ECE5554 – Computer Vision
Lecture 3b – Area-based Matching

Creed Jones, PhD

Today’s Objectives

- Area-based matching
  - Example application: vision-guided aircraft landing
  - Correlation coefficient
- Normalized grayscale correlation
- Variations
  - Pyramid-based matching
  - Vector matching
- Applications
  - Guidance
  - Optical Character Recognition

Template matching on edge images

Binary image template matching

- Overlap counting
- The distance transform
- Chamfer matching

Recognizing objects: what and where they are, along with some quality information

- Very often we want to search an image for an occurrence of one or more known objects
- There are several bits of information we want:
  - Is the object present anywhere in the image?
  - Where in the image is it located?
  - How much does it look like what we expected?
- This is called object detection

Template matching can be used to find specific patterns (seen as subimages)

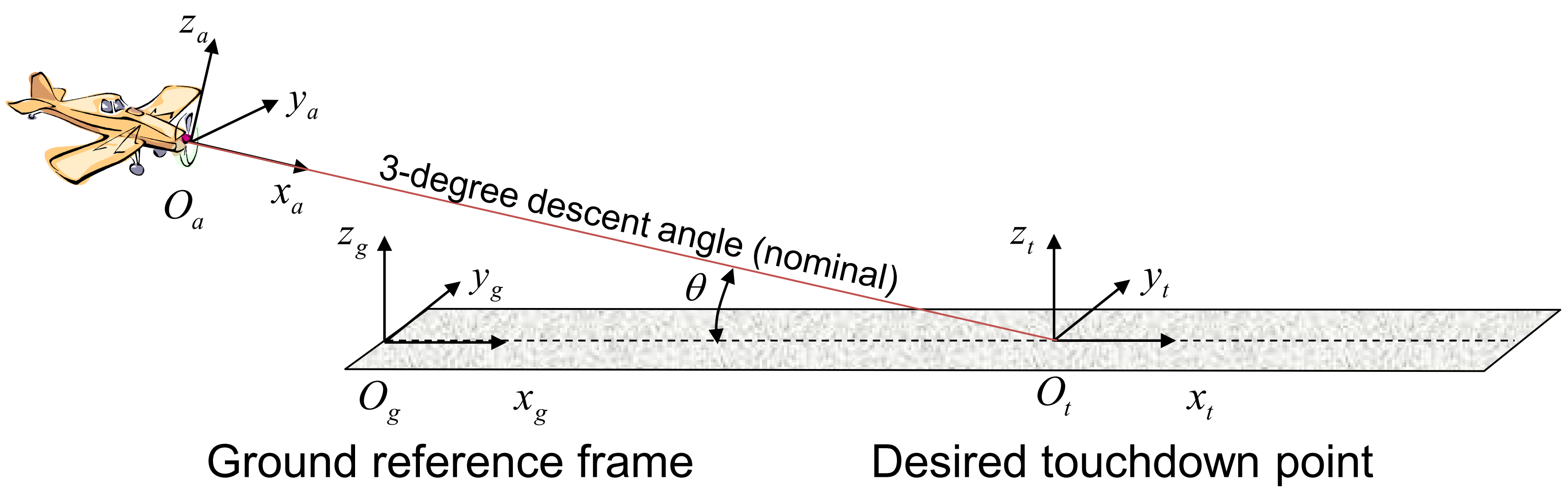
- One method for finding a given object in a scene is called *template matching*
- A *template* is a gray-scale image of the object we are looking for
  - Algorithm finds the part of the image that looks most like the template
- This is nice since the object description is just an image
  - "Teach by example"

Template matching is also called *area-based matching*

- Image-to-image matching is used in many vision-related problems
- Area-based matching refers to the comparison of pixel values from localized parts of images
  - Usually implies a fixed-size window

- The characteristics that we are looking for is a region of image where the pixel values are similar to those that we wish to find

An application: Aircraft landing



(Research supported by NASA Langley Research Center)

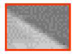
Test images were captured from an OV-10A

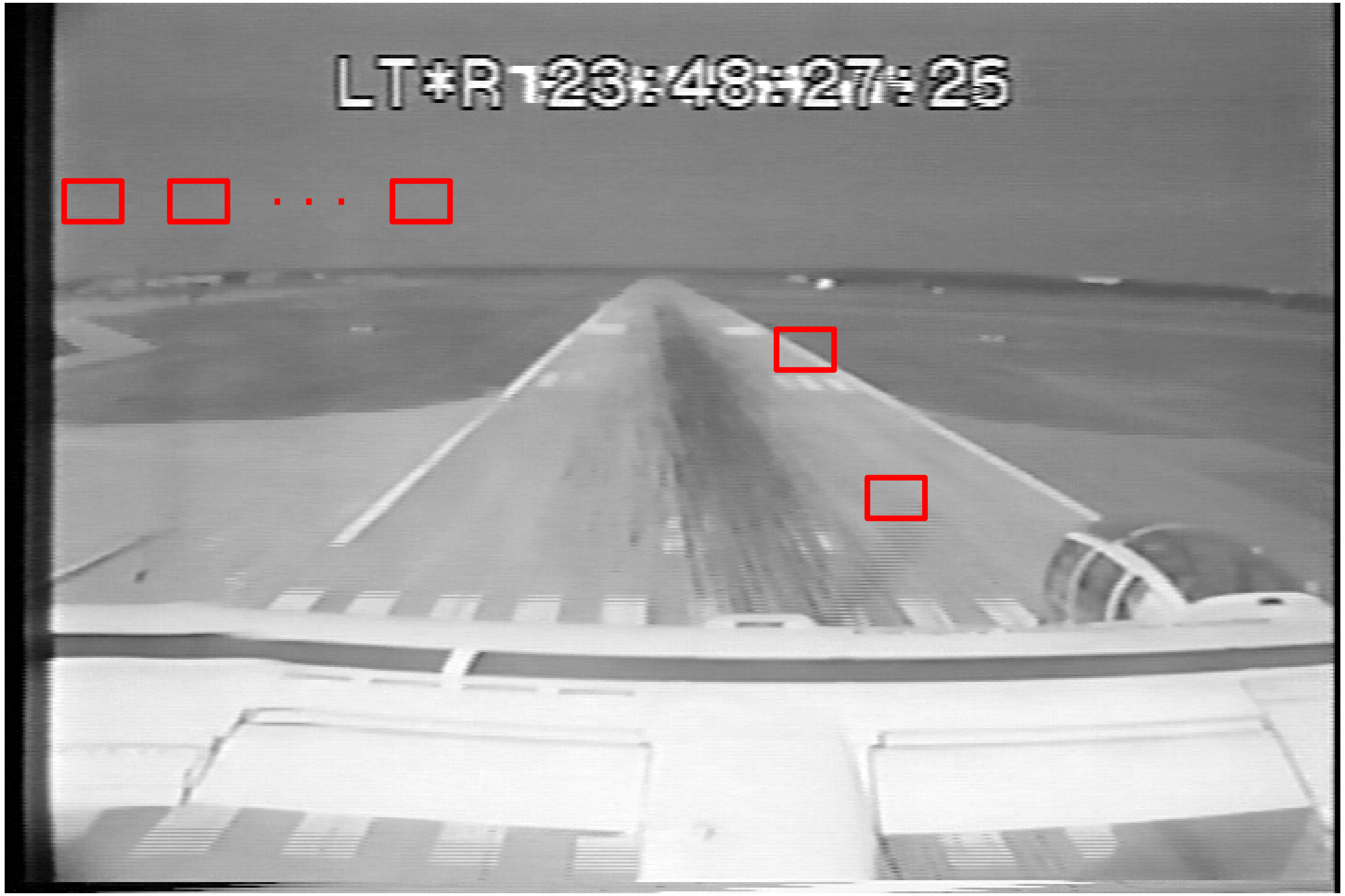
Example image

Suppose we want to automate the problem of landing an airplane

- Let’s consider an area-based method
- Create an image template T that resembles a part of the runway
- Repeat:
  - Capture an image from the airplane-mounted camera
  - In the image, find the location (r, c) that represents the closest match to template T
  - Based on (r, c), send signals to the airplane to control its speed and direction

Area-based search: the high-level idea

Find or synthesize a template for an "object of interest", such as the side of a runway → 



Perform a search in the image for the best match

Template matching procedure

- Obtain a small image (a "template") that contains only the object we want to find
- Pass a template-sized window over the original image, and perform pixel-by-pixel comparisons at each window location
- Report locations in the original image that are good matches to the template
- But what is a "good match"?

Possible match criterion #1

- Represent the image as $I(r,c)$
- Represent the template as $T(r,c)$
- Report a match at location (r,c) whenever

$$T(i,j) = I(r+i, c+j), \ \forall (i,j) \in T$$

- Usually not effective – why?

Possible match criterion #2

- For all locations (r, c), such that the template is contained in I, compute

$$E(r,c) = \sum_i \sum_j |I(r+i, c+j) - T(i,j)|$$

- Report a possible match at location (r, c) whenever $E(r,c) < \infty$
- This is known as the “sum-of-absolute-differences” (SAD) criterion

Possible match criterion #3

- For all locations (r, c), such that the template is contained in I, compute

$$E(r,c) = \sum_i \sum_j \big(I(r+i, c+j) - T(i,j)\big)^2$$

- Report a possible match at location (r, c) whenever $E(r,c) < \infty$
- This is known as the "sum-of-squared-differences" (SSD) criterion

$$E_{SSD}(r,c) = \sum_i \sum_j [I(i+r, j+c) - T(i,j)]^2$$

But notice . . .

$$[I - T]^2 = \quad I^2 - 2IT + T^2$$

$$[I - T]^2 = \quad I^2 + T^2 - 2IT$$

The image and kernel sums can be brought out…

$$[I - T]^2 = \quad C - 2IT + T^2$$

We minimize the SSD by maximizing the cross-correlation

$$CCorr(r,c) = \sum_i \sum_j I(i+r, j+c)T(i,j)$$

Possible match criterion #4

- For all locations (r, c), such that the template is contained in I, compute

$$E(r,c) = \sum_i \sum_j I(r+i, c+j)T(i,j)$$

- Report a possible match at location (r, c) whenever E(r, c) is large (local maximum)
- This is the familiar cross-correlation of I and T

Possible match criterion #5

- Normalization will help guard against false matches caused by very bright regions:

- **Normalized cross-correlation** (NCC) is popular

$$E_{NCC}(r,c) = \frac{\left(\sum_i \sum_j I(i+r, j+c)\,T(i,j)\right)}{\sqrt{\left(\sum_i \sum_j I^2(i+r, j+c)\right)}\sqrt{\left(\sum_i \sum_j T^2(i,j)\right)}}$$

- **Cross-correlation**

$$E(r,c)=\sum_i\sum_j I(i+r,j+c)\,T(i,j)$$

- **Cross-covariance**

$$E(r,c)=\sum_i\sum_j\left(I(i+r,j+c)-\mu_I\right)\left(T(i,j)-\mu_T\right)$$

(average intensities)

- **Normalized cross-covariance**

$$E(r,c)=\frac{\sum_i\sum_j\left(I(i+r,j+c)-\mu_I\right)\left(T(i,j)-\mu_T\right)}{\sqrt{\sum_i\sum_j\left(I(i+r,j+c)-\mu_I\right)^2}\sqrt{\sum_i\sum_j\left(T(i,j)-\mu_T\right)^2}}$$

Cross-correlation is a measure of similarity between two functions or sets of data

- *Cross-correlation* refers to the computation of the *inner product* of two functions
  - At the heart of it, we are multiplying two function or sets of data, point by point
  - To get the math to work right, we subtract means and normalize
- The result is a single number – the *correlation coefficient*
  - If two sets of data are identical, result =1
  - If one is the inverse of the other, result = -1

Correlation can be calculated as follows…

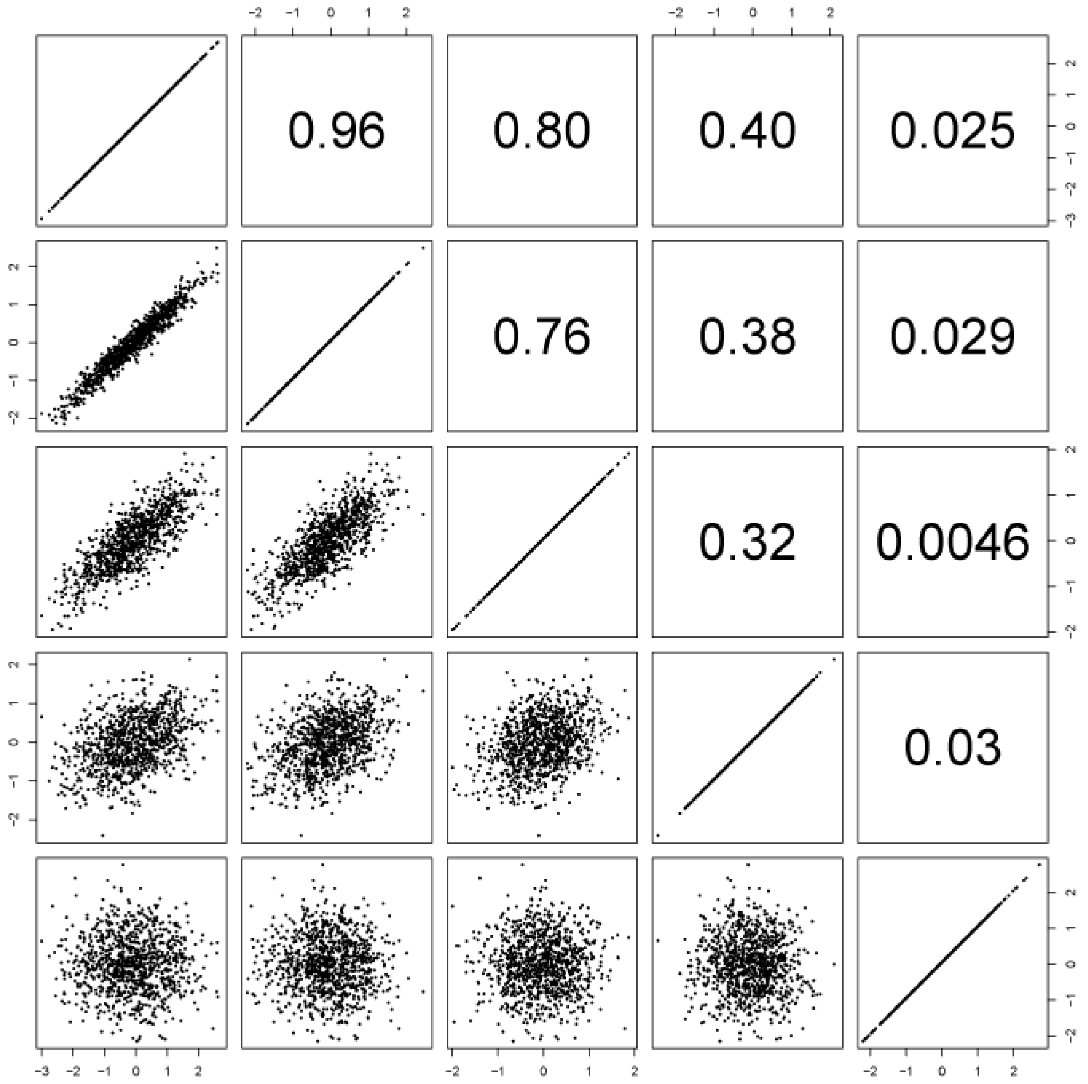
- Correlation between two sets of data is defined as:

$$\rho_{AB} = \frac{cov(A,B)}{\sigma_A \sigma_B} = \frac{E[(a-\mu_A)(b-\mu_B)]}{\sigma_A \sigma_B} = \frac{\sum_{i=0}^{N_A-1}(a_i-\mu_A)\sum_{i=0}^{N_B-1}(b_i-\mu_B)}{N_A N_B \sigma_A \sigma_B}$$

  - ρ = 1 : identical data (A = B)
  - ρ = 0.9 : nearly identical data
  - ρ = 0.5 : data are somewhat alike
  - ρ = 0 : no relationship (uncorrelated)
  - ρ = -0.9 : data are nearly opposite
  - ρ = -1 : data are inversions (A = -B)

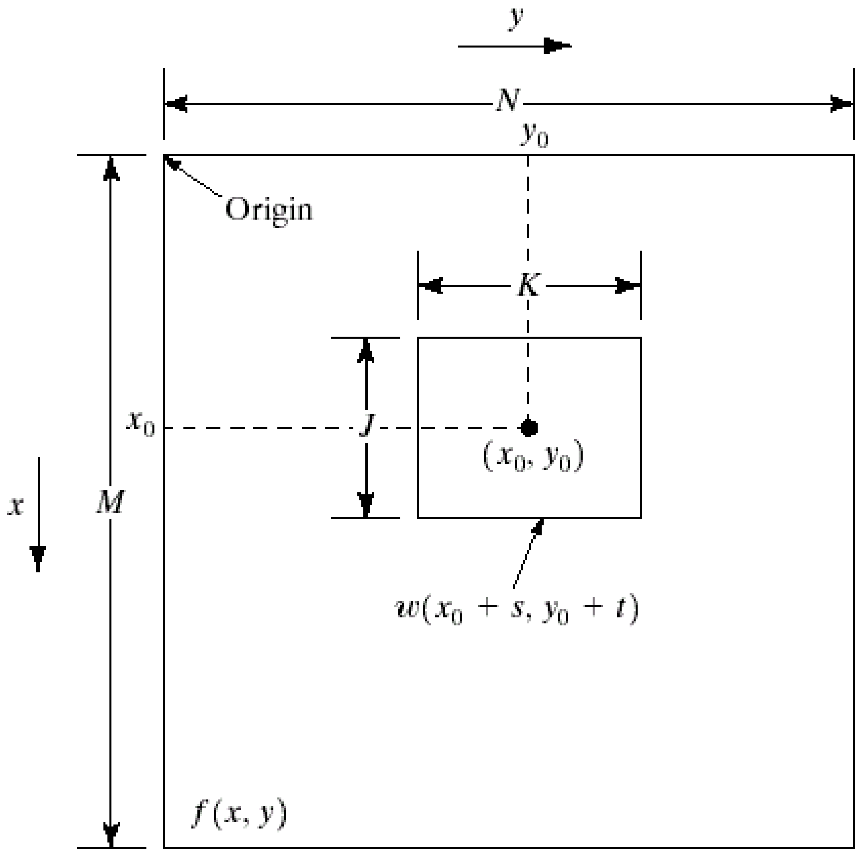
Data with various correlations

- Each cell plots a pair of data sets with a given correlation
- Cells on the axis have ρ=1
- Lower ρ indicates that the data is less similar



http://en.wikipedia.org/wiki/Correlation

Image Correlation extends the correlation coefficient to two dimensions



- The template *w* is overlaid at every candidate position *x,y* in the image
- Compute the correlation coefficient of *w* and the portion of the image aligned with *w*
- The best match is the point at which the correlation coefficient is highest

Normalized Grayscale Correlation (NGC) is an enhanced calculation that can be compared for different parts of the image

- For each point in the image, we can compute the coefficient of correlation of the template as:

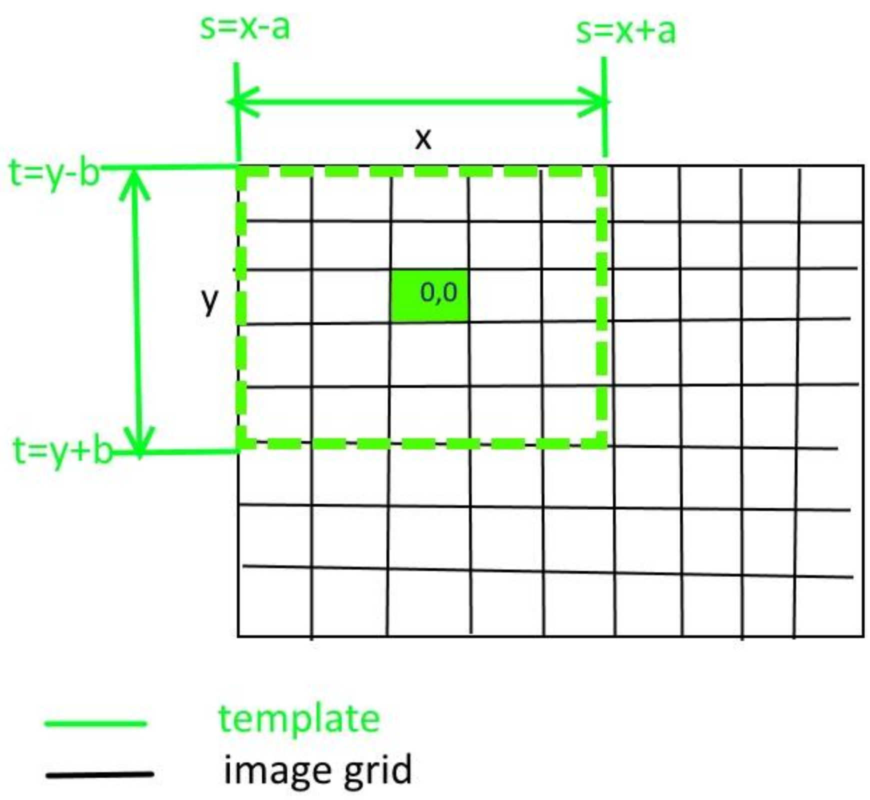
$$R(x,y) = \frac{\sum_i \sum_j (I(x+i, y+j) - \mu_I)(T(i,j) - \mu_T)}{\sqrt{\sum_i \sum_j (I(x+i, y+j) - \mu_I)^2 \sum_i \sum_j (T(i,j) - \mu_T)^2}}$$

- This is a function of x and y; each point in the image will have a value
- The point with the highest correlation coefficient is the best match to the template

C++ source code for computing the normalized grayscale correlation

```
for (y = ymarg; y < (I.dimy() - ymarg); y++)
    for (x = xmarg; x < (I.dimx() - xmarg); x++)
    {
          Imean = 0 = num = den1 = den2 = 0;
          for (j = 0; j < T.dimy(); j++)
              for (i = 0; i < T.dimx(); i++)
                   Imean += I(x+i, y+j);
          Imean /= Tsize;
          for (j = 0; j < T.dimy(); j++)
          {
              for (i = 0; i < T.dimx(); i++)
              {
                   num += (I(x+i, y+j) - Imean)*(T(i,j) - Tmean);
                   den1 += (I(x+i, y+j) - Imean)*(I(x+i, y+j) - Imean);
                   den2 += (T(i,j) - Tmean)*(T(i,j) - Tmean);
              }
          }
          R = num / sqrt(den1*den2);
          if (R > max.score)
          {
              max.score = R;
              max.x = x;    max.y = y;
          }
    }
```

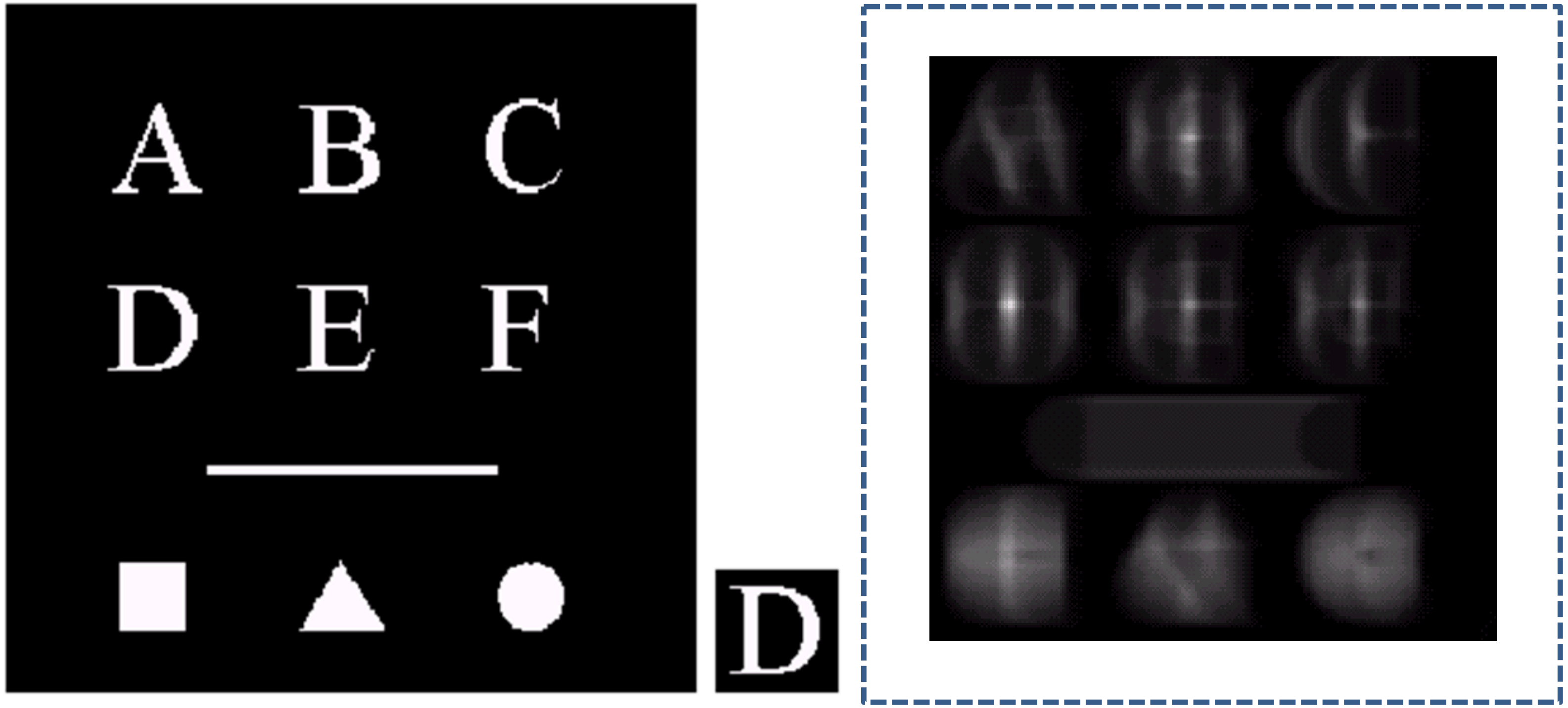
Correlation cannot be naively done out to the edges of the image (same problem occurred when applying a kernel)



- Generally, we only apply the template for the portion of the image where it fits
- If we want to find matches to partial templates, we can modify the equation for the NGC appropriately

Result of image correlation is an intensity is proportional to degree of match

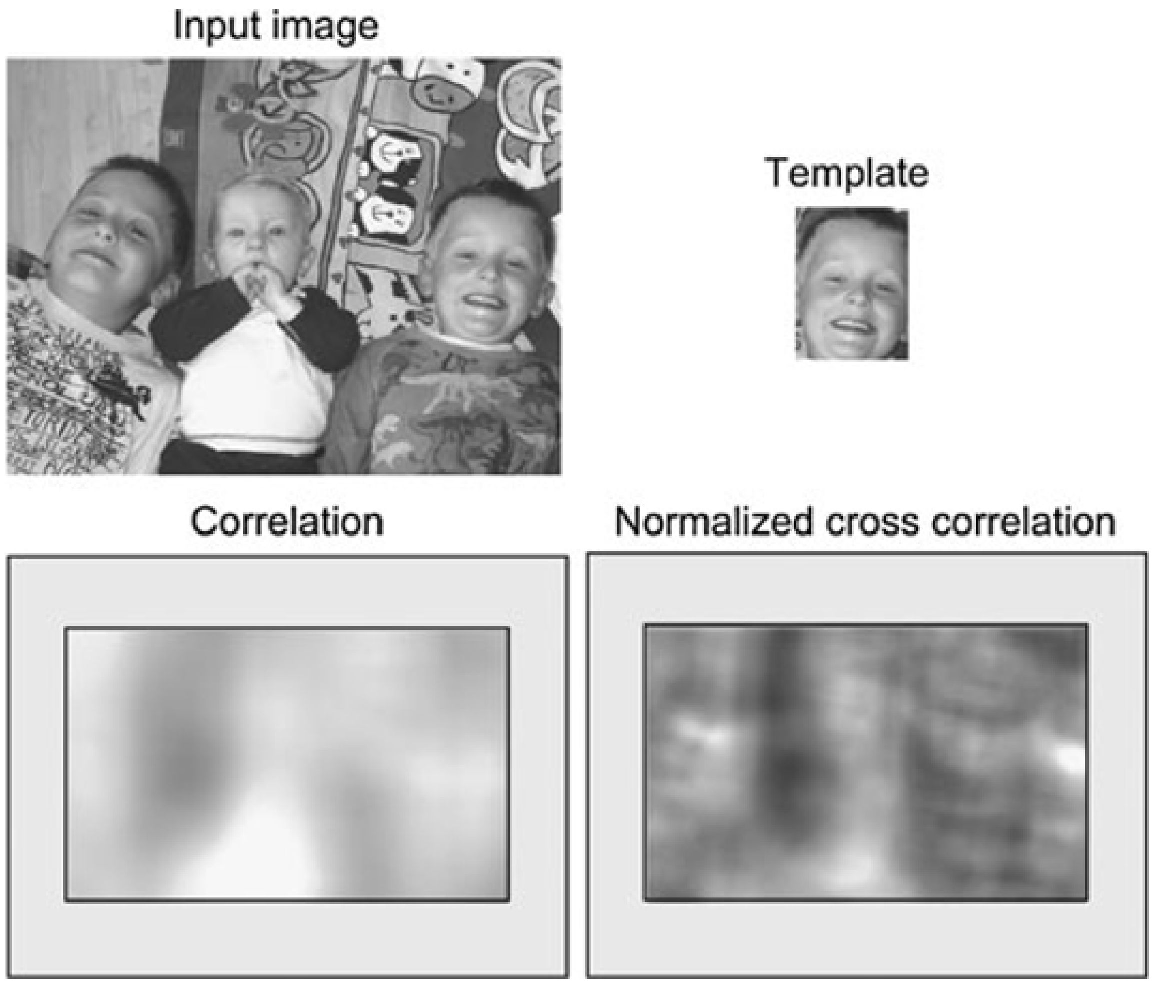
- Bright spots indicate high correlation
- Areas of high similarity
- Highest correlation ρ is on the “D”, as expected



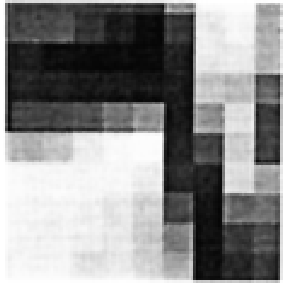
a b c

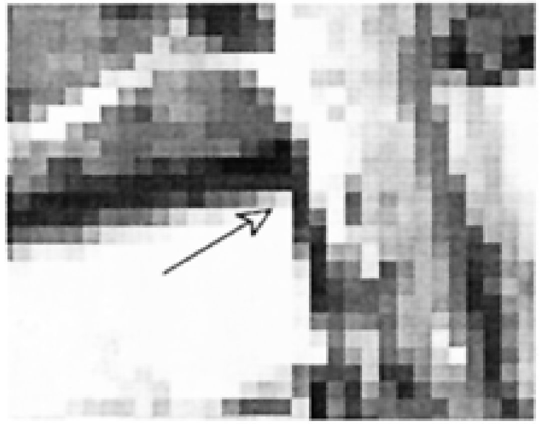
FIGURE 12.9
(a) Image.
(b) Subimage.
(c) Correlation coefficient of (a) and (b). Note that the highest (brighter) point in (c) occurs when subimage (b) is coincident with the letter “D” in (a).

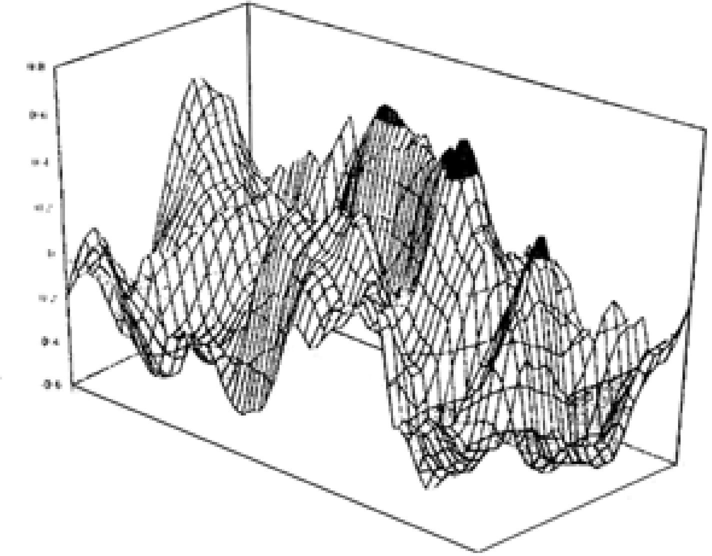
Without normalizing the correlation (dividing by average intensities), the algorithm will tend towards the brightest parts of the image



If the template is not present, the correlation coefficient will not reach significant values





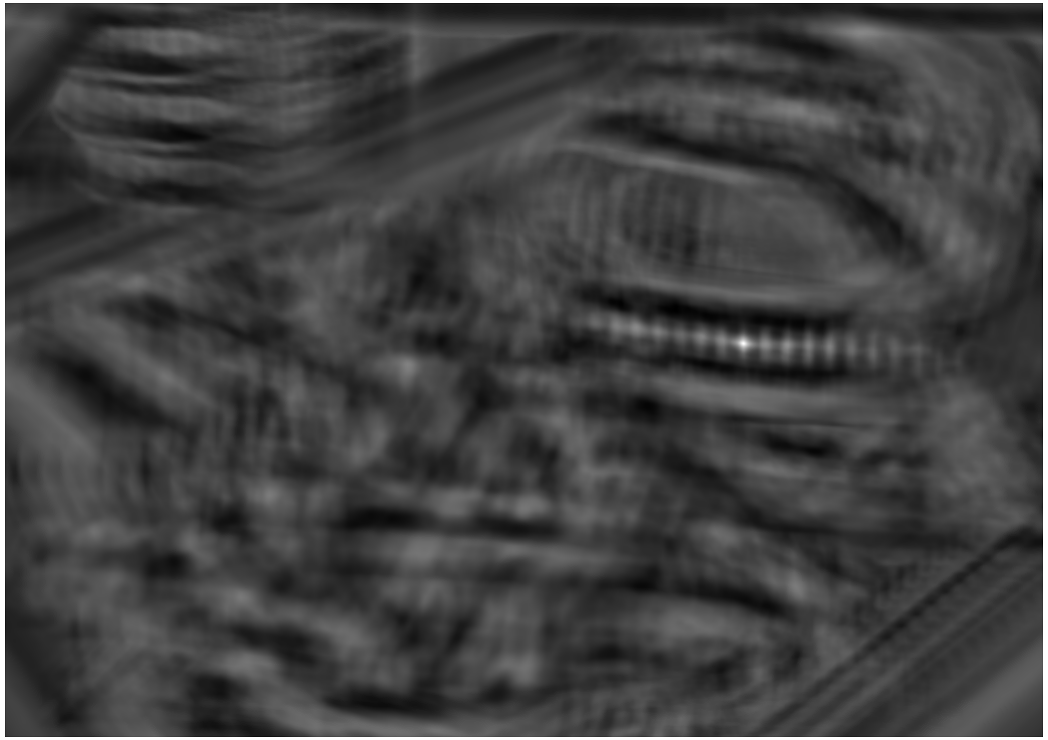


- The surface at the bottom is a plot of ρ as a function of X and Y in the image
- If ρ is too low everywhere, we conclude that the template pattern does <u>not</u> appear in the image
- Note that the template is a pretty primitive pattern

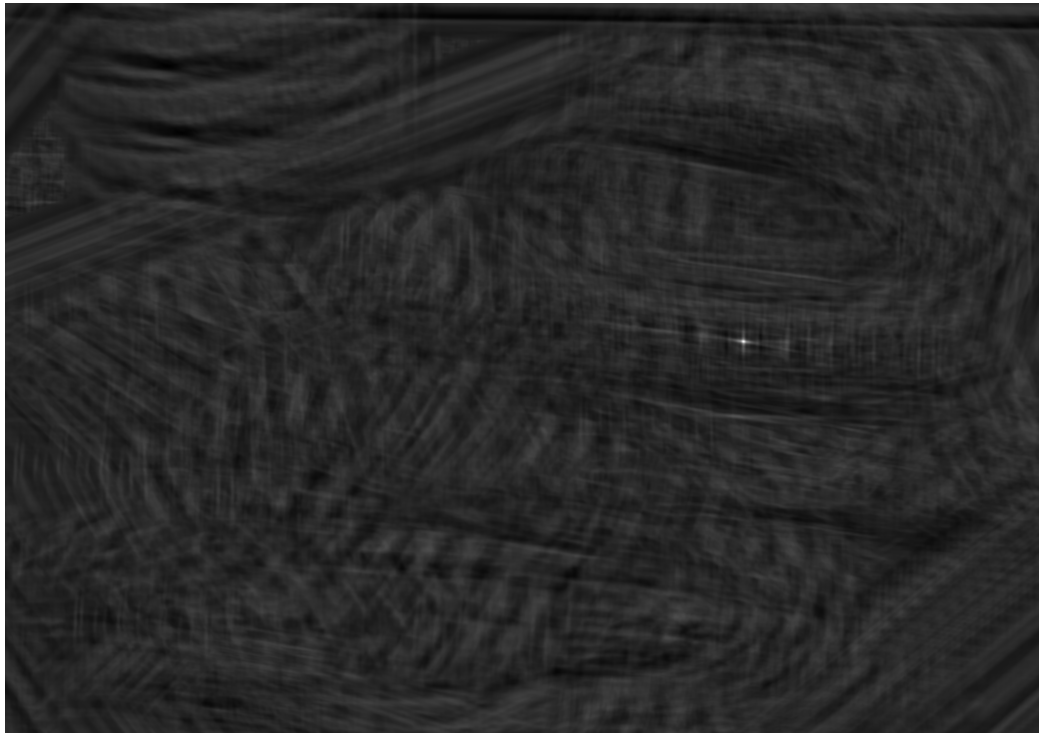
Answers to our object recognition questions: Is it there, where is it and how much does it look like the template?

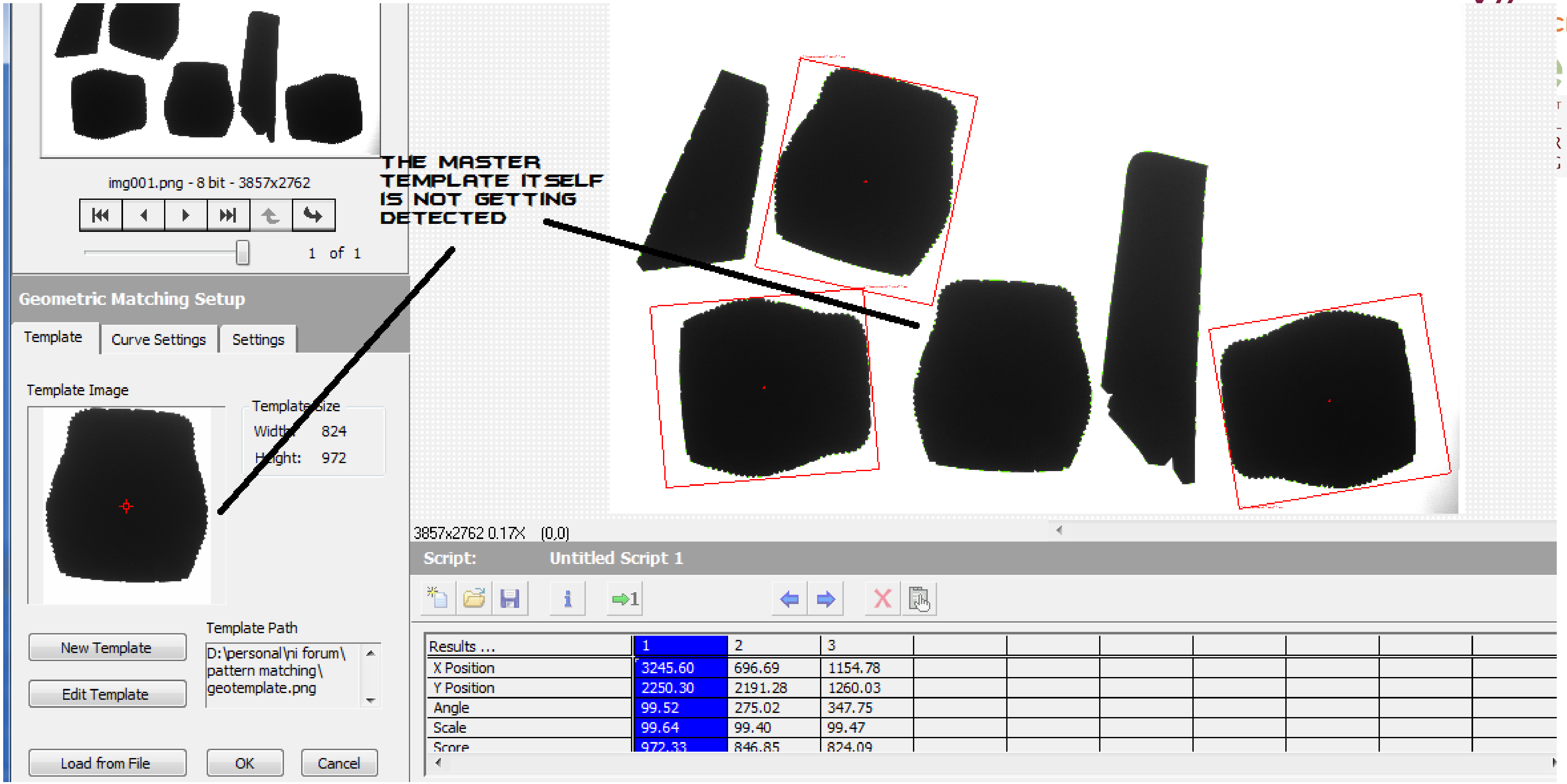
- Is the object in the image?
  - yes, if the maximum ρ is greater than some threshold (how to set the threshold?)
  - Usually, keep the top *n* ρ values and locations in a sorted *candidate list*
- Where is the object?
  - The location of the maximum ρ
- How much does it look like the template?
  - ρ – if $\rho = 1$, exactly like i

Edge images like this are very useful; for example, template matching is often carried out on edge images rather than intensity images



Edge images like this are very useful; for example, template matching is often carried out on edge images rather than intensity images



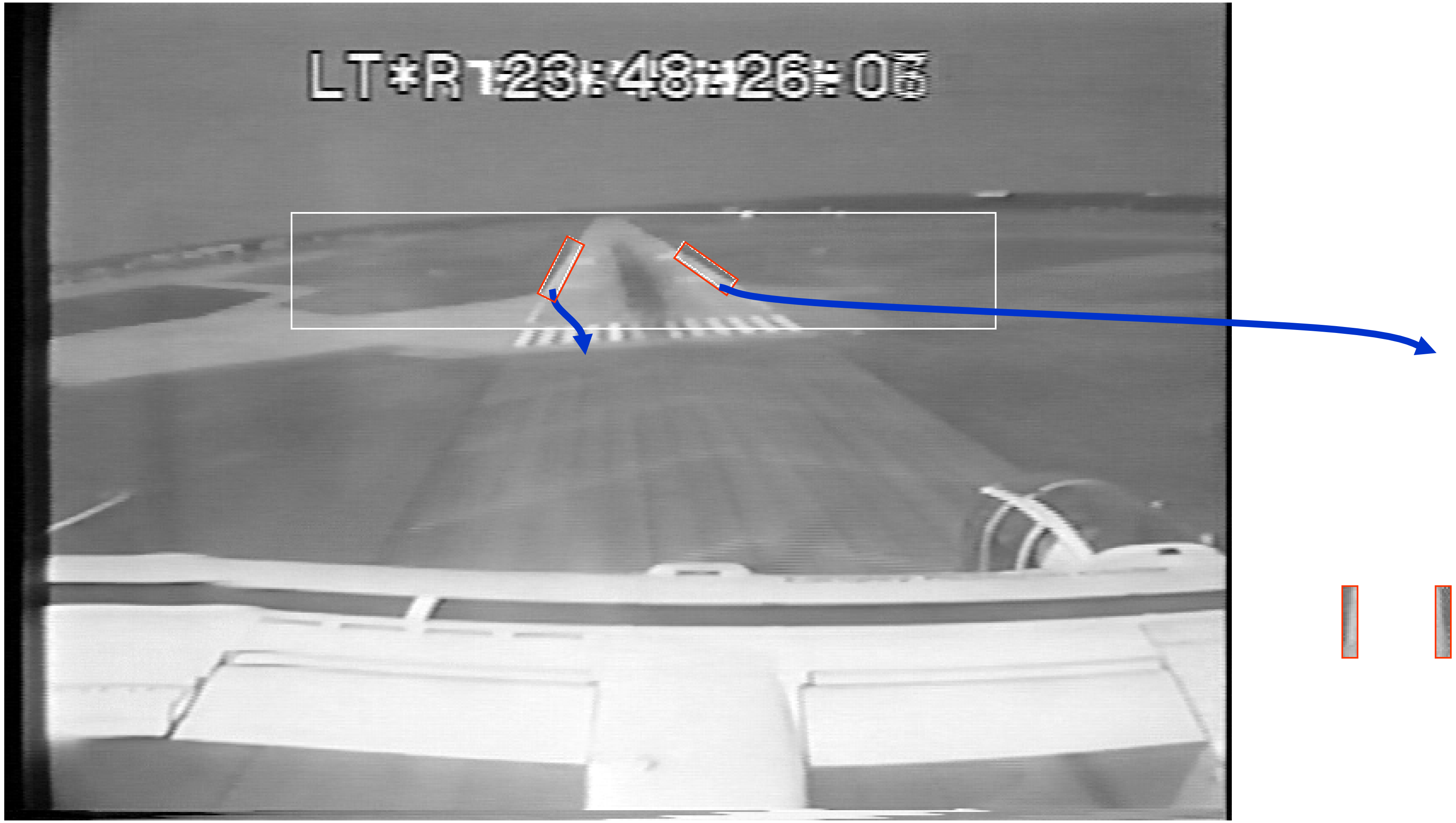


img001.png - 8 bit - 3857x2762
1 of 1
THE MASTER TEMPLATE ITSELF IS NOT GETTING DETECTED
Geometric Matching Setup
Template
Curve Settings
Settings
Template Image
Template Size
Width: 824
Height: 972
New Template
Edit Template
Template Path
D:\personal\ni forum\pattern matching\geotemplate.png
Load from File
OK
Cancel
3857x2762 0.17X (0,0)
Script: Untitled Script 1
Results ... 1 2 3
X Position 3245.60 696.69 1154.78
Y Position 2250.30 2191.28 1260.03
Angle 99.52 275.02 347.75
Scale 99.64 99.40 99.47
Score 972.33 846.85 824.09

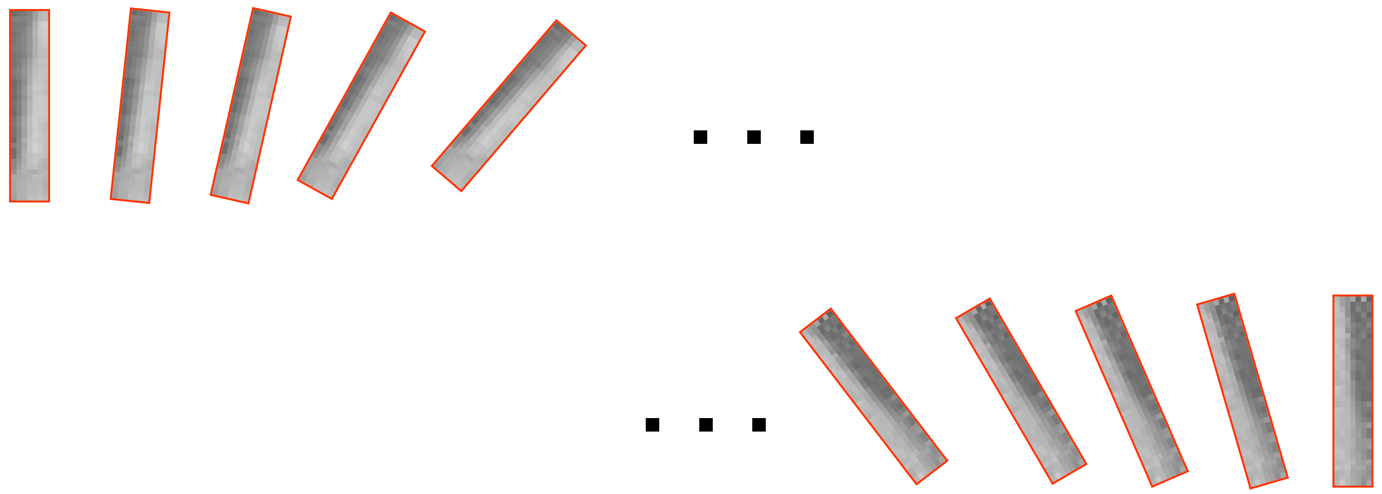
What are some fundamental problems with this approach?

- It is not rotation-invariant
- It is not scale-invariant

Back to the landing application . . .



Generate ***rotated versions*** of the original 2 templates



Perform template matching using <u>all</u> of these, and select the best match

Template matching: pros and cons

- Pros
  - Intuitive, easy to understand
  - Easy to implement
  - Fast? (with appropriate hardware and window sizes)
  - Effective for translational differences
  - Effective in controlled situations, such as industrial settings
- Cons
  - May not work if object undergoes geometric changes in the image (rotation, scale, etc.)
  - May not work if object is partially obscured
  - May not work under variations in illumination
  - Assumes exhaustive search

Template matching is $O(N_I^2 N_T^2) \sim O(N^4)$ – which is horrifyingly bad; there are ways to address execution time

- Multiscale template matching
  - Search at varying resolutions
- Code optimizations – pointer access instead of pixel access methods
- Vector-based correlation
  - Only compare pixel locations that are “interesting”
  - From a set of images, compute the variance
- Hardware assist
- Feature-based methods

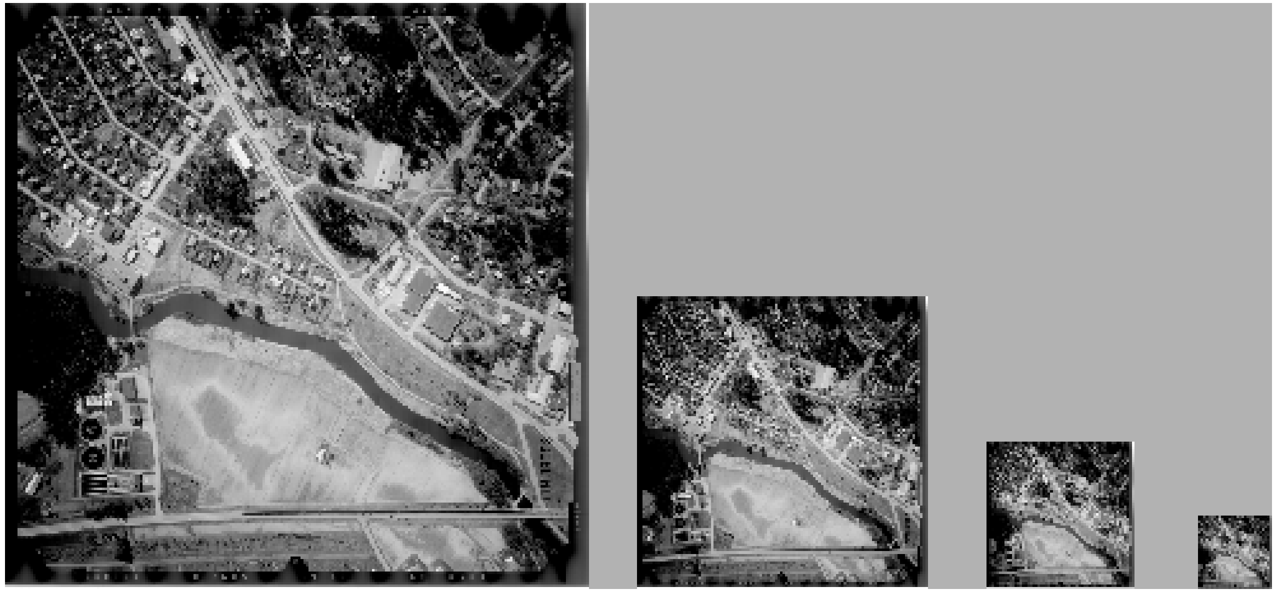
Multiscale template matching

- For each pixel location being examined, we form several products of images the size of the template
- What if we could work at lower resolution?
  - Fewer pixel locations to examine
  - Template is smaller
- Key realization: If the template looks like the image at a certain point, then a lower resolution version of the template will look like a lower resolution version of the image

Steps in multiscale template match

- Form subsampled versions of the template and the image at various resolutions
- Examine the subsampled images to determine which level to start at
- At the starting resolution, find all candidate positions
- For each candidate position, search in the vicinity at full resolution to compute the actual correlation coefficient

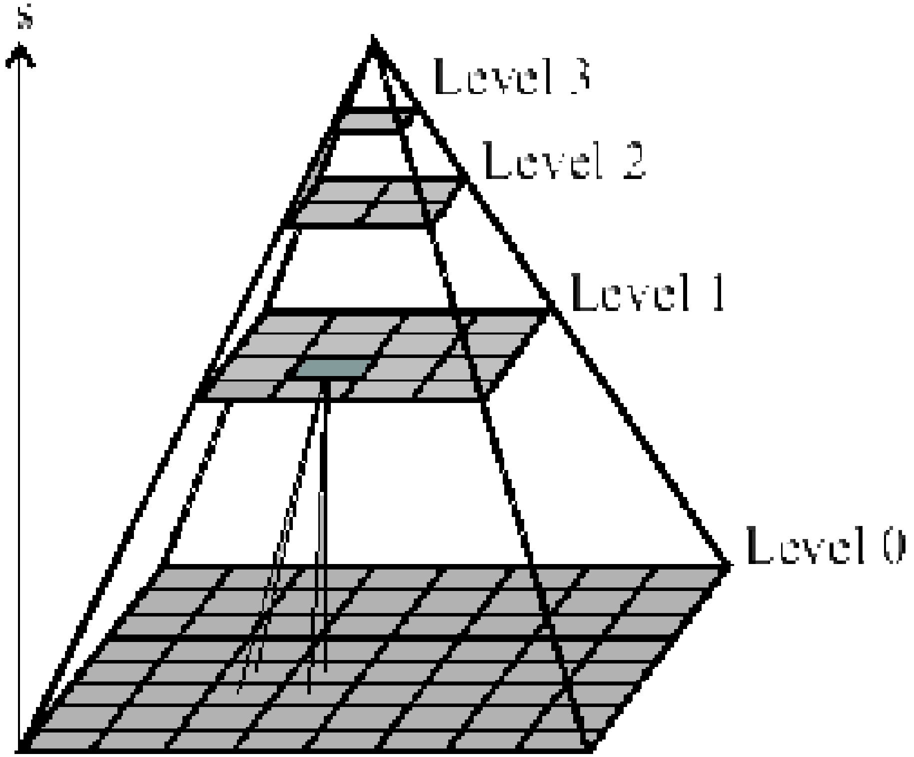
Image Pyramid



- The pyramid contains the same image at several resolutions (usually powers of 2)
- Quick and dirty: subsample the image (throw away pixels)
- Better: represent each group of pixels by their average

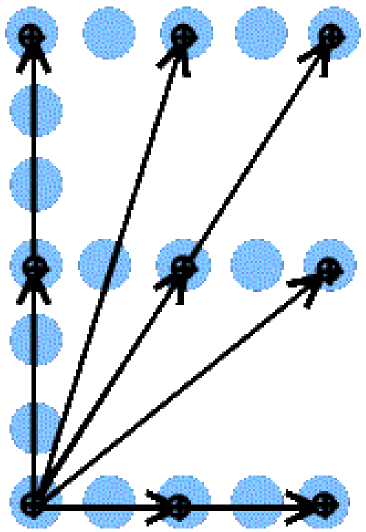
Gaussian pyramid

- Most often, the generation of the lower resolution image is done using a Gaussian filter
  - Our old friend
- Each pixel in the new, subsampled image is replaced by the result of the Gaussian kernel centered at that point
- σ is chosen to keep the proper amount of detail ($\sigma = 1$???)

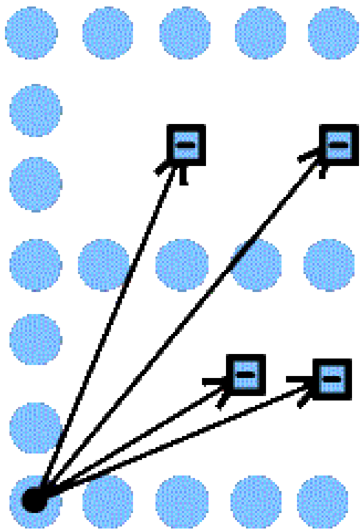


Vector correlation

- Certain points are designated as positive points and negative points for each template
- At each image point, only the relevant points in the template
  are examined
- Significant speed increase is possible
- Website claims this is “patented”



Vectors to
Positive Points
of Symbol "E"



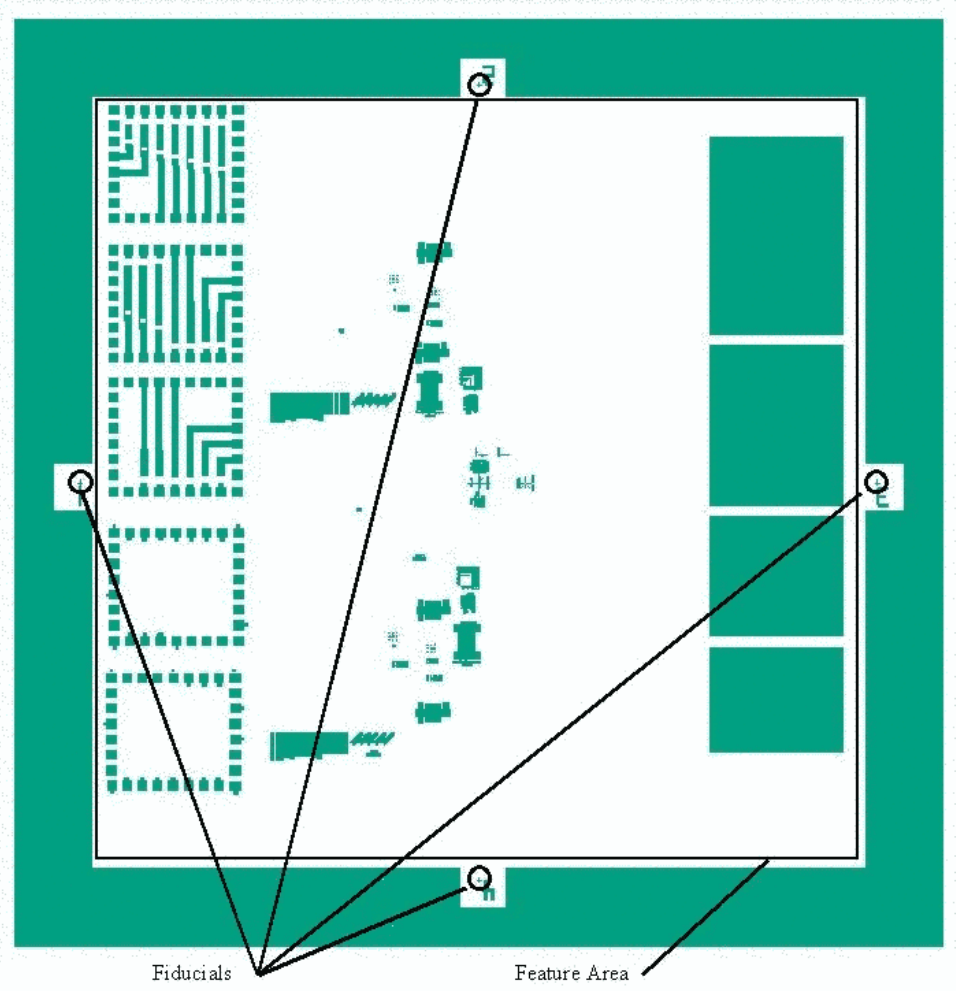
Vectors to
Negative Points
of Symbol "E"

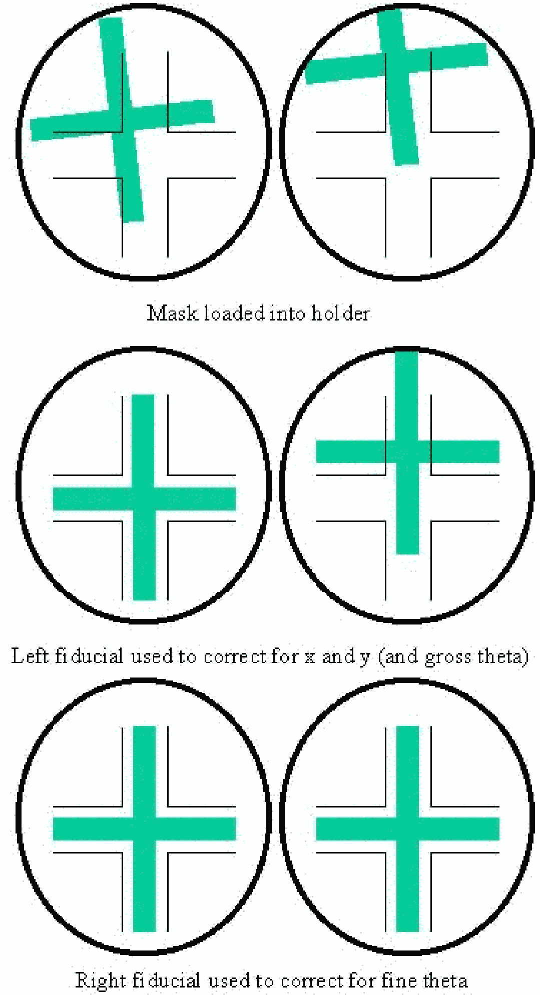
http://www.phoeniximaging.com/vector.htm

Applications of Object Recognition

- OCR (Optical Character Recognition)
  - Find all the A’s, then all the B’s, …
  - You have recognized the text
- Image registration
  - Aligning semiconductor masks
- Target recognition
  - “Find the T-68 tanks in the East German woods”
- Robot / automation guidance

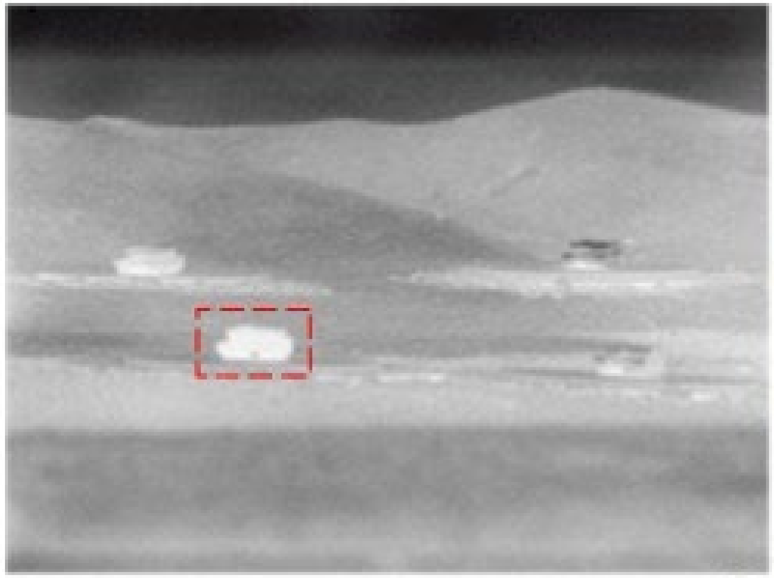
Semiconductor wafer alignment uses template matching





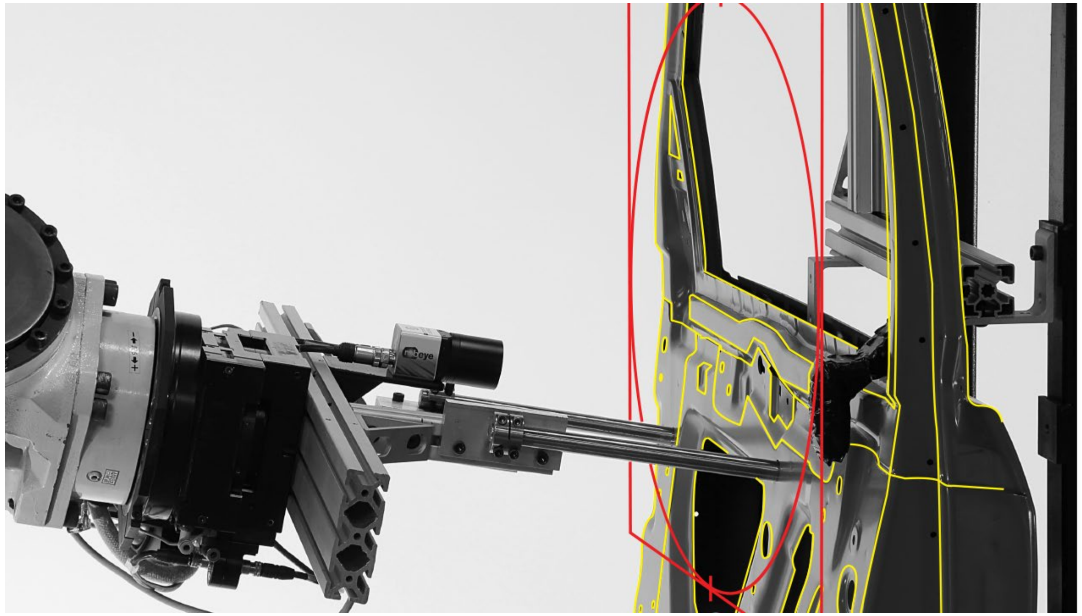
Target Recognition (IR images)

- Multiple templates, representing different angles and scales
  - Geometric transformations
- Several models of tanks

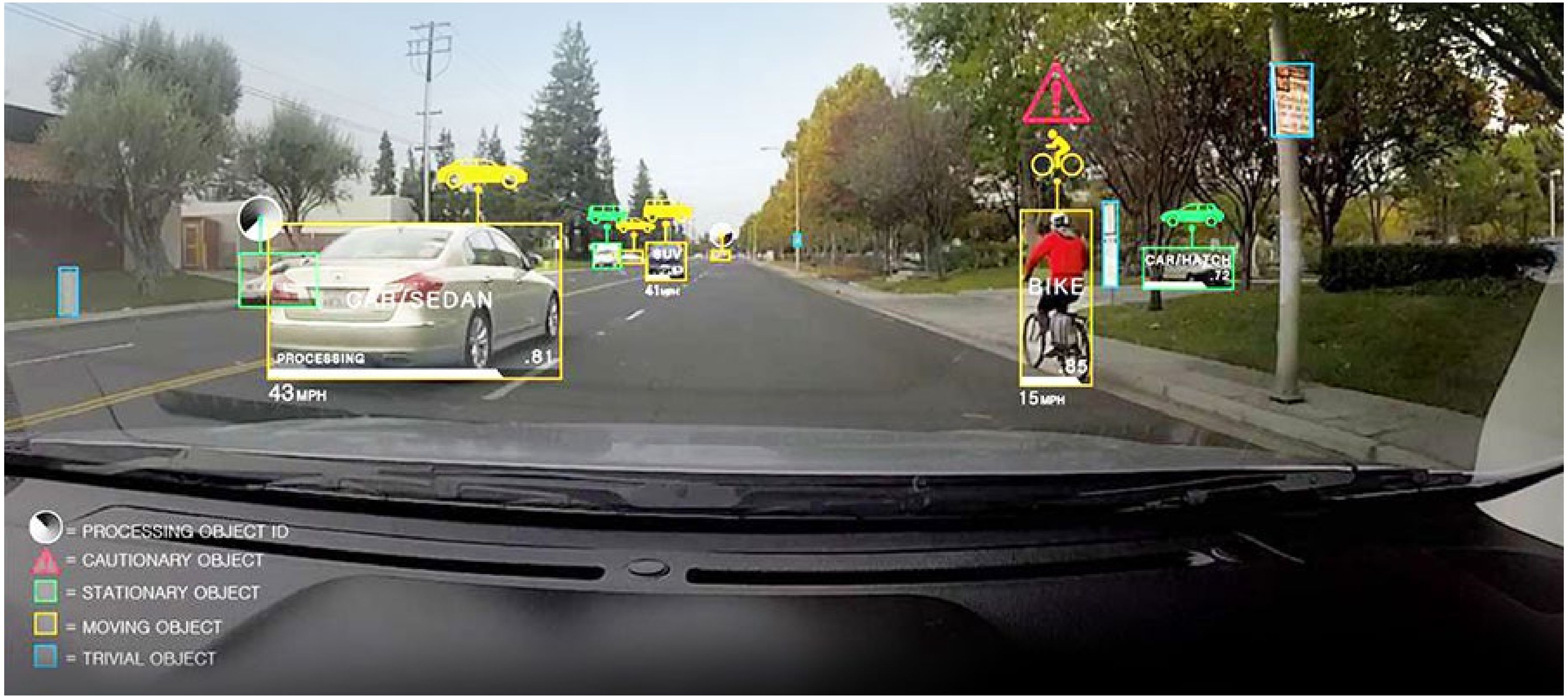


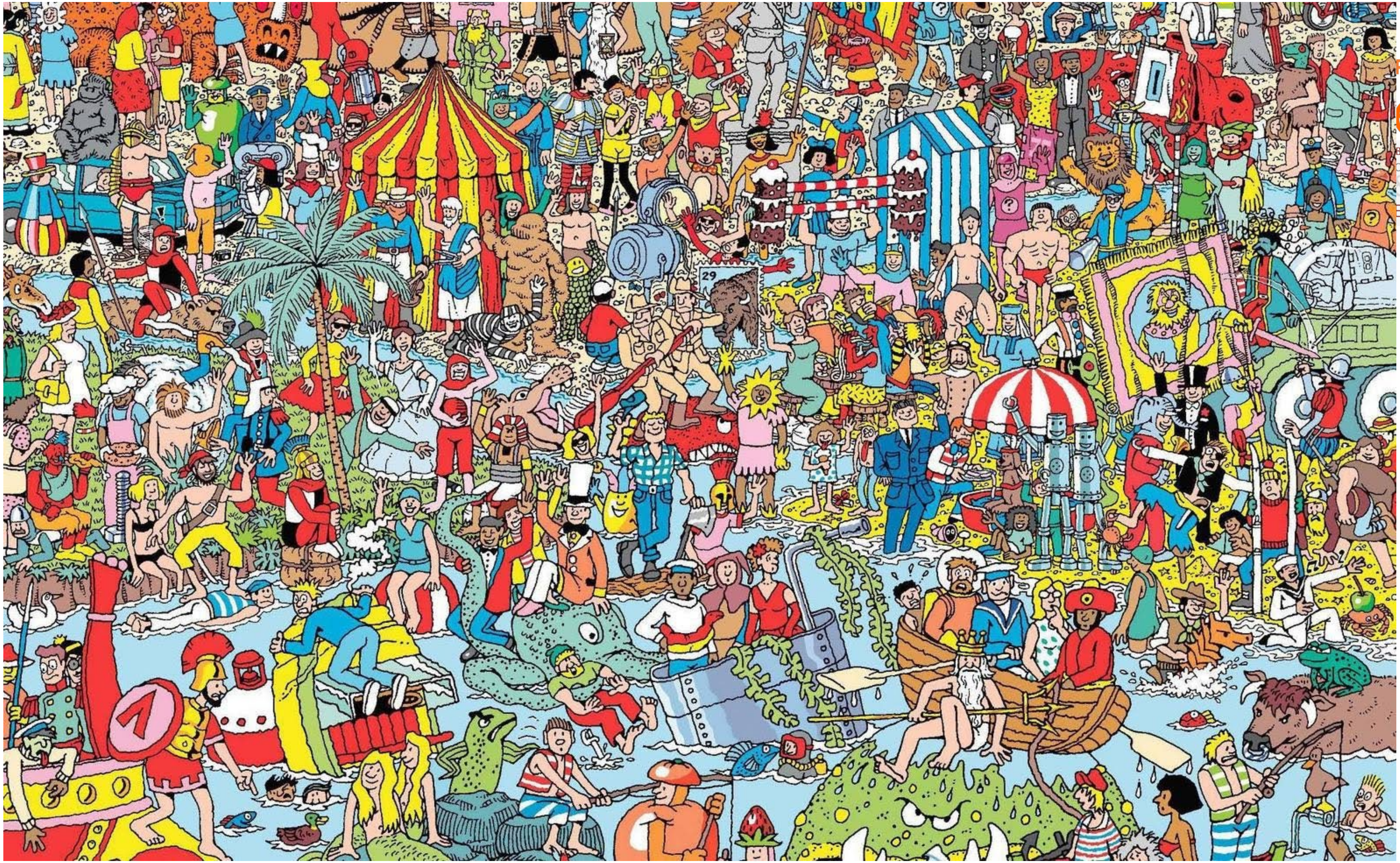
Industrial vision applications of object recognition include "bin-picking", robot guidance and inspection

- "Bin-picking"
- Robot guidance

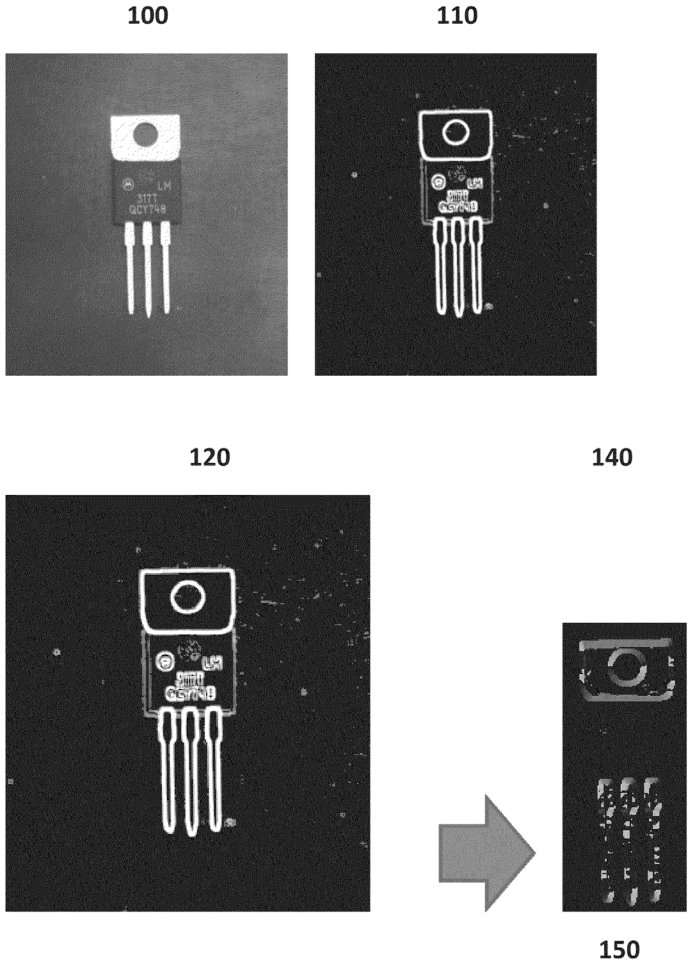


Autonomous vehicles make extensive use of object recognition via template matching





Often, template matching is done on an edge image rather than an original intensity image



(12) **United States Patent**
Van Beek et al.

(10) **Patent No.: US 8,483,489 B2**
(45) **Date of Patent: Jul. 9, 2013**

(54) **EDGE BASED TEMPLATE MATCHING**

(75) Inventors: **Petrus J. L. Van Beek**, Camas, WA (US); **Chang Yuan**, Vancouver, WA (US); **Xinyu Xu**, Vancouver, WA (US); **Xiaofan Feng**, Camas, WA (US); **Xiaofeng Fan**, Billerica, MA (US)

(73) Assignee: **Sharp Laboratories of America, Inc.**, Camas, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 129 days.

(21) Appl. No.: **13/225,271**

(22) Filed: **Sep. 2, 2011**

(65) **Prior Publication Data**

US 2013/0058582 A1 Mar. 7, 2013

(51) **Int. Cl.**
G06K 9/64 (2006.01)

(52) **U.S. Cl.**
USPC **382/217**; 382/103; 382/298

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,842,538	B2	1/2005	Lee et al.	
7,085,418	B2	8/2006	Kaneko et al.	
7,139,432	B2	11/2006	Wenzel et al.	
7,235,782	B2	6/2007	Takane et al.	
7,492,929	B2	2/2009	Itoh et al.	
7,599,512	B2	10/2009	Shimizu et al.	
7,634,139	B2	12/2009	Yang et al.	
7,636,478	B2	12/2009	Bryll	
7,792,191	B2	9/2010	Kwon	
7,792,192	B2	9/2010	Nagarajan	
2004/0071346	A1	4/2004	Clark et al.	
2005/0169512	A1*	8/2005	Fang et al.	382/141
2007/0009159	A1	1/2007	Fan	
2008/0166016	A1	7/2008	Sibiryakov et al.	
2010/0098339	A1	4/2010	Kido	
2012/0082385	A1*	4/2012	Xu et al.	382/199

FOREIGN PATENT DOCUMENTS

EP 2153379 A 10/2008

OTHER PUBLICATIONS

S. Hinterstoisser et al., "Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2010), Jun. 2010, 8 pgs.

* cited by examiner

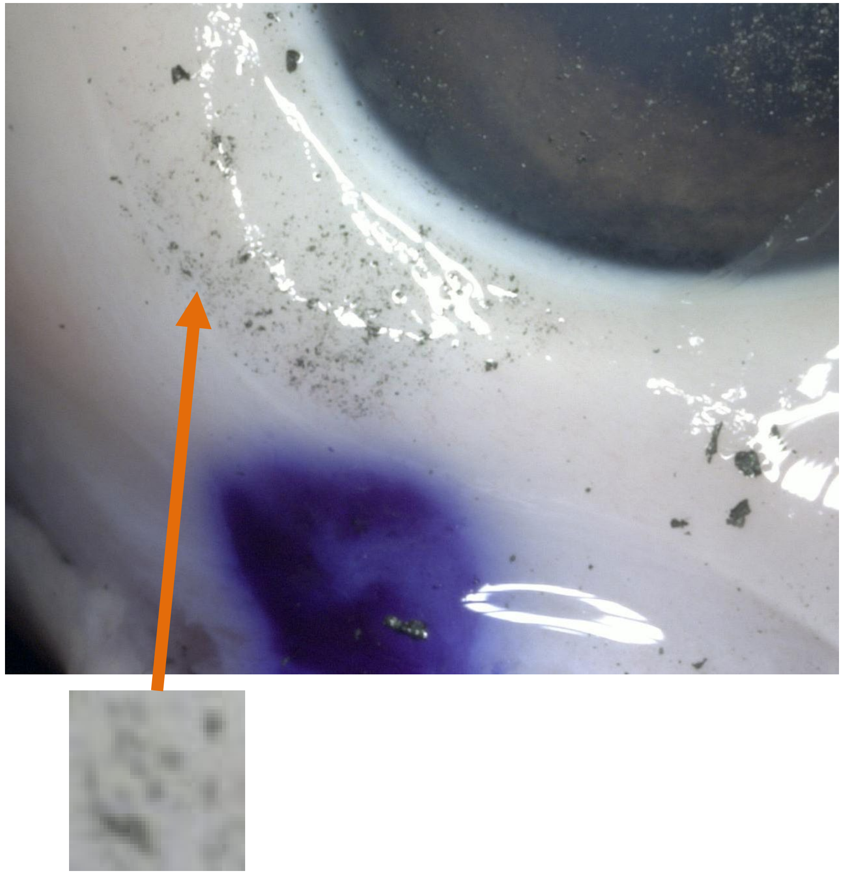
Primary Examiner — Bhavesh Mehta
Assistant Examiner — Siamak Harandi
(74) *Attorney, Agent, or Firm* — Chernoff Vilhauer McClung & Stenzel

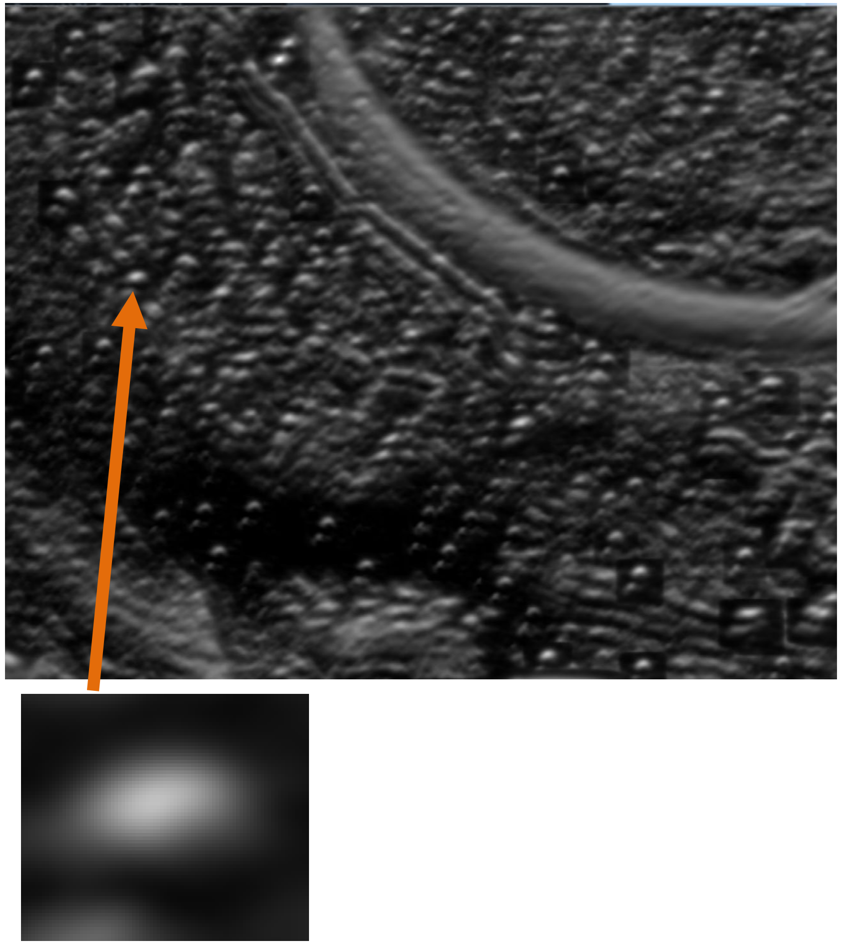
(57) **ABSTRACT**

A method for image processing that includes determining edge pixels of a model image using an edge based technique, and an angular orientation for each of the edge pixels of the model image. The method determines a lower spatial resolution model image based upon the model image and determining respective angular orientations for the lower spatial resolution model image. The method determines edge pixels of an input image using an edge based technique, and an angular orientation for each of the edge pixels of the input image. The method determines a lower spatial resolution input image based upon the input image and determining respective angular orientations for the lower spatial resolution input image. The method matches the lower spatial resolution model image with the lower spatial resolution input image to determine candidate locations of an object within the input image and based upon the candidate locations matching the input image with the model image.

21 Claims, 15 Drawing Sheets

Template matching on grayscale images will produce near-maximal results when images are close to being aligned





Template matching methods (such as normalized grayscale correlation) produce a nearly image-sized result with magnitude equal to the goodness of match

Matching of binary images can use a simpler formulation of the correlation coefficient

The normalized correlation coefficient usually used for grayscale matching:

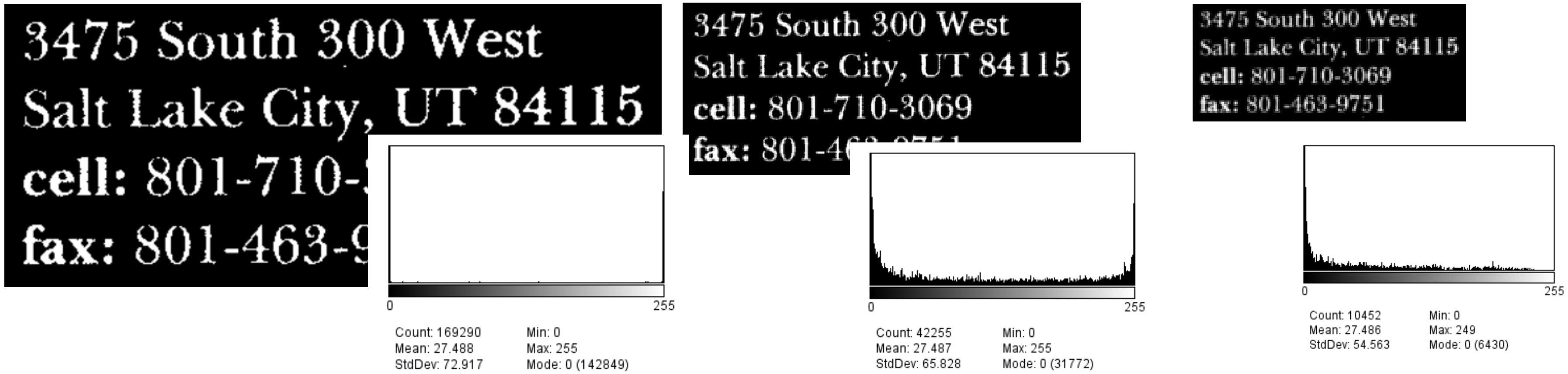
$$R(x, y) = \frac{\sum_i \sum_j (I(x+i, y+j) - \mu_I)(T(i, j) - \mu_T)}{\sqrt{\sum_i \sum_j (I(x+i, y+j) - \mu_I)^2 \sum_i \sum_j (T(i, j) - \mu_T)^2}}$$

is simplified since binary image pixels are always either foreground pr background:

$$R_{bin}(x, y) = \frac{\sum_i \sum_j \left(1 - I(x+i, y+j) - T(i, j)\right)^2}{\sqrt{\sum_i \sum_j \left(I(x+i, y+j)\right)^2 \sum_i \sum_j \left(T(i, j)\right)^2}}$$

With binary images, this “nearby” effect is greatly reduced – which impacts the effectiveness of minor variations in size and rotation, as well as multiscale matching

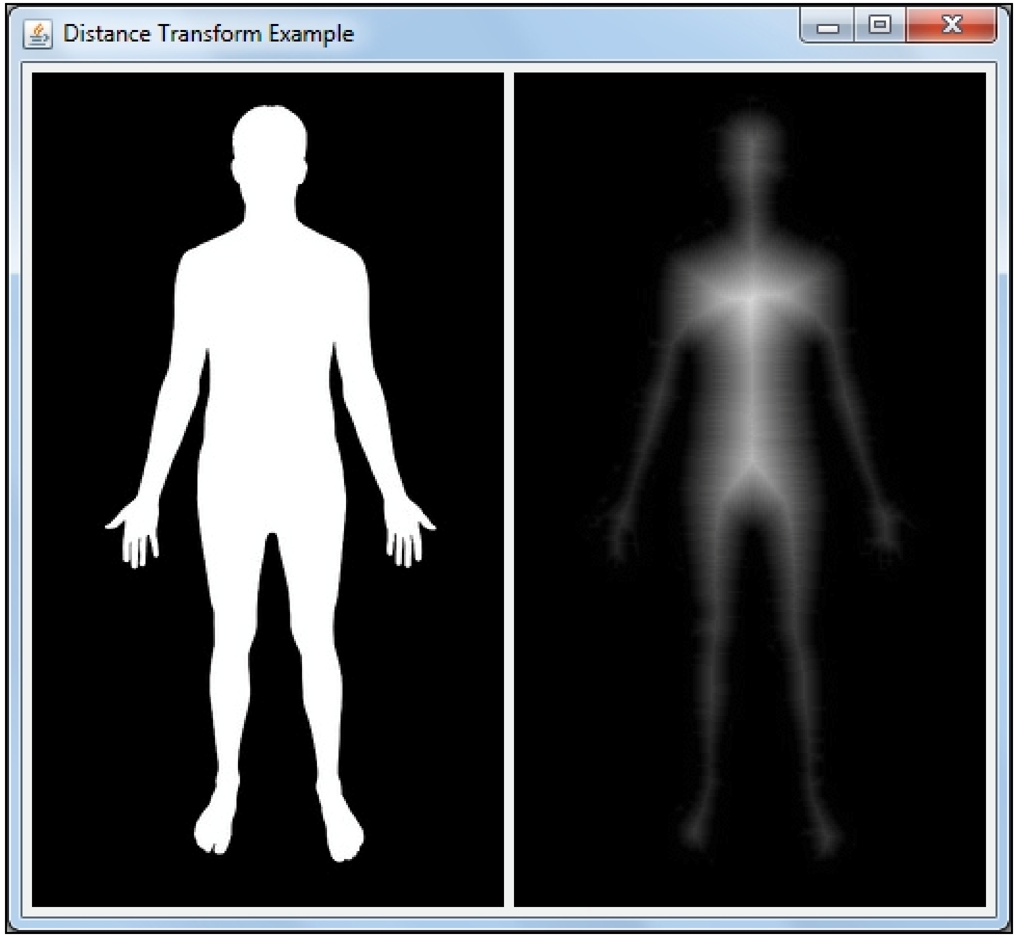
- Since object boundaries in binary images are stark (no smooth transition), matching of nearly-aligned templates can still produce very low match results
  - This is especially true for images with thin objects
- As a consequence, multiscale matching should be done in grayscale
  - A binary base image can still produce multi-valued lower resolution images through averaging



Proximity indication can be restored by applying the *distance transform* to the binary image – values are replaced by the distance from the object boundary

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	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	1	1	1	1	1	1	0
0	1	2	2	2	2	1	0
0	1	2	3	3	2	1	0
0	1	2	2	2	2	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0



The distance transform can be expensive to compute; the *chamfer transform* is an approximation to it that is often used

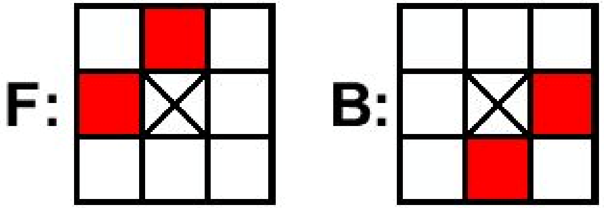
- First, the pixel values are replaced by 0 for background and ∞ (or a huge value) for foreground pixels
- In the forward pass (raster order), each binary pixel is replaced by the minimum of its current distance value and all of the *projected distances* from its neighbors, using a forward distance mask
  - this mask can be $\begin{bmatrix} \sqrt{2} & 1 & \sqrt{2} \\ 1 & x & - \\ - & - & - \end{bmatrix}$ for Euclidean or $\begin{bmatrix} 2 & 1 & 2 \\ 1 & x & - \\ - & - & - \end{bmatrix}$ for Manhattan distance
- Then we conduct a backward pass (starting in lower right corner) using flipped versions of these masks

- Chamfering is process of computing distance to each foreground pixel
- Two-pass algorithm
  - Traverse top-to-bottom, left-to-right
  - Traverse bottom-to-top, right-to-left

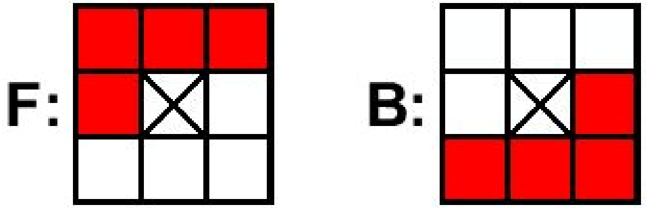
  In each case, compute minimum distance

$$f_1(p) = \begin{cases} 0 & \text{if } p \in P \\ \min\{ f_1(q) + 1 : q \in F(q)\} & \text{otherwise} \end{cases}$$

$$f_2(p) = \min\{ f_1(p), f_2(q) + 1 : q \in B(q)\}$$



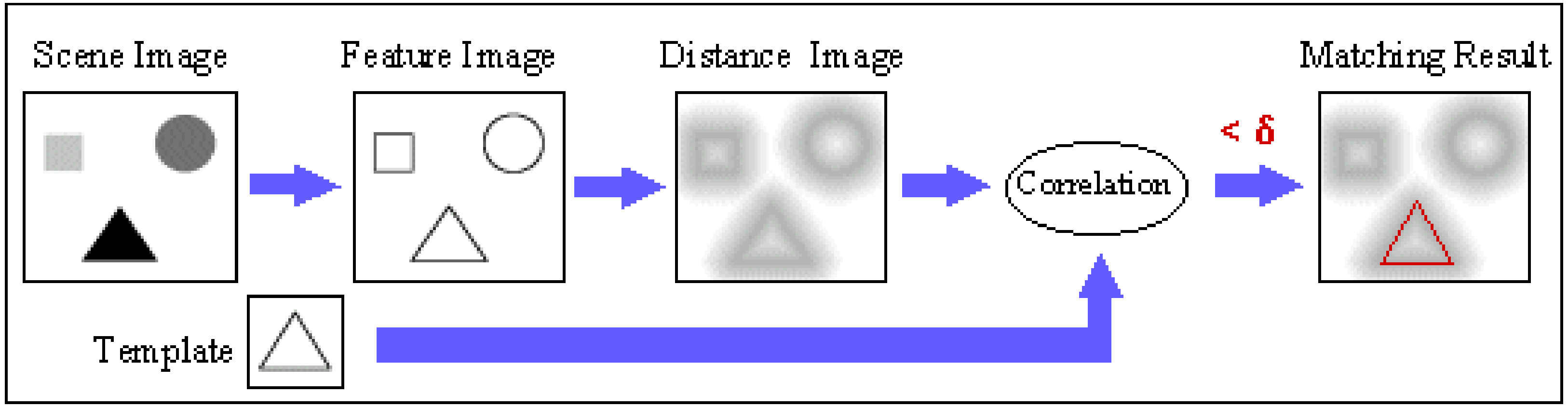
4-connectivity



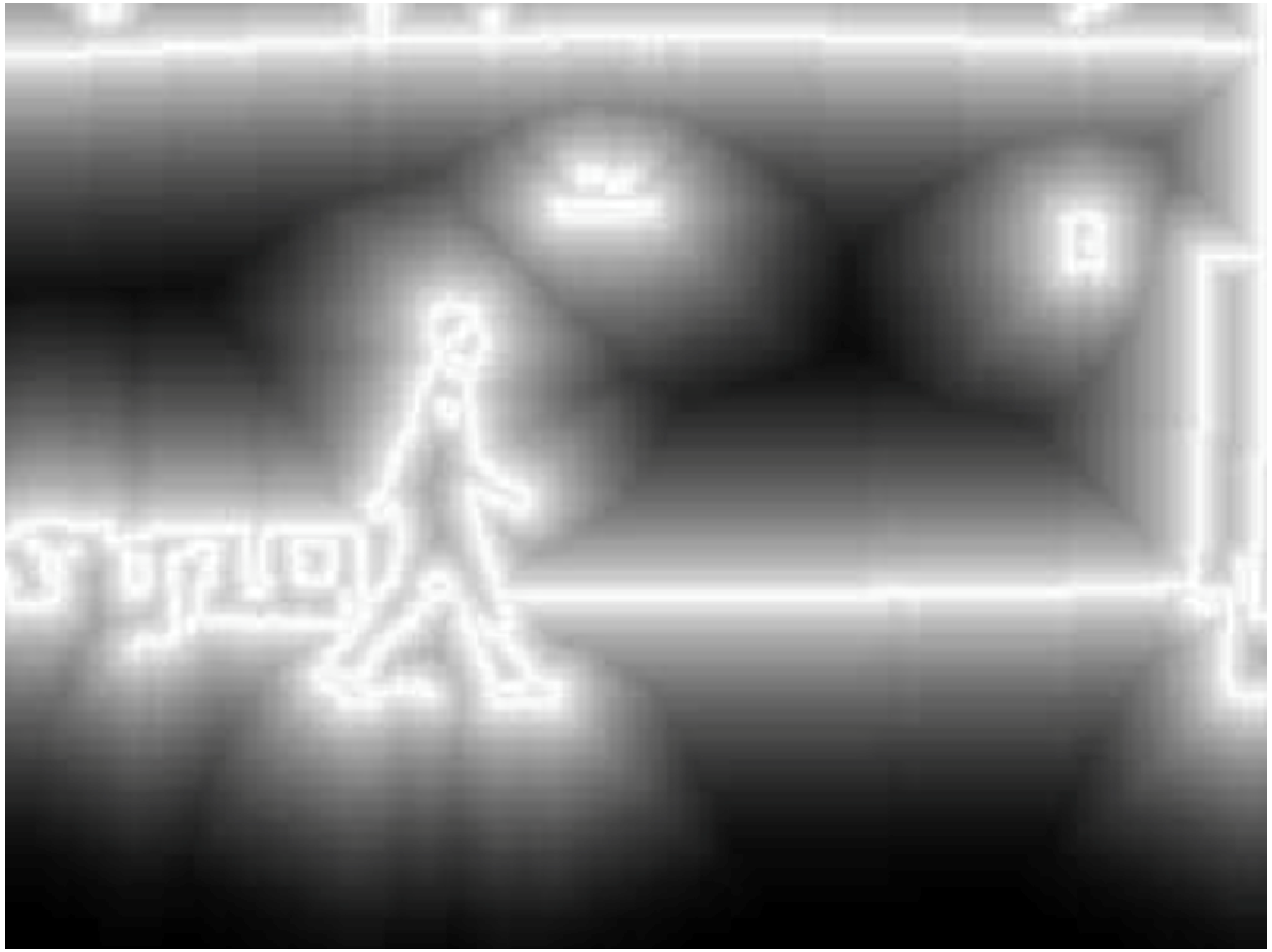
8-connectivity

When the distance transform has been applied, matching can be carried out in the usual manner – or using a number of optimizations

- Because the chamfer transform only requires two passes through the image, it's efficient
- Match results can be shown to be the same as for the original image



Variations of the distance transform exist for edge images, to facilitate matching using edges



Today's Objectives

- Area-based matching
  - Example application: vision-guided aircraft landing
  - Correlation coefficient
- Normalized grayscale correlation
- Variations
  - Pyramid-based matching
  - Vector matching
- Applications
  - Guidance
  - Optical Character Recognition

Template matching on edge images

Binary image template matching

- Overlap counting
- The distance transform
- Chamfer matching