Imperial College London

Feature Description II

Dr Wenjia Bai

Department of Computing & Brain Sciences

Previously

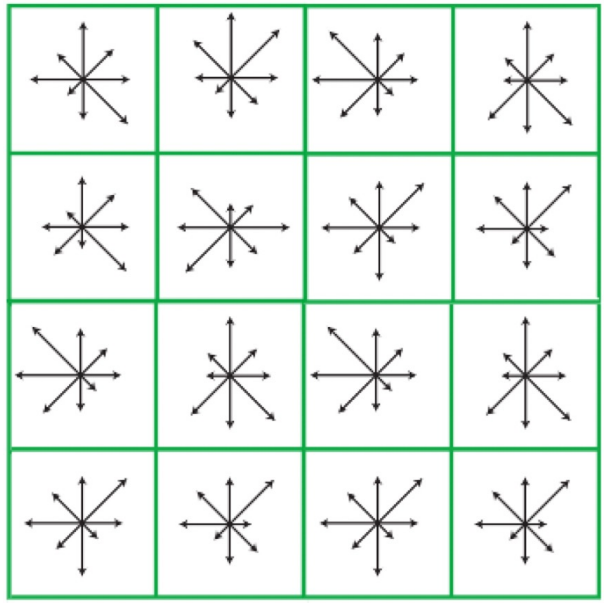
- We went through the details of the SIFT algorithm.
- SIFT is a classical algorithm in which you will understand the main concepts about feature descriptors.
- It is used for image matching and other applications.

In this lecture

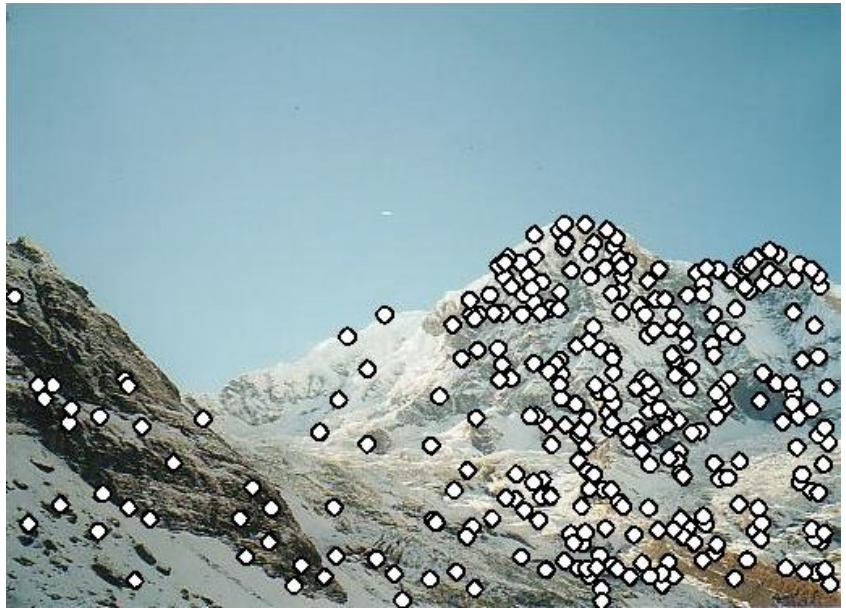
- We will describe more feature descriptors that are relevant to SIFT.
  - SURF
  - BRIEF
- We will then extend the idea of feature description for an interest point to feature description for an image.

SIFT

- The SIFT descriptor at each interest point is formed by concatenating the histograms of gradient orientations for 16 subregions.
- However, calculating the gradient magnitudes and orientations can be slow, especially if you want to achieve real-time performance (e.g. image matching and stitching on camera) and you were still working in early 2000.



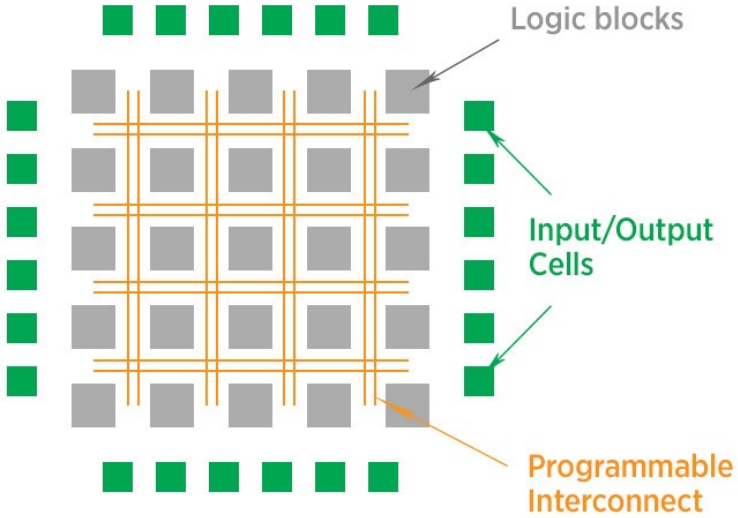
SIFT descriptor



How do we improve the speed for image matching?

Faster feature description

- You may have different ideas.
  - Use faster hardware
  - Optimise software implementation
  - Develop a new algorithm
- For example, one solution is to implement SIFT on field programmable gate array (FPGA), which is much faster than CPU.

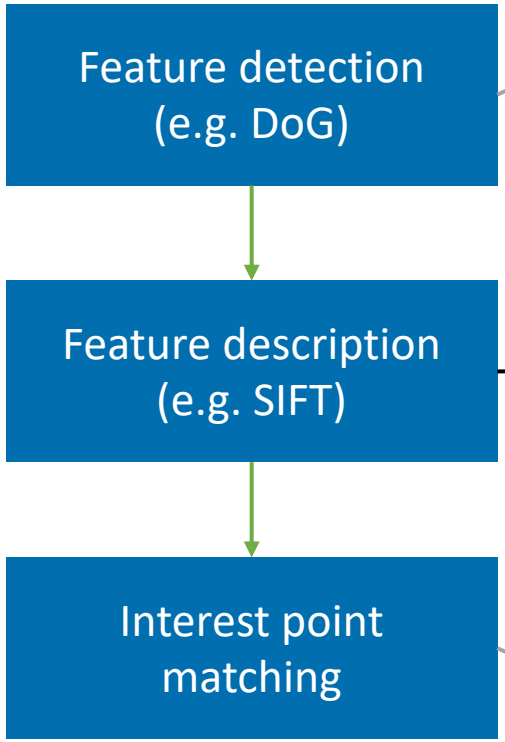


FPGA is an integrated circuit that contains many logic blocks, configured to perform complex functions in a very fast way.

Se et al. Vision based modeling and localization for planetary exploration rovers. IAC, 2004.

Faster feature description

- Another way is to improve the software or algorithm.
  - Decompose a pipeline into several steps
  - Reconsider and evaluate each step
  - How can we improve this step?



Pipeline for image matching

Faster feature detection:

- Separable filtering
- Downsample image
- Crop Gaussian kernel
- …

Faster feature description:

- Can we evaluate gradient orientation faster?
- Can we describe the local content in a different way?
- …

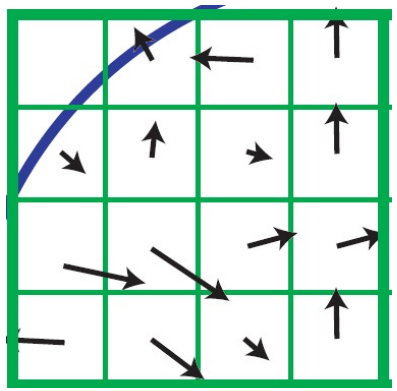
Faster matching:

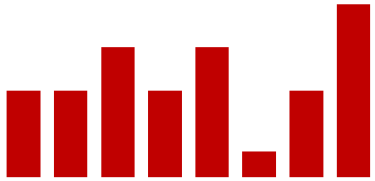
- Approximate nearest neighbour
- Use a lower-dimensional feature vector
- …

Speeded-Up Robust Features (SURF)

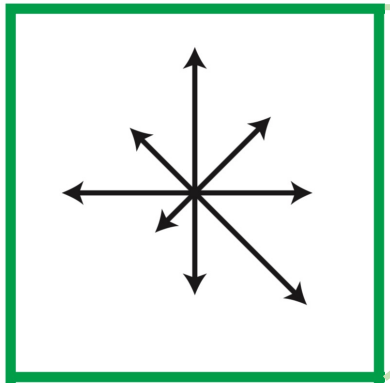
- SIFT uses histograms of gradient orientations for describing local features.
- To accelerate computation, SURF was proposed, which only calculates the gradients along horizontal and vertical directions by using the Haar wavelets.

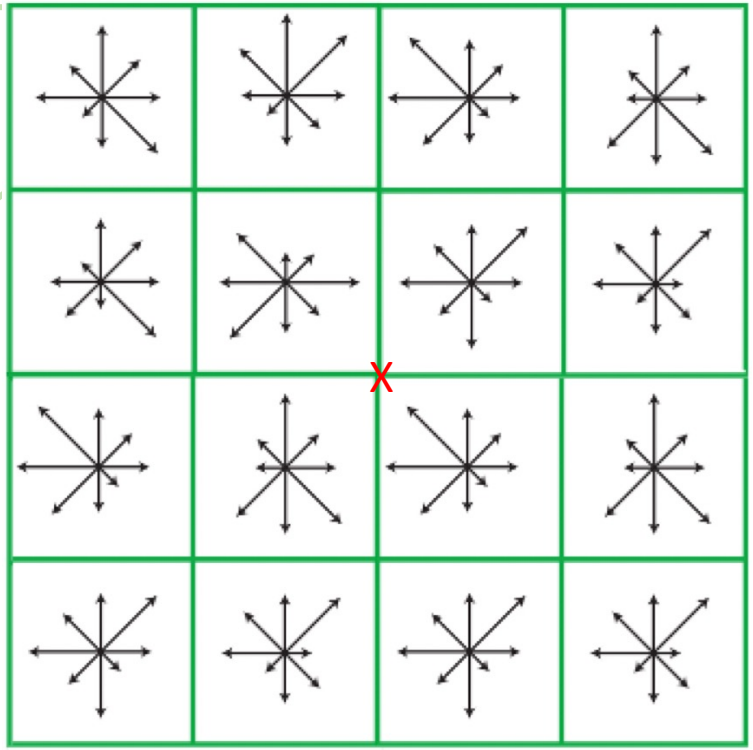
H. Bay et al. SURF: Speeded Up Robust Features. ECCV, 2006.



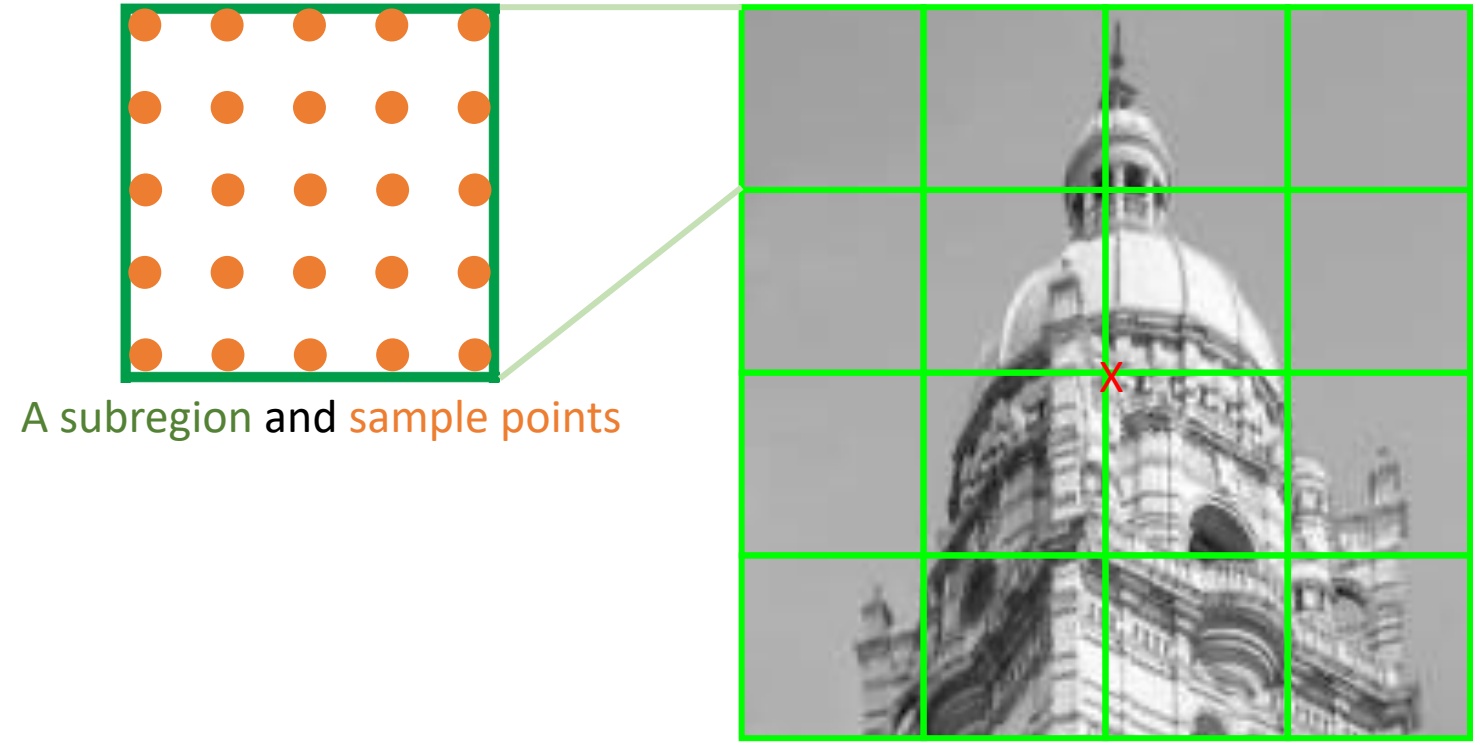


Orientation histogram
(8 bins)





SIFT descriptor calculates a histogram of gradient orientations, for each of the 16 subregions centred at an interest point X.



The idea of SURF is that we do not need to calculate gradients for arbitrary orientations. Let’s only look at horizontal and vertical directions.

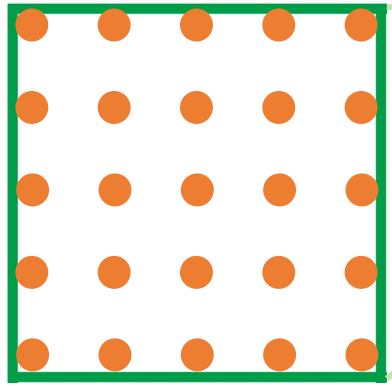
- SURF applies very simple filters d_x and d_y onto the sample points to calculate gradients along x and y.
- They are called Haar wavelets.

d_x

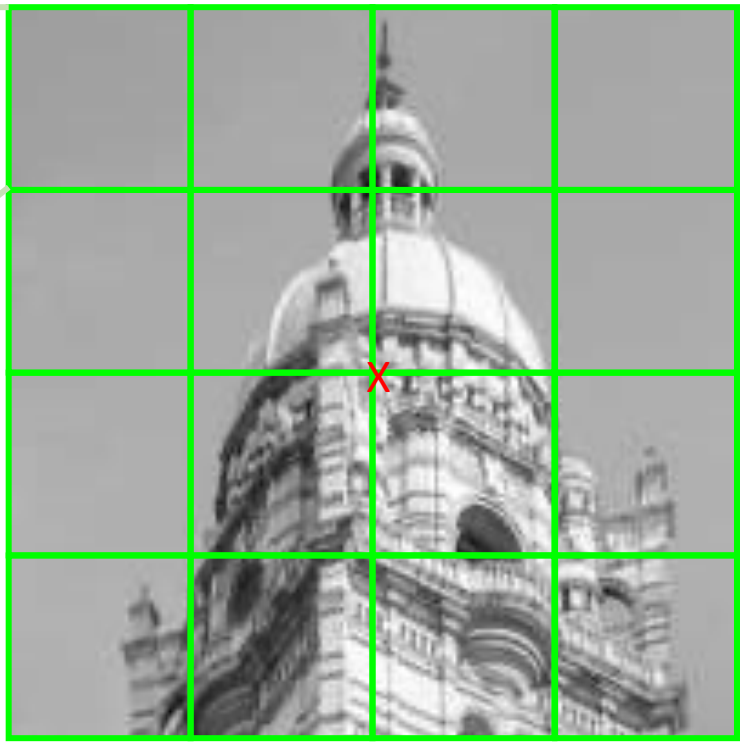
-	+
1	1

d_y

-1
+1



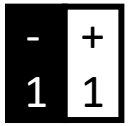
A subregion and sample points



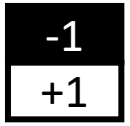
The idea of SURF is that we do not need to calculate gradients for arbitrary orientations. Let's only look at horizontal and vertical directions.

SURF

A subregion and sample points



d_x



d_y

Haar wavelets, summing up the pixel intensities with weight +1 or -1, so very fast.

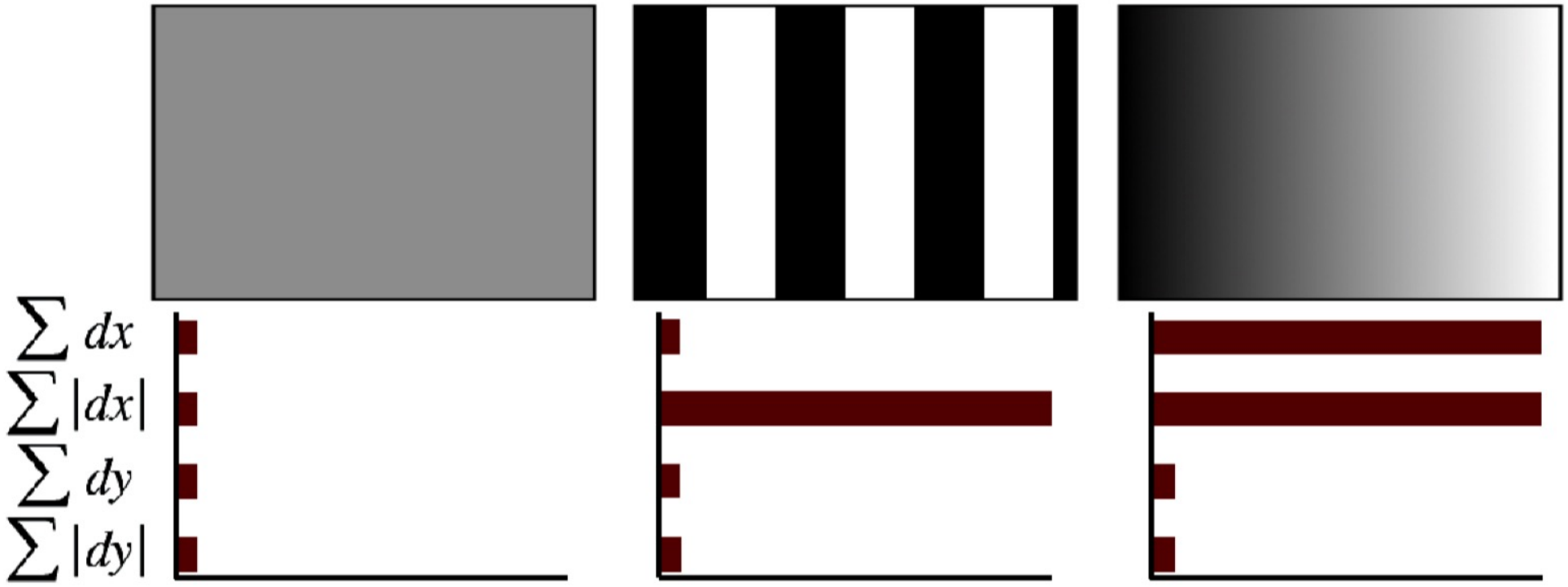
$$(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$$

summing up values — summing up absolute values

For each subregion, sum up the Haar wavelet responses over the sample points. The descriptor for this subregion is defined by 4 elements.

H. Bay et al. SURF: Speeded Up Robust Features. ECCV, 2006.

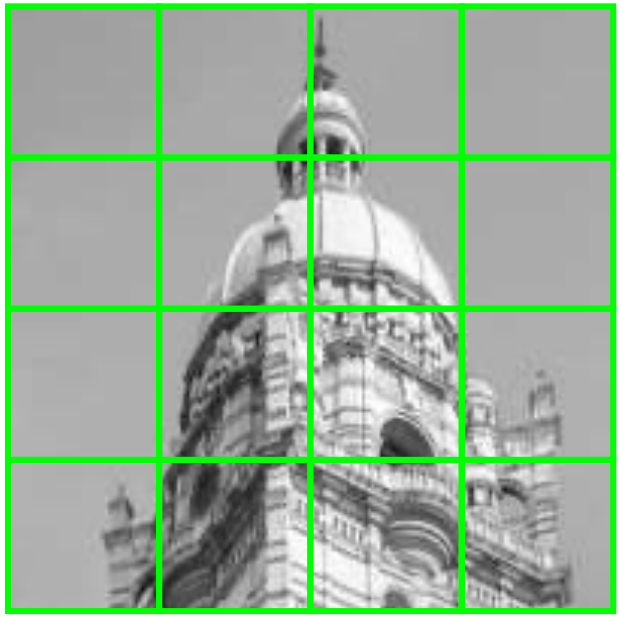
SURF



This seemingly simple descriptor $(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ can represent the intensity patterns of a subregion. Left: a homogeneous region, all values are low; Middle: zebra pattern, $\sum |d_x|$ is high but other values are low; Right: gradually increasing intensities, $\sum d_x$ and $\sum |d_x|$ are high.

SURF

- Dimension of SURF descriptor
  - 4 x 4 subregions
  - 4 elements per subregion
  - 16 subregions x 4 elements = 64
- SURF is ~5 times faster than SIFT, due to the use of simple Haar wavelets for feature computation.



SURF

- After describing features for each interest point, we can perform matching for points from image A and image B.
- If we know the ground truth (transformation between A and B, we can evaluate the matching accuracy.
- SURF performs as well as SIFT in matching [1].

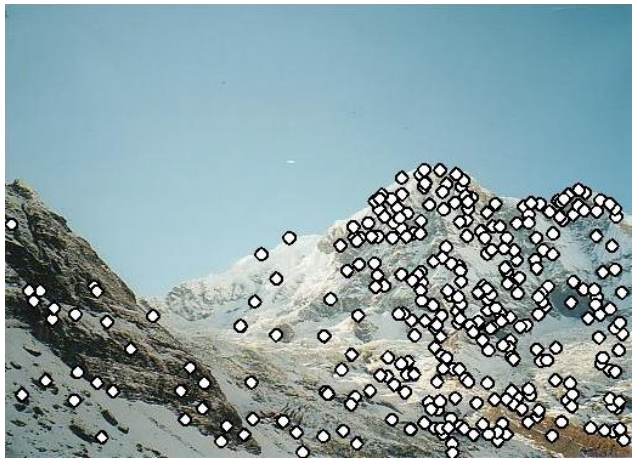


Image A

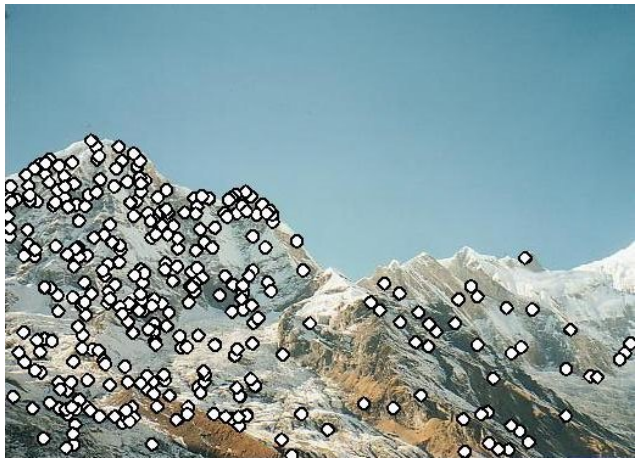


Image B

[1] H. Bay et al. SURF: Speeded Up Robust Features. ECCV, 2006.

Even faster?

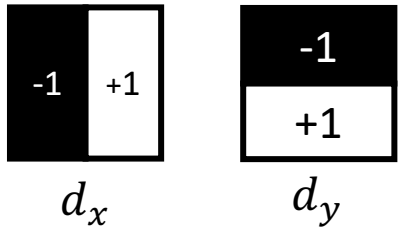
- Feature descriptors
  - SIFT: 16 x 8 = 128-D vector
  - SURF: 16 x 4 = 64-D vector
  - Each dimension is a floating-point number or 4 bytes.
  - To compare two feature vectors, we calculate their Euclidean distance.
- Can we make feature description and matching even faster?
- Similar motivation as we develop SURF
  - We can deploy feature description and image matching algorithms faster on devices, such as camera, mobile phones, robots etc.

Even faster?

- How to further shorten the descriptor?
  - What about quantize or even binarize the floating-point number (4 bytes)?
  - Quantization means converting a continuous number into a discrete number (e.g. 0 to 255, which means 8 bits).
  - Binarization means converting into a binary number (0 or 1, which means 2 bits).
- How to compare two feature vectors faster?
  - Distances other than Euclidean?
- ...

Binary Robust Independent Elementary Features (BRIEF)

- When we use Haar wavelets in SURF, we compare a local region to another and calculate the difference, which is a floating point number.

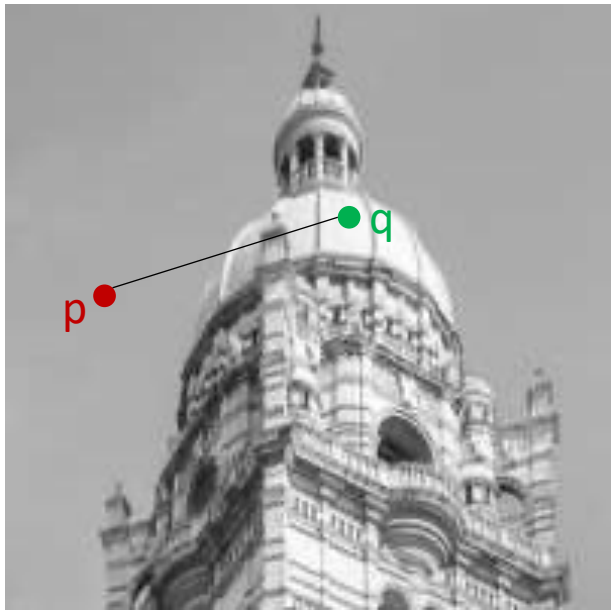


Haar wavelets used in SURF

- In BRIEF, we compare a point p to another point q and get a binary value as output.

$$\tau(p,q) = \begin{cases} 1, & \text{if } I(p) < I(q) \\ 0, & \text{otherwise} \end{cases}$$

- How do we sample these point pairs (p, q)?

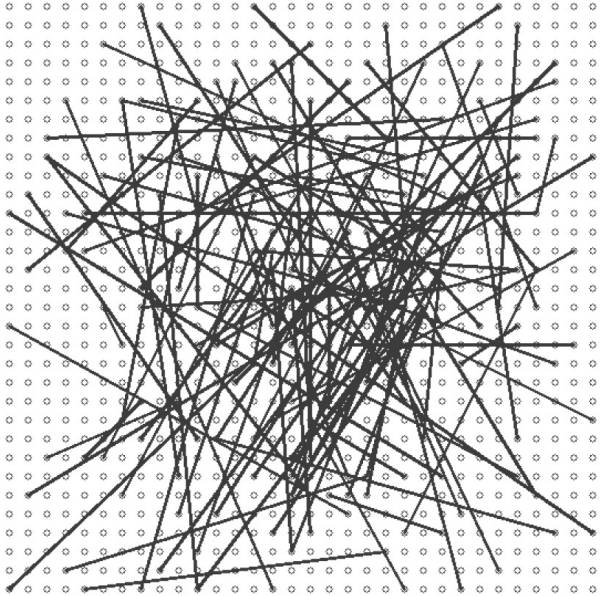


Binary test in BRIEF

M. Calonder. BRIEF: Binary Robust Independent Elementary Features. ECCV 2010.

BRIEF

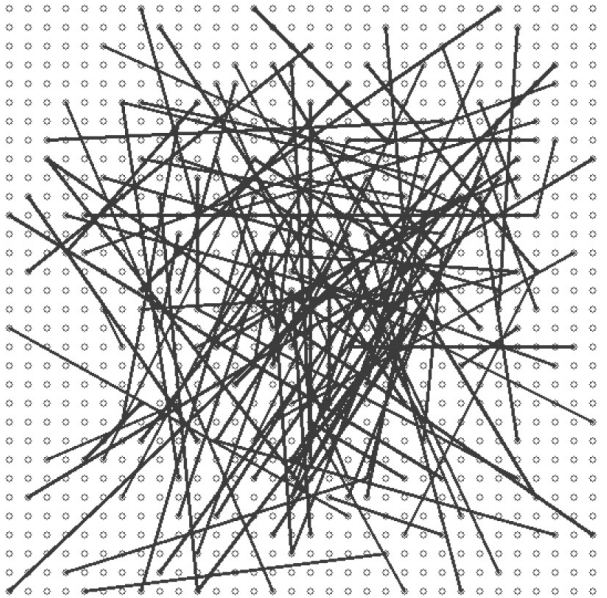
- We randomly sample n_d pairs of points for binary tests.
  - The random pattern is determined only once. Then we always apply the same pattern to all interest points.
- If $n_d = 256$, we perform 256 tests and each test gives us 1 bit (0 or 1).
- The BRIEF descriptor is described as a n_d-dimensional bitstring.
  - For example, [1, 0, 0, 0, 1, …, 0, 0, 1].



Random sampling pattern for binary tests

BRIEF

- If $n_d = 256$, the BRIEF descriptor is 256-bit long or 32 bytes.
- This is shorter than SIFT (512 bytes) or SURF (256 bytes).
- The computation is much faster.
  - We only compare two numbers, without calculating the gradient orientation (SIFT) or intensity difference (SURF).
  - When we compare two BRIEF descriptors, it is also faster as we do not need to calculate the Euclidean distance.



Random sampling pattern for binary tests

Fast to compute the BRIEF descriptor

For example, to perform 8 binary tests and get one byte of descriptor:

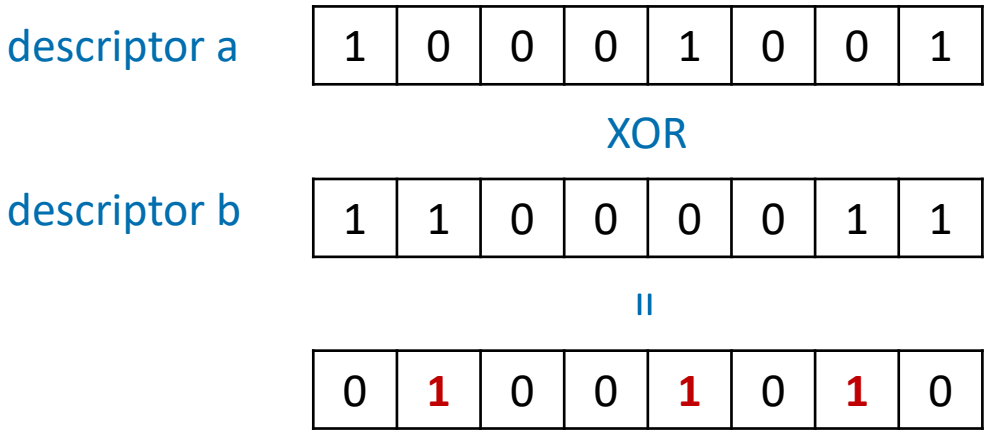
$$\text{descriptor} = ((I(p_1) < I(q_1)) << 7) + ((I(p_2) < I(q_2)) << 6) + ((I(p_3) < I(q_3)) << 5)$$
$$+ ((I(p_4) < I(q_4)) << 4) + ((I(p_5) < I(q_5)) << 3) + ((I(p_6) < I(q_6)) << 2)$$
$$+ ((I(p_7) < I(q_7)) << 1) + ((I(p_8) < I(q_8)) << 0)$$

p and q are pre-defined sampling points.

“<< n” means shifting the bit by n places.

Fast to compare two BRIEF descriptors

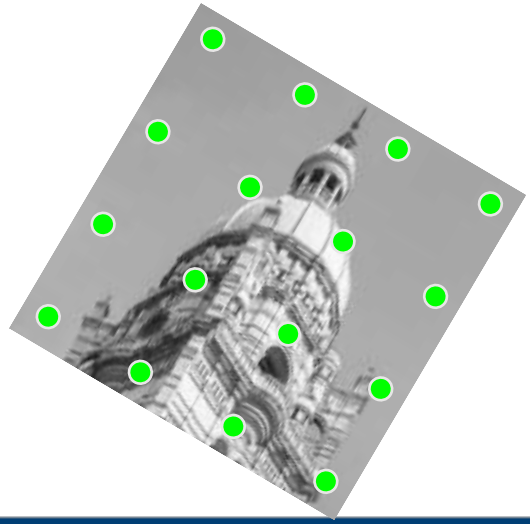
- When we perform image matching and comparing two BRIEF descriptors, we can measure their difference using the Hamming distance.
  - Hamming distance can be computed very efficiently with a bitwise XOR operation followed by a bit count.



By counting the bit of **1**, we get the Hamming distance = 3.

BRIEF

- BRIEF ignores rotation- and scale-invariance.
- It assumes that the images are taken from a moving camera that only involves translation, and it does not account for rotation or scaling.

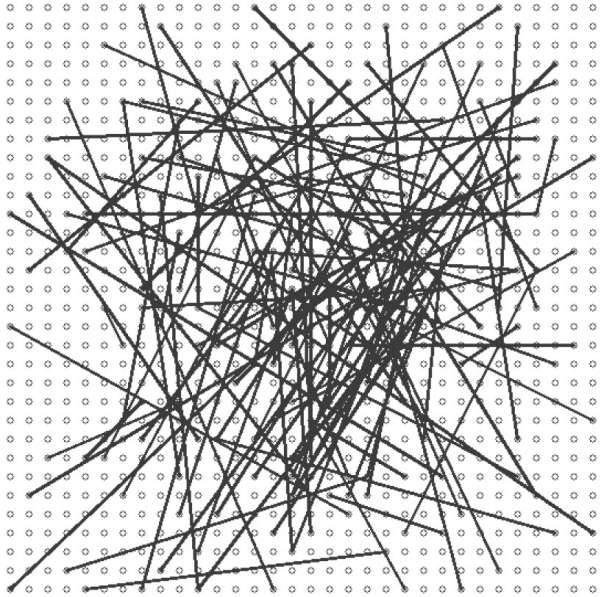


BRIEF only accounts for translation

It does not account for rotation or scaling.

BRIEF

- BRIEF ignores rotation- and scale-invariance.
- It assumes that the images are taken from a moving camera that only involves translation, and it does not account for rotation or scaling.
- BRIEF uses a upright window of fixed size (e.g. 48 x 48) for sampling.
- This further reduces computation.



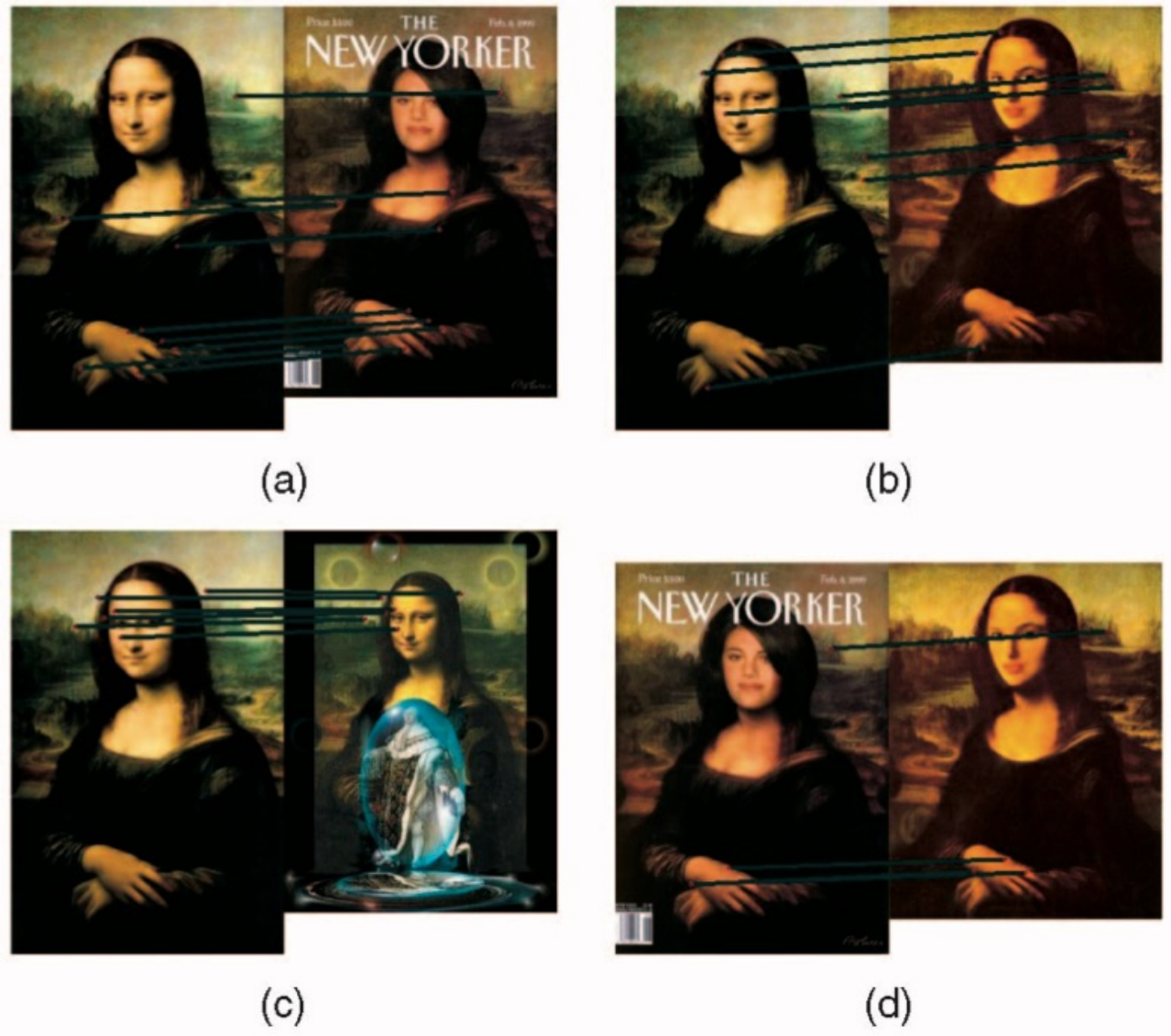
BRIEF

- By combining all these components, BRIEF achieves ~40-fold speed-up over SURF [1].
  - Binary tests
  - Hamming distance
  - Ignore orientations and scales
- Recall that SURF is already ~5 times faster than SIFT [2].
- So BRIEF is ~200 times faster than SIFT.
- If there is no rotation or scaling involved, BRIEF achieves a performance comparable to SURF in terms of interest point matching accuracy [1].

[1] M. Calonder. BRIEF: Binary Robust Independent Elementary Features. ECCV 2010.
[2] H. Bay et al. SURF: Speeded Up Robust Features. ECCV, 2006.

Feature descriptors

- We first detect interest points, then describe them.
  - SIFT
  - SURF
  - BRIEF
- Feature descriptors can be used for image matching or other image analysis tasks.
  - Object recognition by matching
  - Measuring image similarity
  - …



If two images are similar, there should be more interest points that are matched to each other.
Image similarity can be defined by the number of matched points.

Y. Jing et al. VisualRank: Applying Page Rank to Large-Scale Image Search. TPAMI, 2008.

A similar graph can be constructed, which can be used for image search.

Y. Jing et al. VisualRank: Applying Page Rank to Large-Scale Image Search. TPAMI, 2008.

Feature descriptors

- These feature descriptors are defined for an interest point for describing local content, centred at this point.
- Can we extend the idea to describe the feature of a whole image or a image region?
  - Yes.

Can we describe the feature of human for an image or a region?

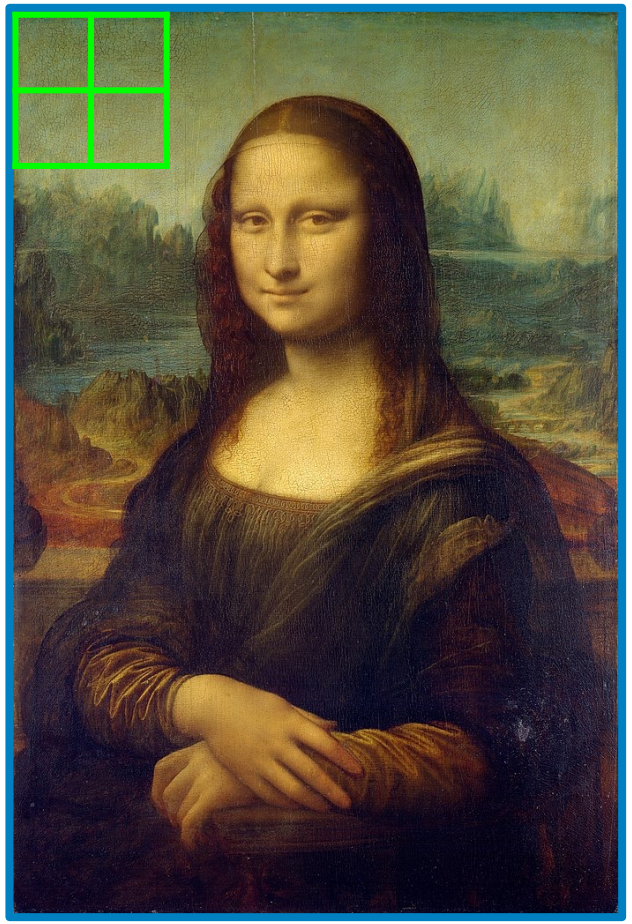
Histograms of Oriented Gradient (HOG)

- HOG is similar to SIFT. They both use gradient orientation histograms for feature description.
- The difference is that HOG describes features for a large image region, instead of just around a point.
- The idea of HOG is that it divides a large region into a dense grid of cells, describes each cell, then concatenate these local descriptions to form a global description.

HOG

- Suppose we are to describe the feature for Mona-Lisa.
- We can divide the image into equally spaced cells.
  - Each cell contains 8 x 8 pixels.
  - 4 cells form a block.
- The orientation histograms of this block describes the top-left corner of this image.

Calculate gradient orientation histograms for this block.



128x64 pixel image.

N. Dalal et al. Histograms of Oriented Gradients for Human Detection, CVPR 2005.

HOG

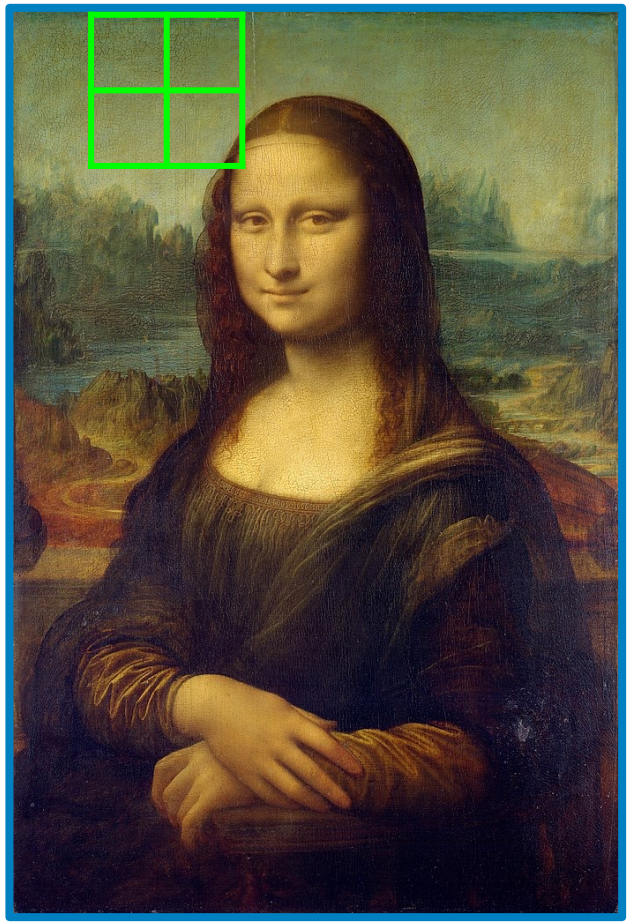
- We move to the next block and similarly describe the content there using orientation histograms.
- There is an overlap between blocks.
- For each block, the descriptor vector v (concatenation of 4 histograms) is normalised,

$$v_{norm} = \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}}$$

locally normalised descriptor

a small value

Calculate the histograms at the next block.



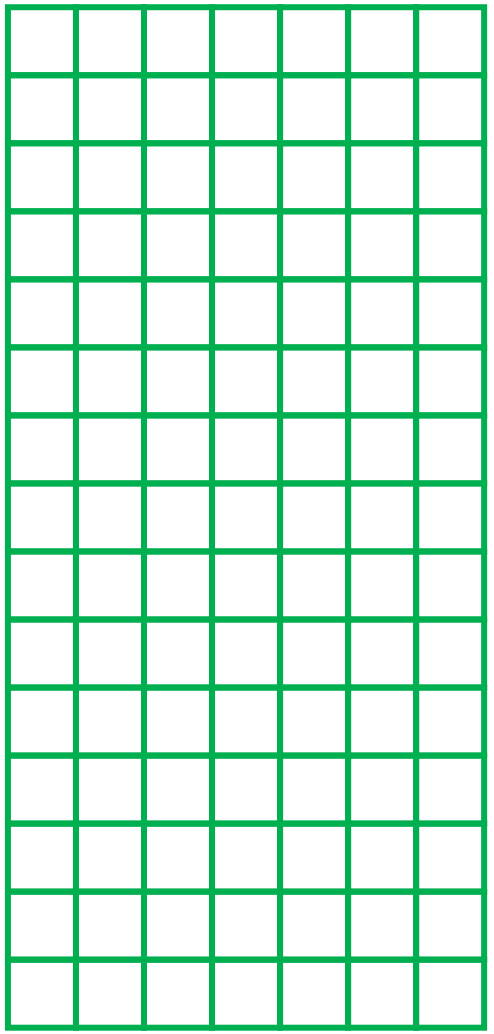
128x64 pixel image.

N. Dalal et al. Histograms of Oriented Gradients for Human Detection, CVPR 2005.

HOG

- The HOG descriptor is formed by concatenating the normalised local descriptors for all blocks.
- In this way, we can describe a full image or a large image region.
- We use a dense grid to cover and describe the image. In contrary, SIFT only looks at a single point.

The descriptors for all blocks in this image.



N. Dalal et al. Histograms of Oriented Gradients for Human Detection, CVPR 2005.

Feature descriptors

- What can we do if we can describe a full image or a region?
  - We can perform image classification based on image features.
  - We can detect whether a region contains human or not.
  - We can retrieve similar images by their features.
  - …

Image classification

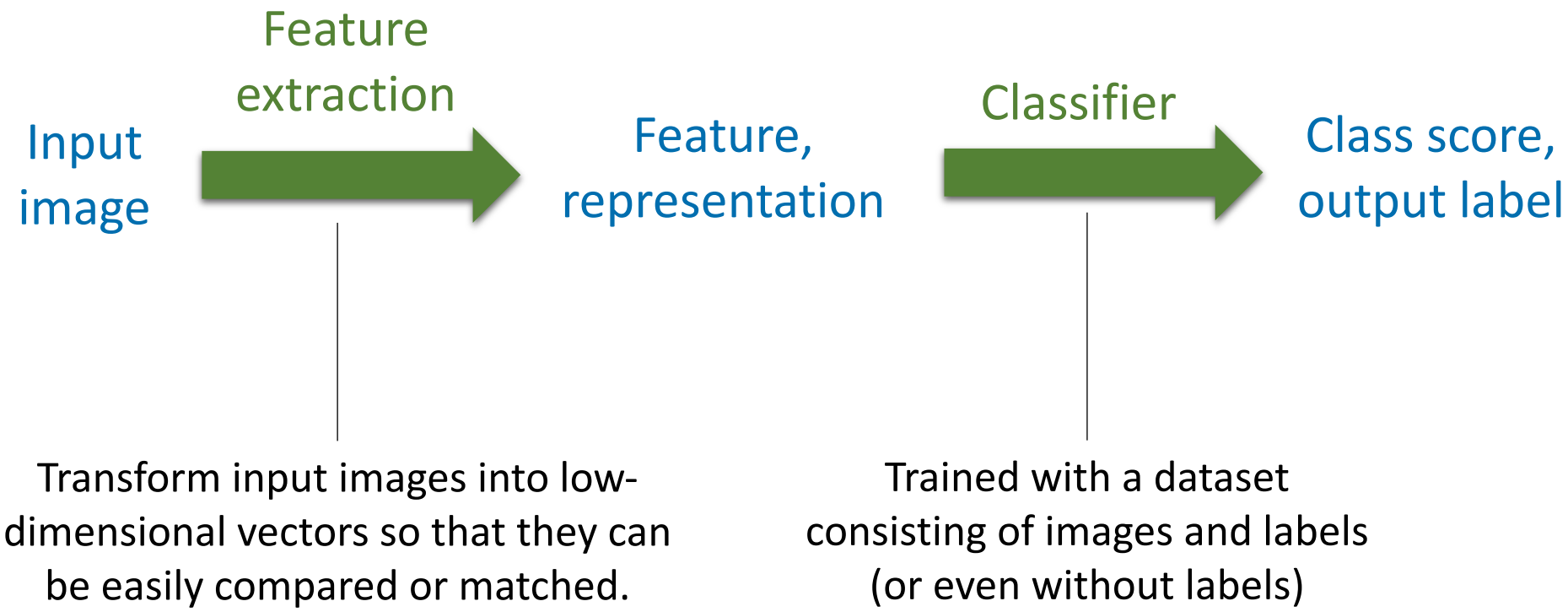


Image classification

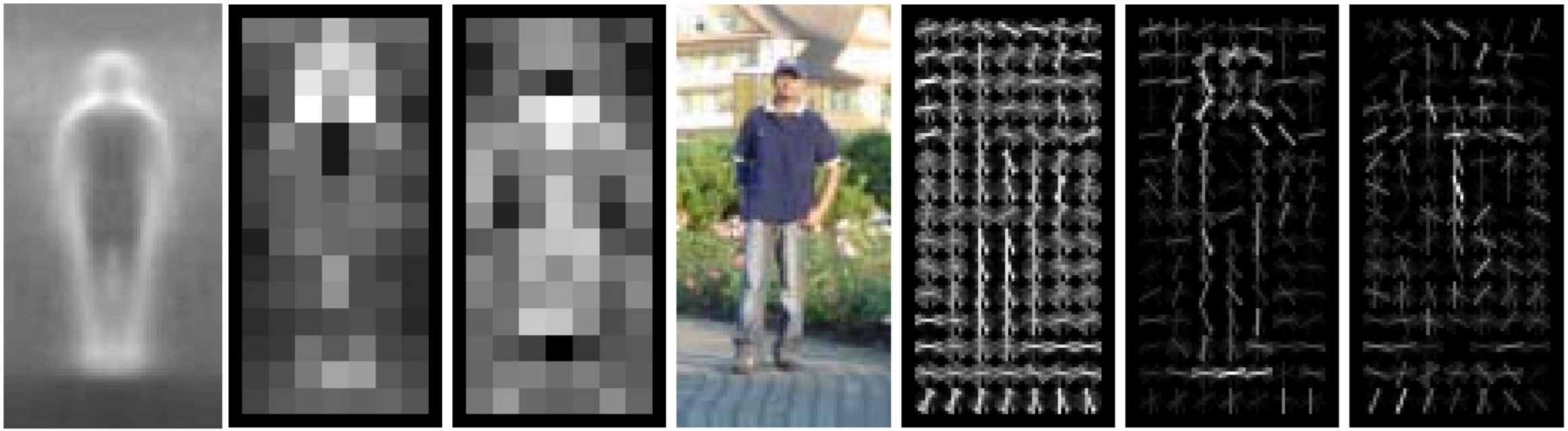


Figure 6. Our HOG detectors cue mainly on silhouette contours (especially the head, shoulders and feet). The most active blocks are centred on the image background just *outside* the contour. (a) The average gradient image over the training examples. (b) Each "pixel" shows the maximum positive SVM weight in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) A test image. (e) It's computed R-HOG descriptor. (f,g) The R-HOG descriptor weighted by respectively the positive and the negative SVM weights.

N. Dalal et al. Histograms of Oriented Gradients for Human Detection, CVPR 2005.

HOG feature for human detection

Credit: https://medium.com/@madhawavidanapathirana

Summary

- We start from feature descriptors that extract local features near an interest point,
  - SIFT
  - SURF
  - BRIEF
- Then move onto a feature descriptor that describes an image,
  - HOG
- This will start our chapter for image classification.

References

- Sec. 4.1.2 Feature descriptors. Richard Szeliski, Computer Vision: Algorithms and Applications (http://szeliski.org/Book).