv.org/
  - http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html
- An interesting review on Computer Vision applications can be found in the book

  S. Szeliski. Computer Vision: Algorithms and Applications. *Springer Science & Business Media*. 2010

Questions?

www.citm.upc.edu