Template Matching
Image Patch Features

- We often want to find distinctive points in the image
  - Could match these points to a 3D model, for object recognition
  - Or track them from one image to another, for motion or structure estimation

- A point can be described by the appearance of a small image “patch”, or template, surrounding the point

Patches are 15x15 pixels
Sum of Squared Differences (SSD)

• Say we want to match a patch (template) from image $I_0$ to image $I_1$
  – We’ll assume that intensities don’t change, and there is only translational motion within the image
  – So we expect that
    \[ I_1(x_i + u) = I_0(x_i) \]
  – In practice we will have noise, so they won’t be exactly equal

• But we can search for the displacement $u=(x,y)$ that minimizes the sum of squared differences (SSD):
  \[
  E(u) = \sum_i w(x_i) [I_1(x_i + u) - I_0(x_i)]^2
  \]
  \[ w(x_i) \] is an optional weighting function, that weights the center of the patch higher than the periphery
Cross Correlation

• Expanding the expression for SSD:

\[ E(u) = \sum_{i} \left[ I_1(x_i + u) - I_0(x_i) \right]^2 \]

\[ = \sum_i I_1(x_i + u)^2 - 2 \sum_i I_1(x_i + u)I_0(x_i) + \sum_i I_0(x_i)^2 \]

• The last term is just the sum of intensities in the template \( I_0 \), which is a constant.

• The first term is just the sum of intensities in \( I_1 \) under the template ... it’s independent of the template.

• The middle term is the cross-correlation of \( I_1 \) and \( I_0 \) ... it is high when \( I_1 \) matches \( I_0 \)

• So, to minimize SSD, we maximize the cross-correlation score
Vector representation of correlation

- Correlation is a sum of products of corresponding terms
  \[ c(x, y) = \sum_{s=-m/2}^{m/2} \sum_{t=-n/2}^{n/2} w(s, t) f(x + s, y + t) \]
  \[ = w(x, y) \otimes f(x, y) \]

- We can think of correlation as a dot product of vectors \( w \) and \( f \)
  \[ c = w_1 f_1 + w_2 f_2 + \ldots + w_{mn} f_{mn} = w \cdot f \]

- If images \( w \) and \( f \) are similar, their vectors (in \( mn \)-dimensional space) are aligned
  \[ c = |w||f| \cos \theta \]
Correlation can do matching

- Let $w(x,y)$ be a template sub-image
- Try to find an instance of $w$ in image $f(x,y)$
- The correlation score $c(x,y)$ is high where $w$ matches $f$

$$c(x, y) = \sum_{s=-m/2}^{m/2} \sum_{t=-n/2}^{n/2} w(s, t) \ f(x+s, y+t)$$

$$= w(x, y) \otimes f(x, y)$$
Template Matching (continued)

• Since \( f \) is not constant everywhere, we need to normalize

\[
c = \frac{\mathbf{w} \cdot \mathbf{f}}{|\mathbf{w}| |\mathbf{f}|} = \cos \theta
\]

• We can get better precision by subtracting off means

\[
c(x, y) = \frac{\sum_{s,t} [w(s, t) - \bar{w}] [f(x + s, y + t) - \bar{f}]}{\left\{ \sum_{s,t} [w(s, t) - \bar{w}]^2 \sum_{s,t} [f(x + s, y + t) - \bar{f}]^2 \right\}^{1/2}}
\]

This is the normalized cross correlation coefficient

Range: -1.0 ... +1.0

Perfectly opposite
Perfect match
Matlab – normxcorr2

- \( C = \text{normxcorr2}(T, I) \)
  - Does a normalized cross correlation
  - Inputs: template image \( T \), and target image \( I \)
  - Output: correlation scores image \( C \) (values range from -1..+1)
Note on normxcorr2

• The Matlab function normxcorr2 correlates a template T with image I
  – To find the maximum, can do
    
    ```matlab
    C = normxcorr2(T,I);   % Do normalized cross correlation
    cmax = max(C(:));
    [y2,x2] = find(C==cmax);
    ```

• Assume T is size (2*M+1)x(2*M+1)
  – Then the resulting score image is bigger than I, by M rows and M columns along each side and top and bottom
  – You should subtract off M from the coordinates
    
    ```matlab
    y2 = y2-M;
    x2 = x2-M;
    ```
% Match a point
clear all
close all

I0 = imread('test000.jpg');  % Get first image
imshow(I0, []);

% The user will designate the location of a template from first image.
% The function "ginput" gets the location of a mouse click.
[x,y] = ginput(1);  % The "1" means to just get one click
x = round(x);
y = round(y);

% The size of the template will be (2M+1)x(2M+1).
M = 9;
rectangle('Position', [x-M y-M 2*M+1 2*M+1], 'EdgeColor', 'r');

% Extract template. (Note: if (x,y) too close to border, this will
% error)
T = I0(y-M:y+M, x-M:x+M);
figure, imshow(T, []);
Source image I0

Template T
I1 = imread('test012.jpg'); % Get second (target) image
C = normxcorr2(T,I1);       % Do normalized cross correlation
figure, imshow(C, []);
% Find the location of the maximum score value.
cmax = max(C(:));
[y2,x2] = find(C==cmax);
% Correct the location by subtracting off half the template size.
y2 = y2-M;
x2 = x2-M;
% Show the location of the match in the target image.
figure, imshow(I1, []);
rectangle('Position', [x2-M y2-M 2*M+1 2*M+1], 'EdgeColor', 'r');
fprintf('Correlation score = %f at (x,y) = (%d,%d)
', cmax, x2, y2);
Correlation score = 0.851861 at (x,y) = (268,70)