# Image Features

**Goal:** We introduce and motivate several types of image features. These are each designed for specific vision tasks.

We consider features to support the following tasks:

I. Matching image patches between images with significantly different viewpoints, $\Leftarrow$ **Today**

II. Extracting image landmarks; a) their $(x, y)$ position,

III. Extracting image landmarks; b) their scale, and c) their orientation.

**Readings:** Szeliski, Section 4.1 and 4.2.

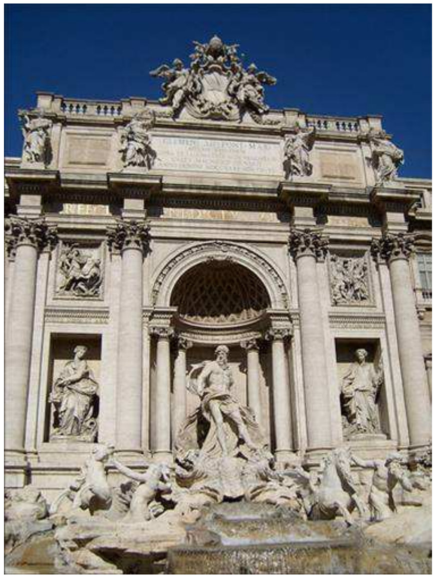
  

# Part I: Image Patch Matching

The problem of identifying a small patch in one image as *corresponding* to a specific patch in another image is called patch matching. It is a central problem in vision.

Fountain Image 2

Fountain Image 1



Fountain Image 3

Patch matching is used for appearance-based scene or object recognition, 3D scene reconstruction via stereo or multiple views, and image motion estimation. We discuss these tasks later in the course.

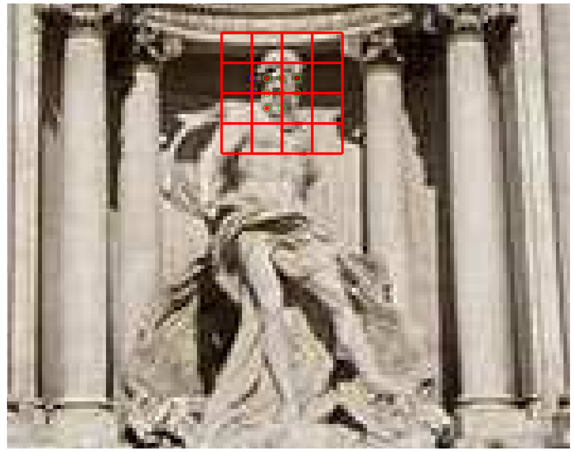
We wish to match a **small patch** rather than a single pixel, since the hope is that such a patch will be **distinctive** enough to be able to find it reliably in other images.

Images are from Steve Seitz's and Rick Szeliski's lecture notes at: http://www.cs.washington.edu/education/courses/cse576/08sp/lectures/features.ppt

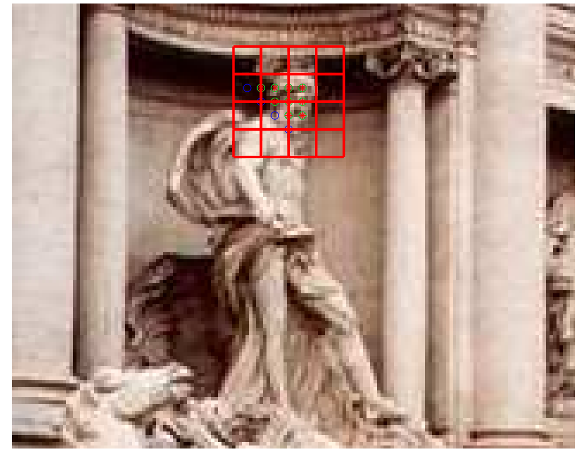
# Definition of Corresponding Points

Detail of Fountain Image 1

Detail of Fountain Image 2



Detail of Fountain Image 3



These regions around the central statue have been cropped from the previous “Fountain” images.

A red grid is centered on a hand selected point in the left image, just under the statue’s nose. The identical sized grids (in pixels) are also drawn approximately over the corresponding image points in the other two images.

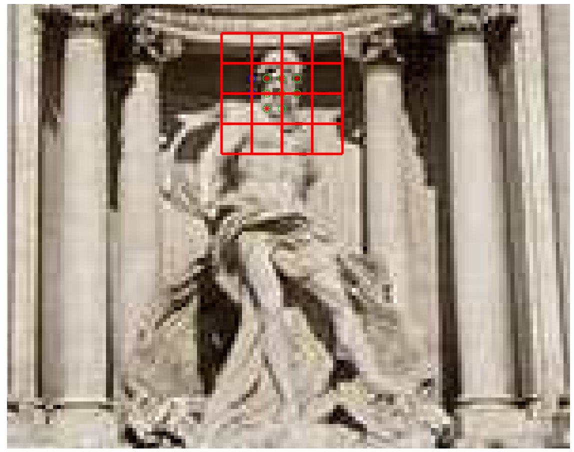
We would consider these grids to be in perfect *correspondence* if the center-most cross-hair in each case marked the image position of the *same scene point*, i.e., just under the statue’s nose.

The correspondence shown above is not perfect. But the errors are less than about 5 or 6 pixels (each cell in the grid is 8 pixels in height and width), despite the large viewpoint change.

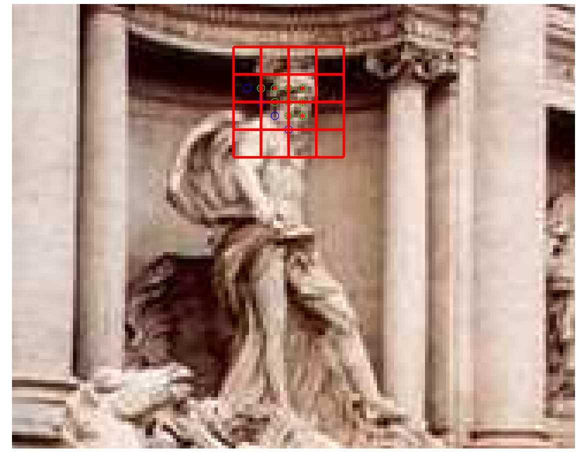
# Difficulties Computing Image Correspondences

Detail of Fountain Image 1

Detail of Fountain Image 2



Detail of Fountain Image 3

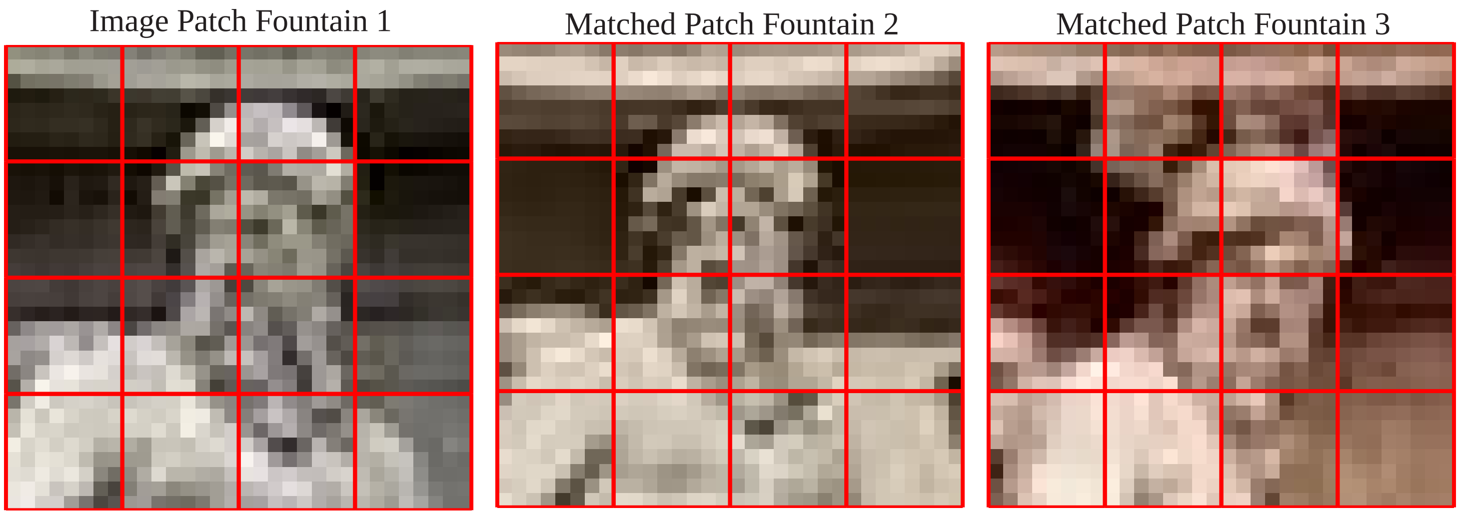


These images also serve to highlight several difficulties in computing corresponding image points:

- The lighting can vary. (Note the left and right images were taken with a mild overcast, so the shadows are softer, while the middle image was taken in direct sunlight.)
- The viewpoint can vary. (For images 1 through 3 the viewpoint moves to the photographer's left.)
- The object's pose might vary. (For example, the statue's head might bend down... perhaps unlikely, but it serves to illustrate my point.)

These factors can all cause an image patch in the neighbourhood of the true corresponding point to change in complex ways from one image to another.

# Difficulties with Image Correspondence: Continued




The local image patch (around the true corresponding point) can also vary due to changes in:

- Image scale. (E.g., if one of these cameras had been zoomed, or had different pixel resolution, or had pixels with different aspect ratios.)
- Image orientation. (E.g., if one of these images was rotated in the image plane.)
- Partial occlusion or background clutter. (Other scene points may appear in the corresponding image patch taken from a different viewpoint. These can provide distracting image texture).

We leave significant changes in image scale or orientation to the next lecture on image landmarks.

# General Strategies for Patch Matching

We have three choices for dealing with the appearance variation of corresponding patches:

1. Ignore the variations. (E.g., image templates, see the next slide.)
2. Model the variations. (E.g., if the image-to-image variations are expected to be small and easily modelled. Such an approach is discussed later in this course for image motion estimation.)
3. Abstract the variations away. (E.g., deliberately design the image feature to be insensitive to small variations, see further below.)

Many practical approaches use mixtures of these choices, that is, ignoring, modelling, and abstracting away different image variations.

# Image Template Matching

A simple idea for matching is to first form a *template*, say $\vec{T}(\vec{x})$, by cropping a region around the target from one image.

Image Template

Given another image $\vec{I}(\vec{x})$, we could try to minimize the sum of squared differences (SSD) error:

$$E(\vec{s}) = \sum_{-K \le x,y \le K} \sum_{c \in \{R,G,B\}} (T_c(\vec{x}) - I_c(\vec{x} + \vec{s}))^2,$$

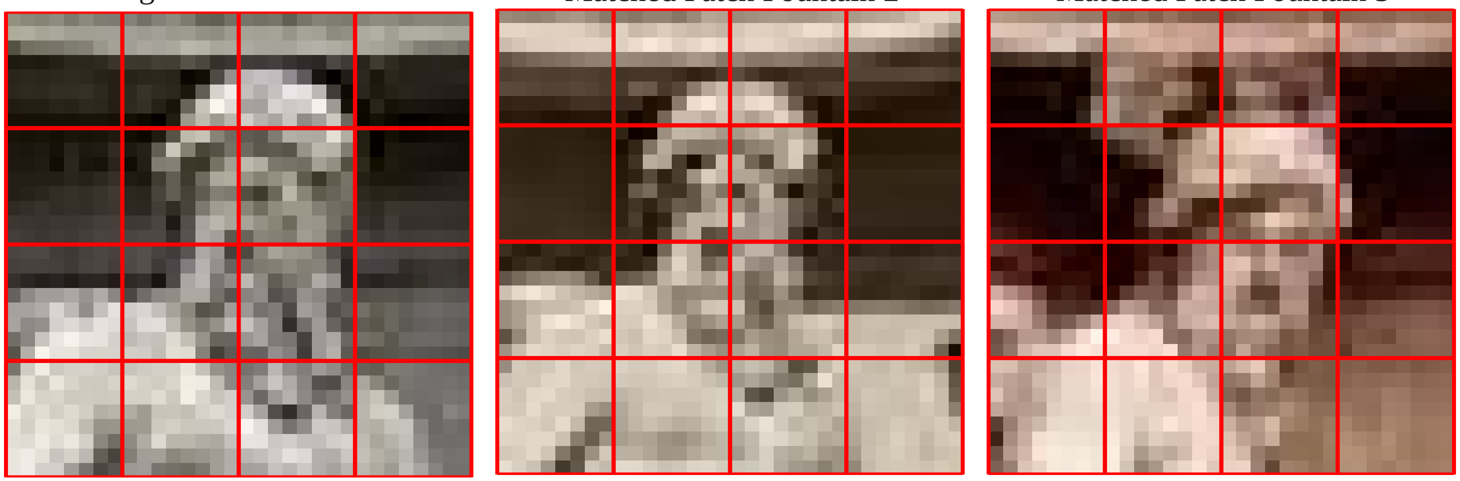
here the $(x, y)$ template coordinates are assumed to be between $-K$ and $K$, and $\vec{s}$ is taken to be any pixel at least $K$ pixels away from the boundaries of $\vec{I}(\vec{x})$.

We discuss this approach for the computation of image motion later in this course. However, for the task being considered here, it is typically overly sensitive to significant viewpoint and lighting changes. (But see Barnes et al, 2009 and 2010.)

# Patch Matching under Significant Viewpoint and Lighting Changes

This motivates the construction of an image feature that is relatively insensitive to the typical variations within small corresponding patchs.

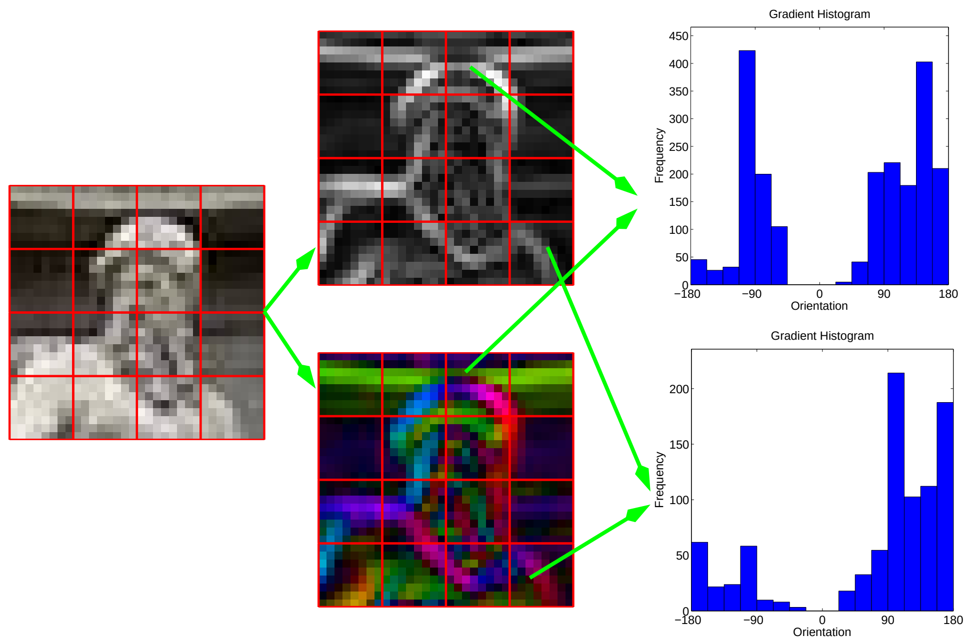
Image Patch Fountain 1 | Matched Patch Fountain 2 | Matched Patch Fountain 3



**Insensitivity to Image Deformations.** Pool a more basic image feature such as image colours or gradients over small spatial cells (e.g., smallest red boxes above), discarding the image position information within each cell. Only this pooled information is stored for each cell. This allows the basic image features to move about within a cell and not change the feature.

**Insensitivity to Lighting.** The pooled information can be normalized to account for simple brightness and/or colour variations.

# Histograms of Gradient (HoG) Features




A histogram of gradient (HoG) descriptor is formed at a specified image location as follows:

- Compute image gradient magnitudes and directions over the whole image, thresholding small gradient magnitudes to zero.
- Center the cell grid on an image location.
- For each cell, form an orientation histogram by quantizing the gradient directions and, for each such orientation bin, add the (thresholded) gradient magnitudes.
- Stack the histograms into one vector of length [(number of orientation bins) $\times$ (number of cells)].
- The resulting HoG vector is normalized (often simply to unit length).

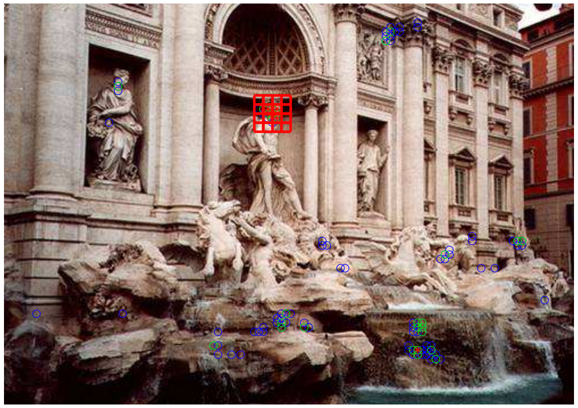
# Patch Matching using HoG

As a demonstration, a pixel is hand selected in the left image, and the HoG response $\vec{h}_1$ is computed.

Fountain Image 1

Fountain Image 2

Fountain Image 3



Given another image (e.g., Fountain 2 or 3), the HoG responses can be computed on a fine grid of pixels (we used a stride of $4$ pixels). Summed area tables (aka, integral images), or running sums (Szeliski, p.120-121) can be used for the efficient computation of the histograms. The distance (often simply Euclidean) between a pixel's HoG response and the original $\vec{h}_1$ is computed.

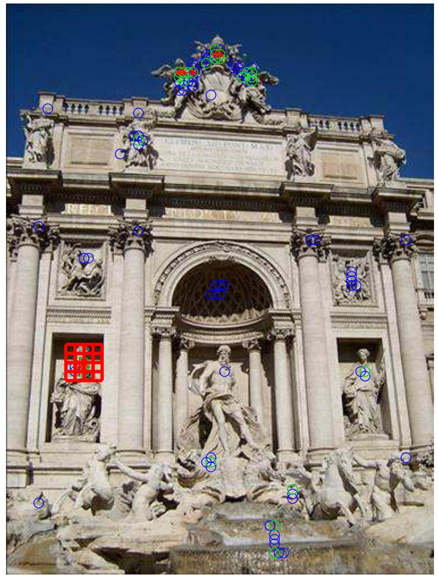
The blue, green, and red markers in the right two images denote pixels with the smallest 100, 30, and 10 distances, respectively. The red grid is centered at the pixel with the smallest distance to $\vec{h}_1$.

These same results are also displayed in the previous figures of the statue's head.

# Patch Matching using HoG: Example 2

Fountain Image 1

Fountain Image 2



Fountain Image 3

Again, the blue, green, and red markers denote pixels with the smallest 100, 30, and 10 distances, respectively. The red grid is drawn centered at the smallest response over all sampled locations.

In each of the above cases the HoG responses were computed at about 11,000 image locations. Of all these HoG vectors, the one nearest the initial HoG descriptor $\vec{h}_1$ serves to identify a patch that is in reasonable correspondence.

# Patch Matching using HoG: Examples 3 and 4

Fountain Image 1

Fountain Image 2

Fountain Image 3

Fountain Image 1

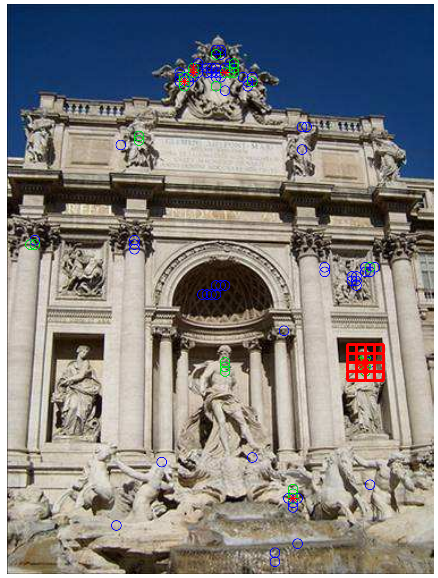
Fountain Image 2

Fountain Image 3

# Patch Matching using HoG: Example 5

Fountain Image 1

Fountain Image 2



Fountain Image 3

In the right figure, the best matching point does not correspond to the point selected in Fountain 1. This is due to the larger change in viewpoint and scale for the corresponding image patch.

# Summary: Image Patch Matching

We defined corresponding points, as the image projection of the same scene point across several images.

Image patches centered at corresponding points can be expected to vary substantially, due to changes in viewpoint, lighting, object pose, and so on.

The HoG feature is an image feature that abstracts away (to some extent) both spatial deviations and variations in lighting. We illustrated that HoG can lead to an effective patch matching tool (see also Dalal and Triggs, 2005).

The current approach is, however, limited to matching image patches that are at similar image orientations and scales.

One way to address this limitation is to use image landmarks which explicitly provide image orientation and scale (next lecture).

Alternatively, in tracking or pose fitting applications, we can explicity infer the image orientation and scale of the corresponding patch.

# References

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman, PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing, *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28, 3, 2009.

Connelly Barnes, Eli Shechtman, Dan B Goldman and Adam Finkelstein, The Generalized PatchMatch Correspondence Algorithm, *European Conference on Computer Vision*, Sept. 2010.

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, *International Conference on Computer Vision and Pattern Recognition*, Vol.2, June 2005, pp.886-893.