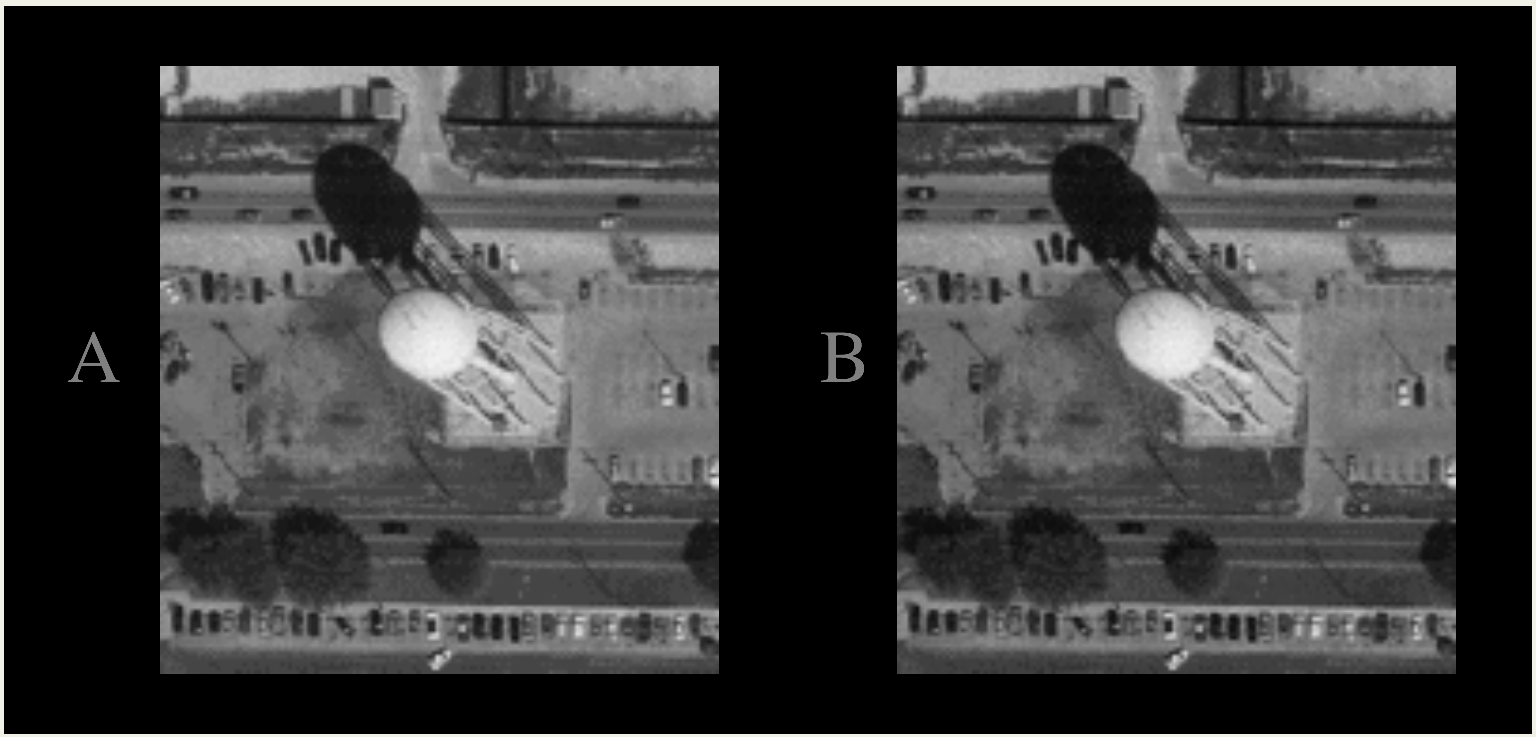
Image Matching

Lecture #8

February 18, 2019

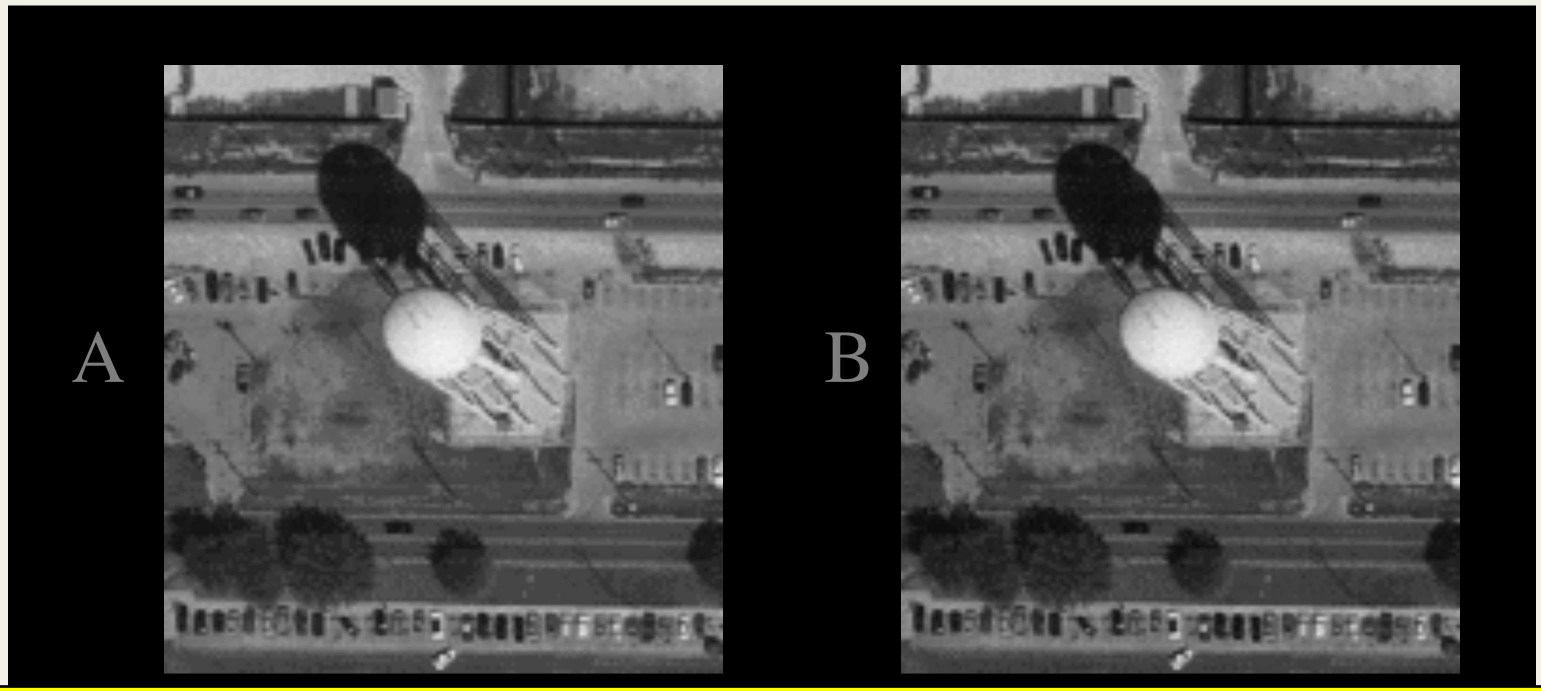
Colorado State University

How do we (directly) compare two images?



Are these images the same? Are they similar?

Pixel-wise Comparison



Or, normalized by image area, about 5 grey values per pixel.

$$8{,}140 = \sum_{x}^{x<148} \sum_{y}^{y<161} \left| A(x,y) - B(x,y) \right|$$

Backup - what is “Similarity”?

Consider two vectors/points.

$$X = \begin{vmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{vmatrix} \quad Y = \begin{vmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{vmatrix}$$

Distance vs. similarity:

$$S : R^n \times R^n \rightarrow R$$

$$D : R^n \times R^n \rightarrow R$$

$$S \propto {}^1\!/_D$$

Common Approaches

- Euclidean (L2) Distance
- City Block (L1) Distance
- Pearson’s Correlation

Slightly Less Common

- Mahalanobis Distance
- Mutual Information

Colorado State University

Simple Distances (norms)

L_1 - City Block Distance $$\sum_{x,y}\left(\left|A(x,y)-B(x,y)\right|\right)$$

L_2 - Euclidean Distance $$\sqrt{\sum_{x,y}\left(A[x,y]-B[x,y]\right)^2}$$

L_∞ - Max Distance $$\underset{x,y}{Max}\left|A[x,y]-B[x,y]\right|$$

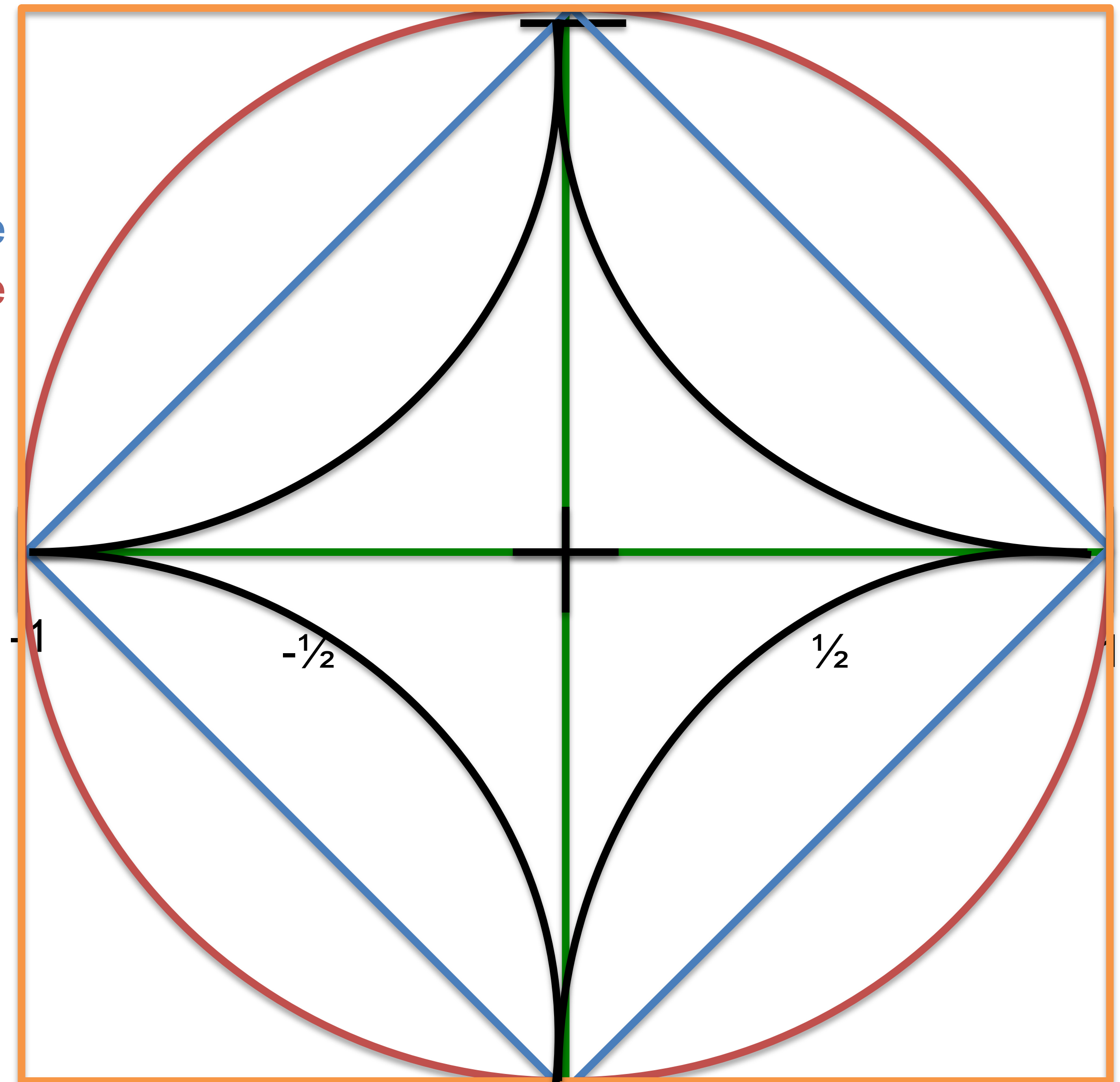
Generalized L-norm $$\sqrt[l]{\sum_{x,y}\left(\left|A(x,y)-B(x,y)\right|\right)^l}$$

L_1 Distance

L_2 Distance

L_∞

$L_{½}$



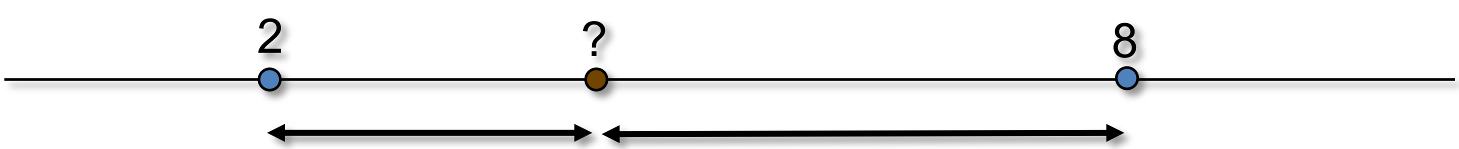
Curves shown are the set of points that are 'one unit' from the origin using different definitions of distance.

Properties of L1 Distance

Consider the following problem:

Find the unique point "closest" to k other points.

For simplicity, do this in R (a line) with k = 2.



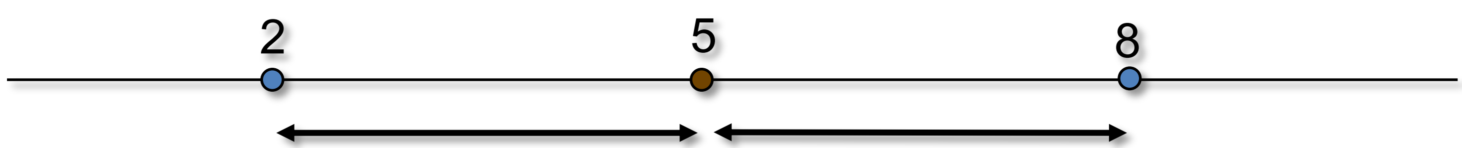
See the problem yet?

$$|2-3|+|8-3|=6 \qquad |2-4|+|8-4|=6$$

In Comparison, Consider L2

Find the unique point "closest" to k other points.

Using L2,



$$\sqrt{(2-5)^2+(8-5)^2}=\sqrt{18} \qquad \sqrt{(2-4)^2+(8-4)^2}=\sqrt{20}$$

Best | Not as Good

Pearson's Correlation

$$\frac{\sum_{x,y}\left(A(x,y)-\overline{A}\right)\left(B(x,y)-\overline{B}\right)}{\sqrt{\sum_{x,y}\left(A(x,y)-\overline{A}\right)^2}\sqrt{\sum_{x,y}\left(B(x,y)-\overline{B}\right)^2}}$$

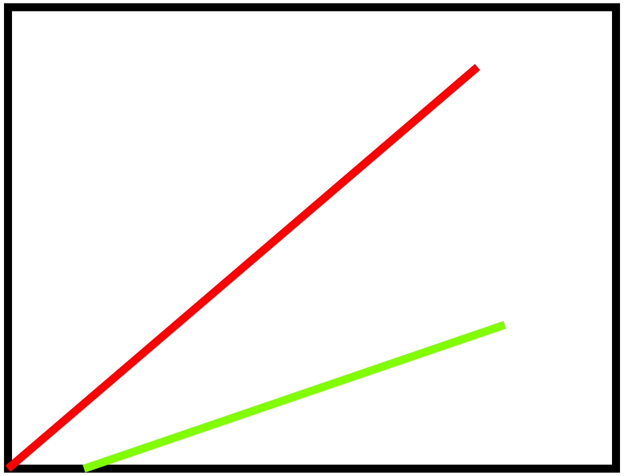
What is the underlying model?

Assumptions of Correlation

$$\frac{\sum_{x,y}\left(A(x,y)-\overline{A}\right)\left(B(x,y)-\overline{B}\right)}{\sqrt{\sum_{x,y}\left(A(x,y)-\overline{A}\right)^2}\sqrt{\sum_{x,y}\left(B(x,y)-\overline{B}\right)^2}}$$

- Two signals vary linearly
  - Constant shift to either signal has no effect.
  - Increased amplitude has no effect.
- This minimizes sensitivity to:
  - changes in (overall) illumination.
  - offset or gain.

Special Cases



- Any two linear functions with positive slope have correlation 1.
  - Only the sign of the slope matters.
- Any two linear functions with differently signed slopes have correlation -1.
  - This is called anti-correlation
  - Anti-correlation = correlation for prediction.
  - For matching, it may or may not be as good…
- Correlation undefined for slope = 0 (σ=0)

Correlation (cont.)

- For Images, correlation is sensitive to:
  - Translation
  - Rotation: in-plane and out-of-plane
  - Scale
- Because it …
  - Assumes pixels align one atop the other.
  - Compares two images pixel by pixel.
- Translation handled by convolution
  - Example, alignment by template matching

Computing Correlation

- Remember adding a constant does not change correlation to any other signal, so
  - Let's subtract average A from A()
  - Let's subtract average B from B()
  - The mean of both signals is now zero
  - Then correlation reduces to:

$$\frac{A \cdot B}{\sqrt{\sum_{x,y}\left(A(x,y)-\overline{A}\right)^2}\sqrt{\sum_{x,y}\left(B(x,y)-\overline{B}\right)^2}}$$

Colorado State University

Computing Correlation (II)

- For zero-mean signals, we can scale them without changing their correlation scores
  - Multiply A by the inverse of its length
  - Multiply B by the inverse of its length
  - Both signals are now unit length
  - Then correlation reduces to:

$$A \cdot B$$

- Gives rise to 'Correlation Space'.

Correlation Space

- Why zero-mean & unit-length images?
- Consider database retrieval
  - Compare new image A …
  - with many images in database.
  - When database images are stored in their zero-mean & unit-length form, then
  - Preprocess A (zero-mean, unit-length)
  - Compute dot products

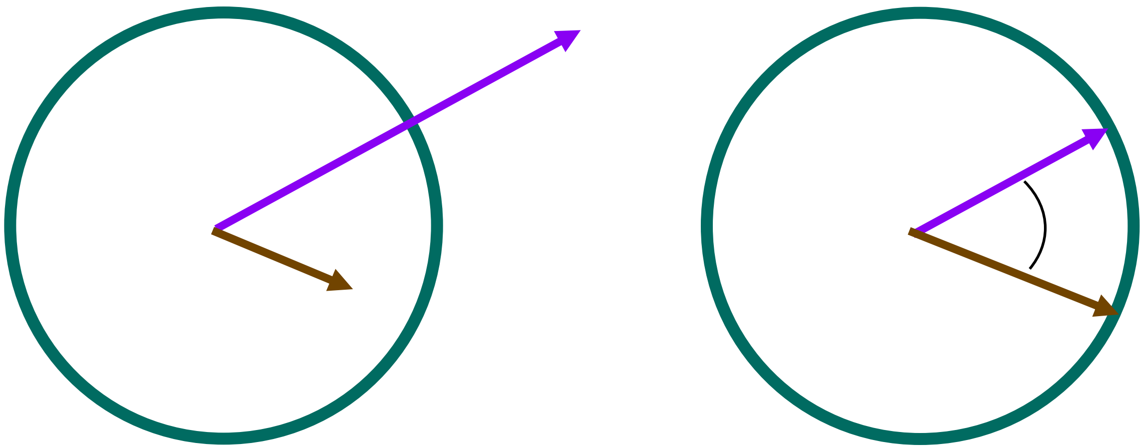
Correlation Space (II)

- New idea: image as a point in an N dimensional space
    - N = width x height
- Zero-mean & unit-length images lie on an N-1 dimensional “correlation space” where the dot product equals correlation.
    - – This is a highly non-linear projection.
    - – Points lie on an N-1 surface within the original N dimensional space.
- So consider points in 3-D ….

Colorado State University

Correlation Space (II)

- Subtracting mean - translation.
- Length one - project onto sphere.
- Correlation is then:
  – Cosine of angle between vectors (points).



Useful Connection …

- Euclidean distance inverse of correlation in correlation space.

$$
\begin{aligned}
\sqrt{\sum_{x,y}\left(A[x,y]-B[x,y]\right)^2} &= \sqrt{\sum_{x,y}A[x,y]^2+\sum_{x,y}B[x,y]^2-2A[x,y]B[x,y]} \\
&= \sqrt{1+1-2\sum_{x,y}A[x,y]B[x,y]} \\
&= \sqrt{2-2A\cdot B} \\
&= \sqrt{2-2Corr(A,B)}
\end{aligned}
$$

Nearest-neighbor classifiers in correlation space maximize correlation

Colorado State University

Limitations

- To match images this way, they must be
  - The same width & height
  - In correspondence : coordinates match
- More importantly, objects in the scene must
  - Be in the same location
  - Be at the same scale
  - Be at the same orientation
  - Be seen from the same viewpoint

Colorado State University

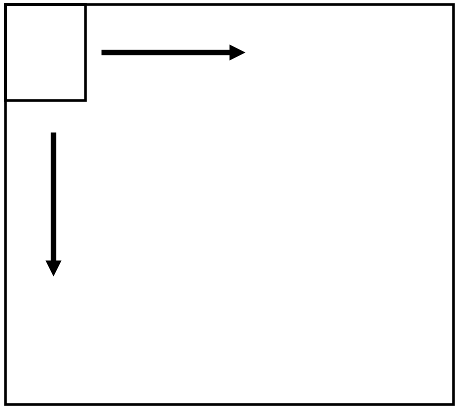
Find similar patterns in a larger image

- The image above is a small piece of the image to the right. But from where?

Colorado State University

Brute-Force Translation Invariance

To find a small image in a large one, "slide" the small one across the large, computing Pearson's correlation at every possible position.



Statistical Cross-Correlation

- The process of "slide & correlate" is called cross-correlation
- Complexity is O(nm)
  - N = # of pixels in image ($w \times h$)
  - M = # of pixels in the template ($w \times h$)
- Highly parallel (every position can be computed independently)
- Still sensitive to
  - Rotation
    - in-plane
    - out-of-plane
  - Scale
  - Perspective

Computing Cross-Correlation

- In cross-correlation, the mask is correlated repeatedly to image windows
  - zero-mean & unit length the mask
  - zero-mean & unit length the image
  - compute the sliding dot product

This is *almost* convolving the image with the mask

Colorado State University

Naming conventions

- In Engineering, convolving a normalized mask with the source image is called correlation
  - Is this exactly the same as Pearson's correlation?
  - Why or why not?
- This is the most common definition of correlation in image processing texts

Colorado State University

Application: Tracking

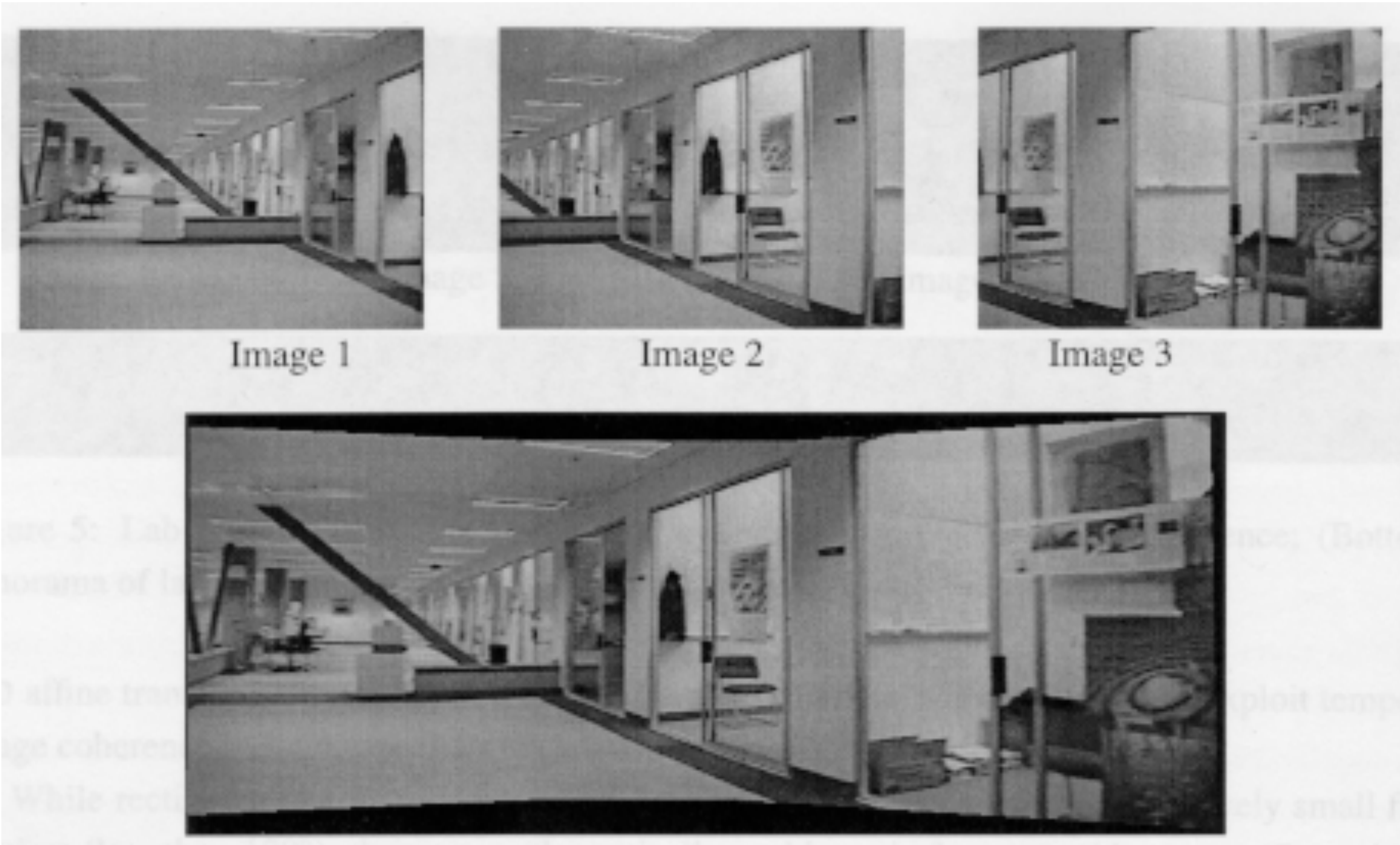
- Cut out a picture of a target from the first frame of a video
  - Use it as a template /mask
- Correlate the target in the following frames
  - Find the location with the highest correlation
- Improvement:
  - update target with each new frame

Application: Tracking

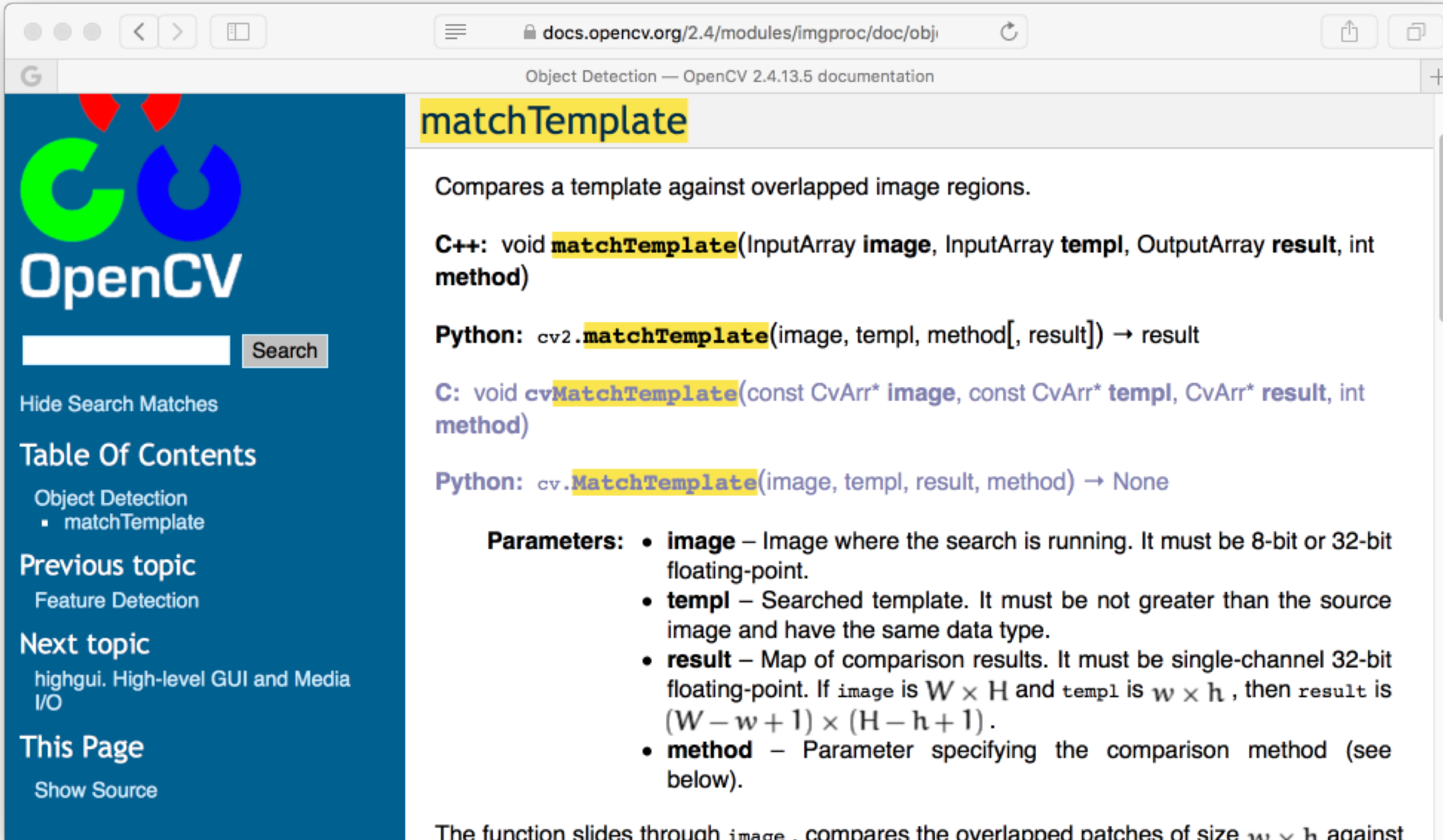
Application: Mosaicing

- Take several, overlapping images from a translating camera
  - Camera cannot move along optical axis
- Correlate the whole images to each other
  - Find location where they match the best
  - Stitch them into a single, larger image

Mosaicing (II)



In OpenCV


docs.opencv.org/2.4/modules/imgproc/doc/obj

Object Detection — OpenCV 2.4.13.5 documentation

OpenCV

Search

Hide Search Matches

Table Of Contents

Object Detection
- matchTemplate

Previous topic

Feature Detection

Next topic

highgui. High-level GUI and Media I/O

This Page

Show Source

matchTemplate

Compares a template against overlapped image regions.

C++: void **matchTemplate**(InputArray **image**, InputArray **templ**, OutputArray **result**, int **method**)

Python: `cv2.matchTemplate(image, templ, method[, result])` → result

C: void **cvMatchTemplate**(const CvArr* **image**, const CvArr* **templ**, CvArr* **result**, int **method**)

Python: `cv.MatchTemplate(image, templ, result, method)` → None

Parameters:
- **image** – Image where the search is running. It must be 8-bit or 32-bit floating-point.
- **templ** – Searched template. It must be not greater than the source image and have the same data type.
- **result** – Map of comparison results. It must be single-channel 32-bit floating-point. If `image` is $W \times H$ and `templ` is $w \times h$, then `result` is $(W - w + 1) \times (H - h + 1)$.
- **method** – Parameter specifying the comparison method (see below).

The function slides through `image`, compares the overlapped patches of size $w \times h$ against

Example

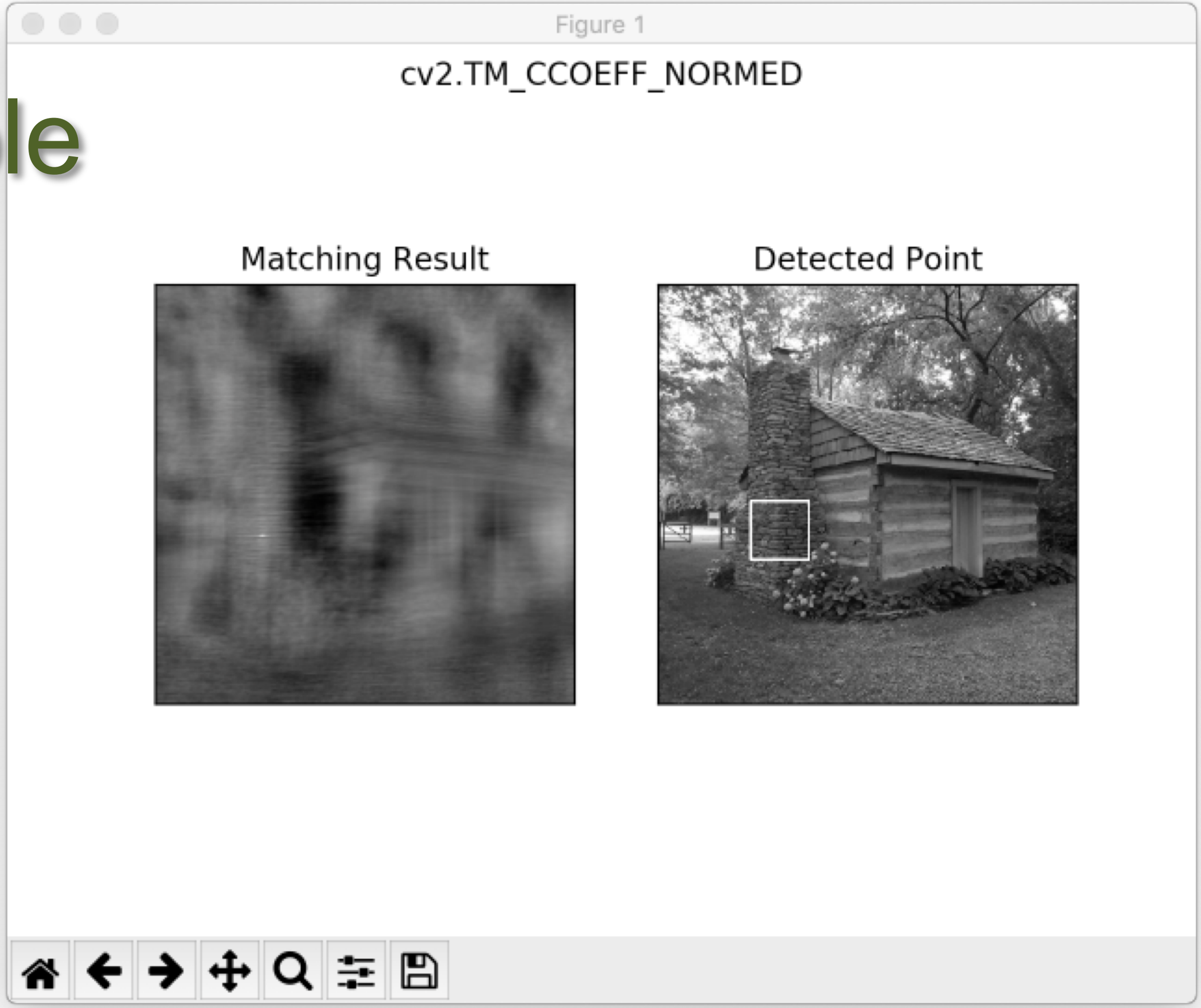


Figure 1
cv2.TM_CCOEFF_NORMED
Matching Result
Detected Point