Feature Extraction

PART 3:
THE PATCH…

Our goal

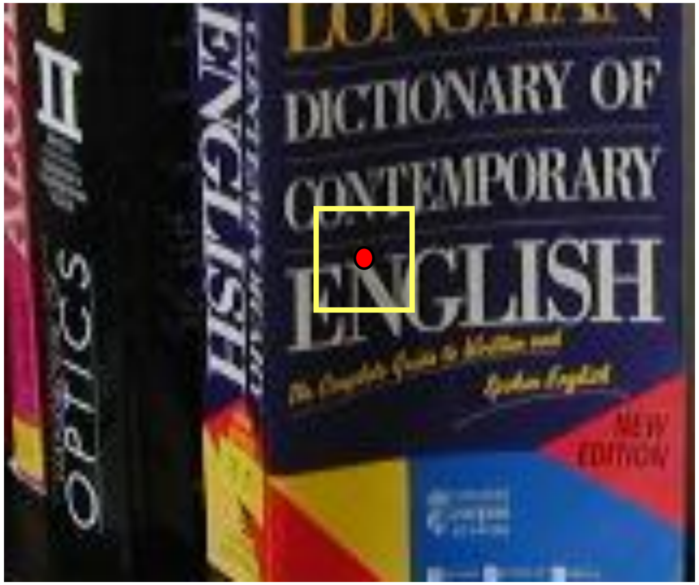
BUT, so far we have glossed over an important issue:

The shape of the patch should change with viewpoint

The need for variable patch shape

Computer Vision

The need for variable patch shape

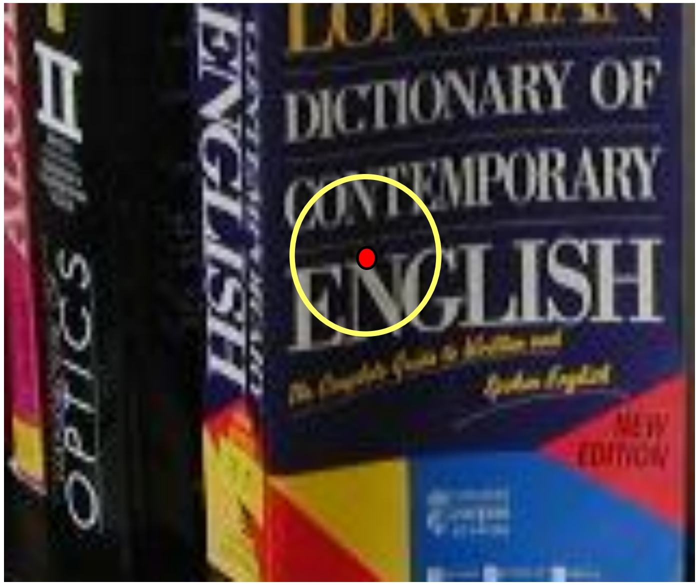


Taking the same square patch around corresponding interest points leads to a very different content of the patches… hence the matching will become hard.

The need for variable patch shape

Replacing the squares by identical circles does not really help much…

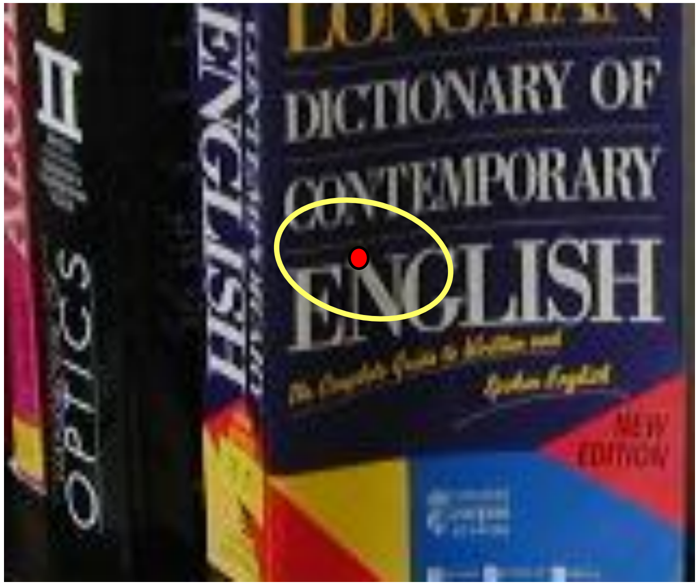
The need for variable patch shape



Allowing the diameters to differ helps somewhat, but contents still quite different (look at the top regions in the circles)

Computer Vision

The need for variable patch shape



Using ellipses works much better: the circle on the left transforms into an ellipse under the affine transf. between the local view change

The need for variable patch shape

The important thing is to achieve such change in patch shape without having to compare the images, i.e. this should happen on the basis of information in one

The need for variable patch shape

An image that we will use as test case…

Note the global perspective/projective distortion!

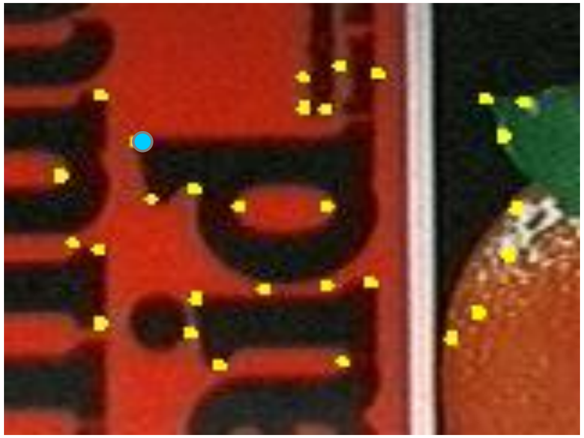
Computer Vision

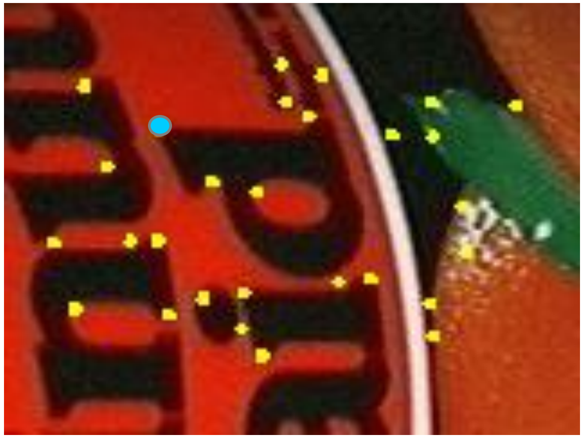
Example: parallelogram next to edge corner

Computer Vision

Example: parallelogram next to edge corner

1. Harris corner detection





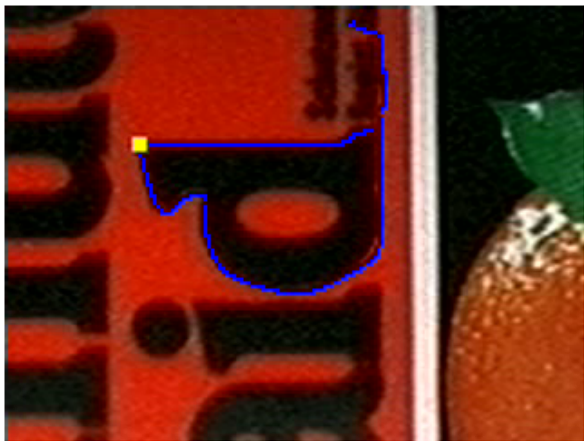
Computer Vision

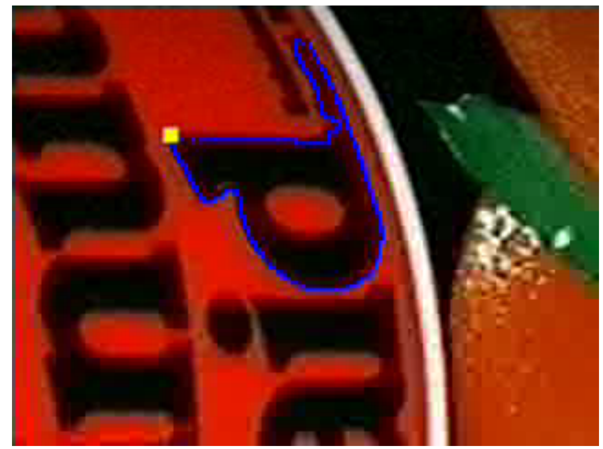
Example: parallelogram next to edge corner

2. (Canny) edge detection

Example: parallelogram next to edge corner

3. Evaluation relative affine invariant parameter along two edges



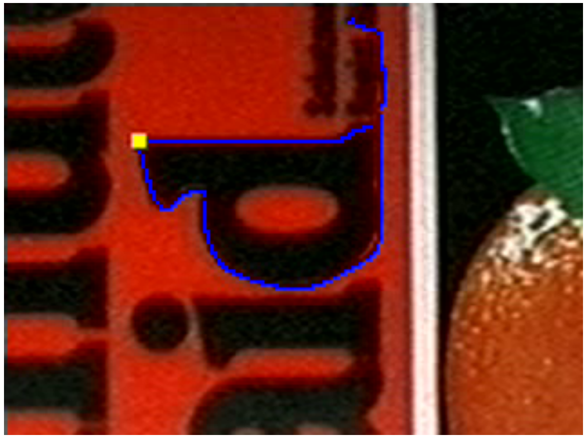


$$l_i = \int abs(|\, p_i^{(1)}(s_i) \quad p - p_i(s_i)\,|)ds_i$$

Letting 2 points evolve away from the corner in opposite directions, such that the above expression is the same for both, yields in the 2 images to corresponding parallelograms

Example: parallelogram next to edge corner

4. Construct 1-dimensional family of parallelogram shaped regions

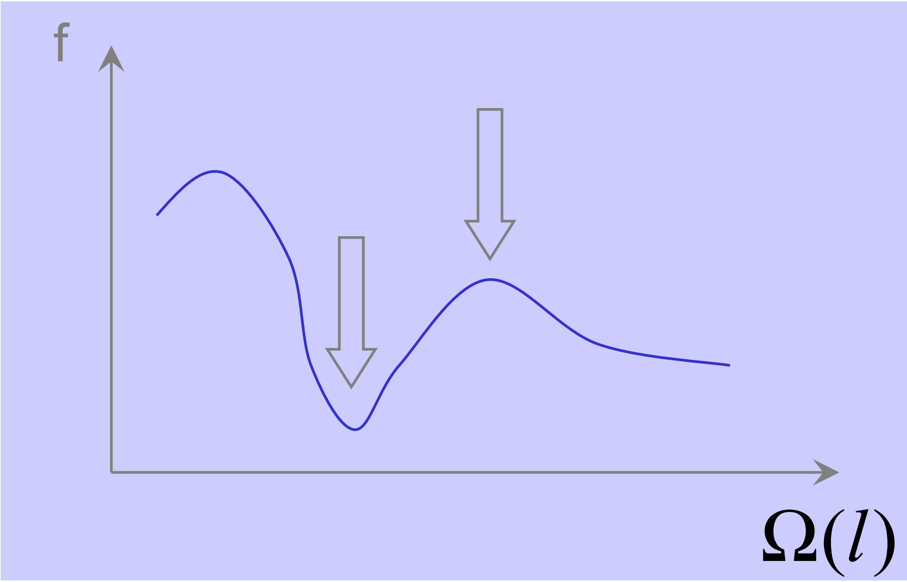


Computer Vision

Example: parallelogram next to edge corner

5. Select parallelograms based on invariant extrema of function

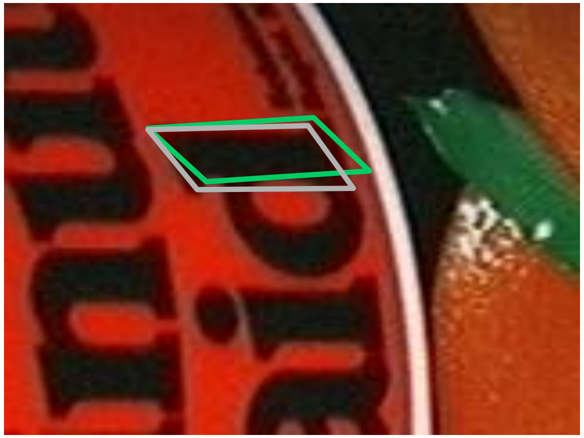
For instance:
extrema of average value of a color band within the patch



Computer
Vision

Example: parallelogram next to edge corner

5. Select parallelograms based on local extrema of invariant function



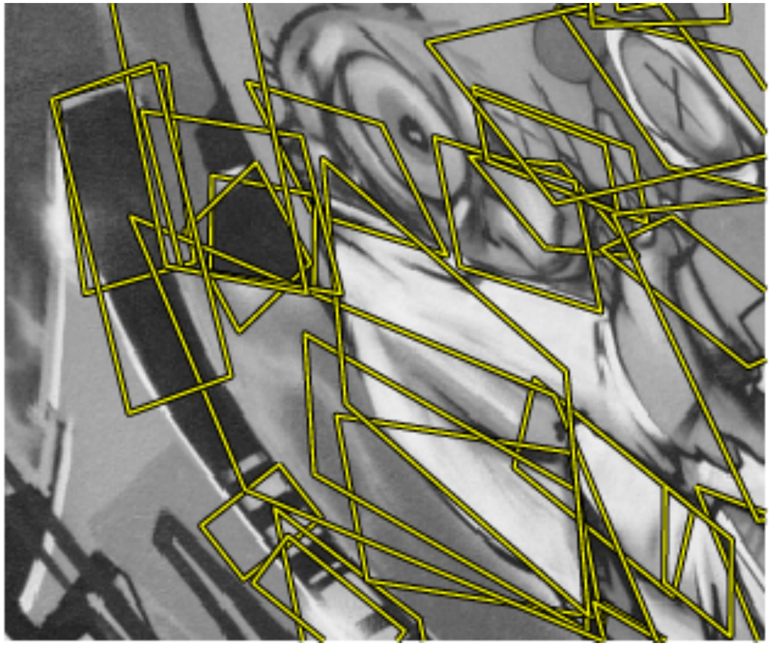
Example: parallelogram next to edge corner

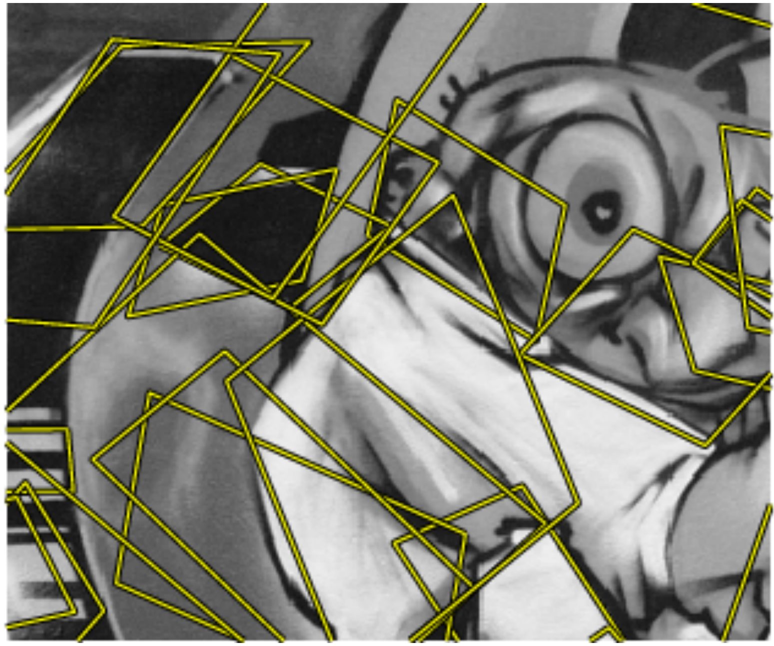
Augmented Reality where an interest point's patch is filled with the texture `BIWI'. The stability shows how well the patch adapts to the viewpoint

PART 4: EXAMPLES

Computer Vision

Example 1: edge corners + affine moments





Example 1: edge corners + affine moments

6. Describe the pattern within the parallelogram with affine invariant moments

Geometric/photometric moment invariants based on generalised colour moments:

$$M_{p,q}^{a,b,c} = \int x^p y^q \, r^a(x,y) \, g^b(x,y) \, b^c(x,y) \, dxdy$$

M_{pq}^{abc} *are not invariant themselves, need to be combined*

Computer Vision

Example 1: edge corners + affine moments

6. Describe the pattern within the parallelogram with affine invariant moments

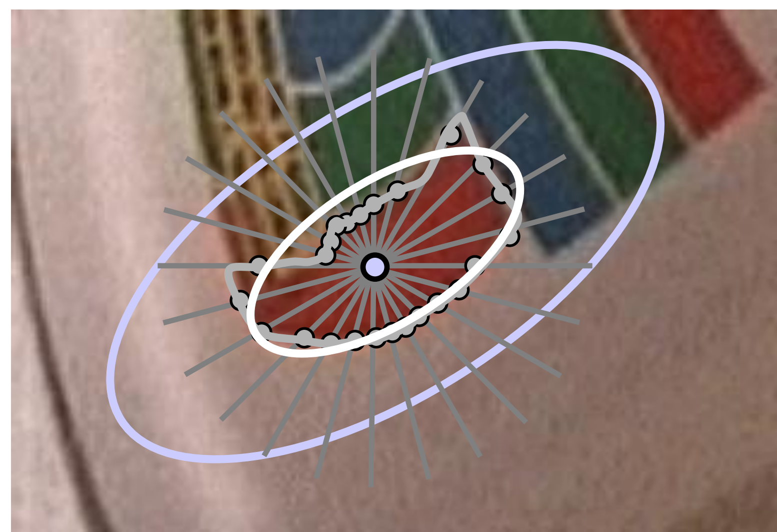
Example moment invariant from only 2 color bands:

$$D_{02} = \frac{\left[M_{00}^{11} M_{00}^{00} - M_{00}^{10} M_{00}^{01}\right]^2}{\left[M_{00}^{20} M_{00}^{00} - (M_{00}^{10})^2\right] \left[M_{00}^{02} M_{00}^{00} - (M_{00}^{01})^2\right]}$$

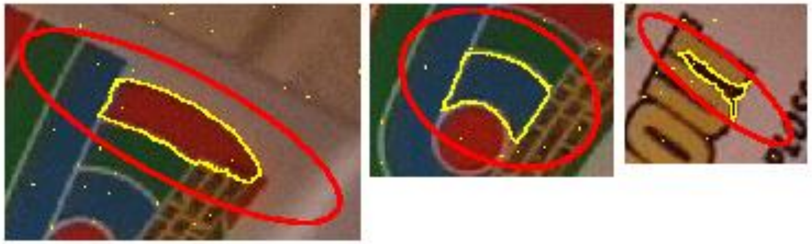
Ex. 2: intensity extrema + affine moments

1. Search intensity extrema
2. Observe intensity profile along rays
3. Search maximum of invariant function f(t) along each ray
4. Connect local maxima
5. Fit ellipse
6. Double ellipse size
7. Describe elliptical patch with moment invariants

$$f(t) = \frac{abs(I_0 - I)}{\max\left(\frac{\int abs(I_0 - I)dt}{t}, d\right)}$$

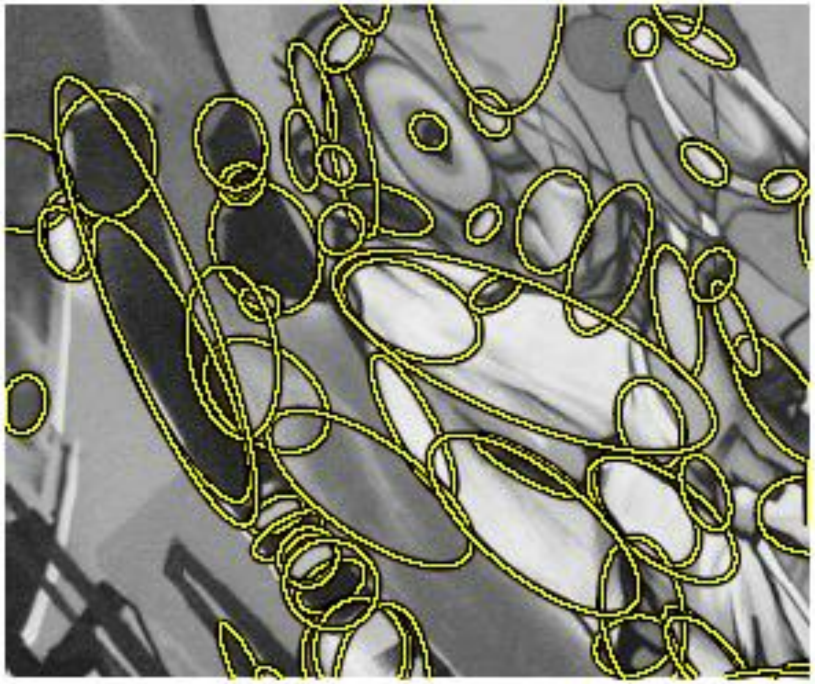


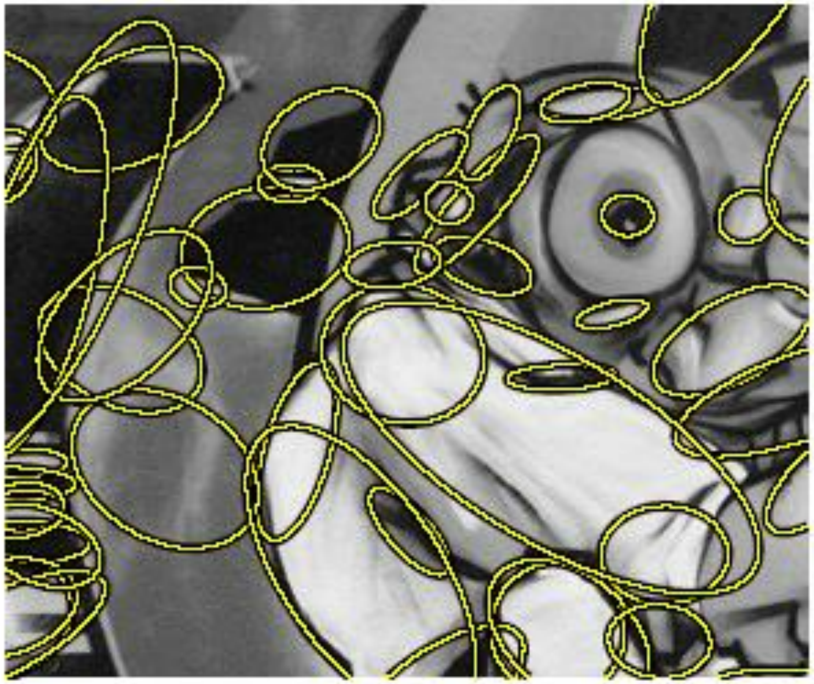
Ex. 2: intensity extrema + affine moments



Computer Vision

Ex. 2: intensity extrema + affine moments



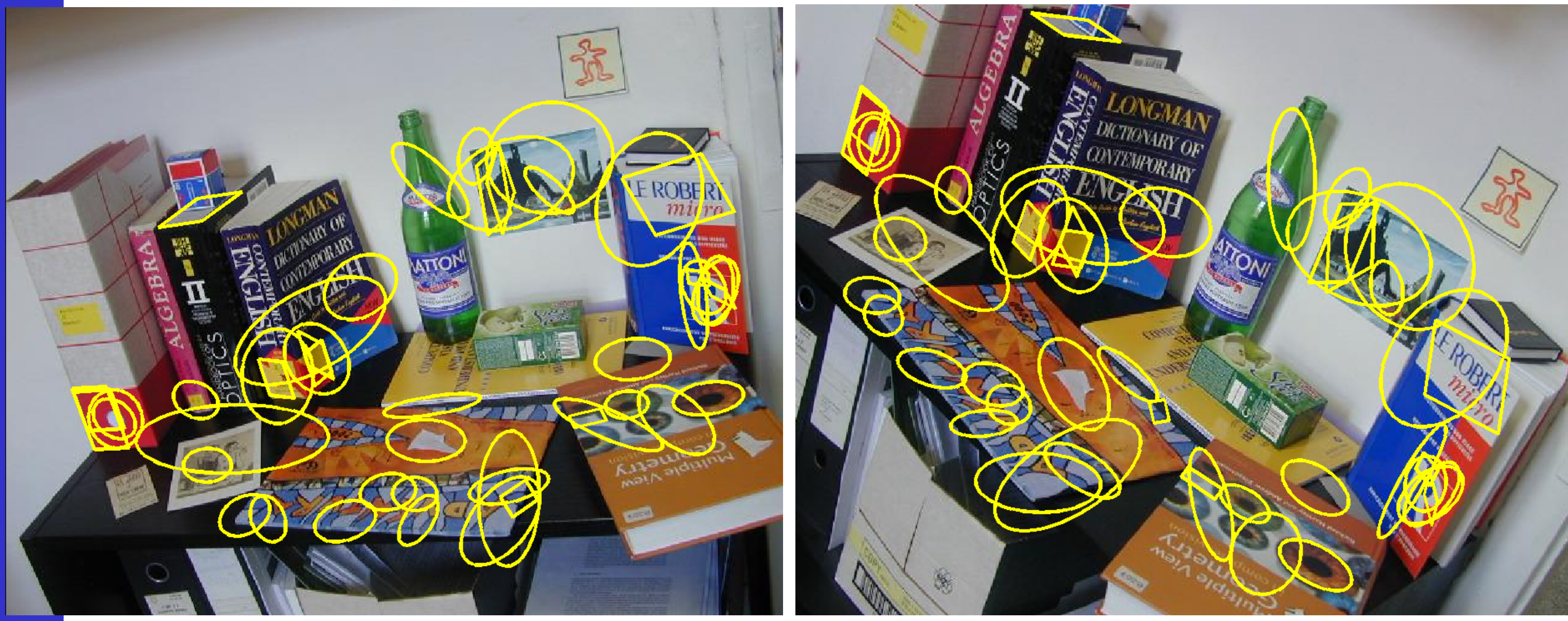


Remark

In practice different types of interest points are often combined

Wide baseline stereo matching example.

based on ex.1 and ex.2 interest points



MSER interest points

MSER = Maximally Stable Extremal Regions

- Similar to the Intensity-Based Regions we just saw
- Came later, but is more often used
- Start with intensity extremum
- Then move intensity threshold away from its value and watch the super/sub-threshold region grow
- Take regions at thresholds where the growth is slowest (happens when region is bounded by strong edges)

Computer Vision

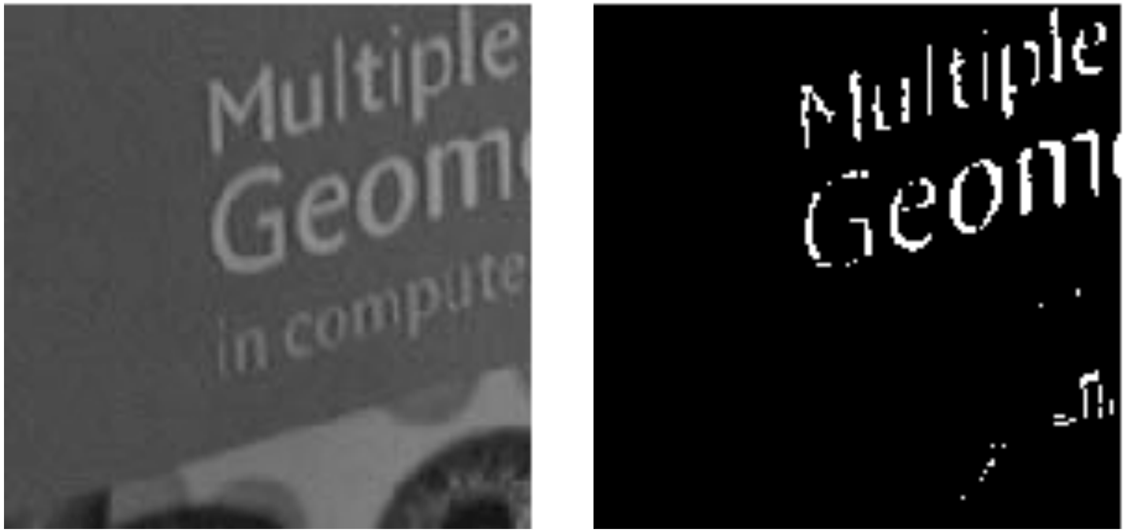
MSER

Computer Vision

MSER

Computer Vision

MSER



MSER

Computer Vision

MSER

Computer Vision

MSER

MSER

Computer Vision

MSER

Computer Vision

MSER

MSER

Computer Vision

MSER

Computer Vision

MSER

Computer Vision

MSER

Computer Vision

MSER

MSER

MSER

MSER

- *Extremal region*: region such that

$$\forall p \in Q, \forall q \in \delta Q : \begin{matrix} I(p) > I(q) \\ I(p) < I(q) \end{matrix}$$

- Order regions, following increasing or decreasing threshold

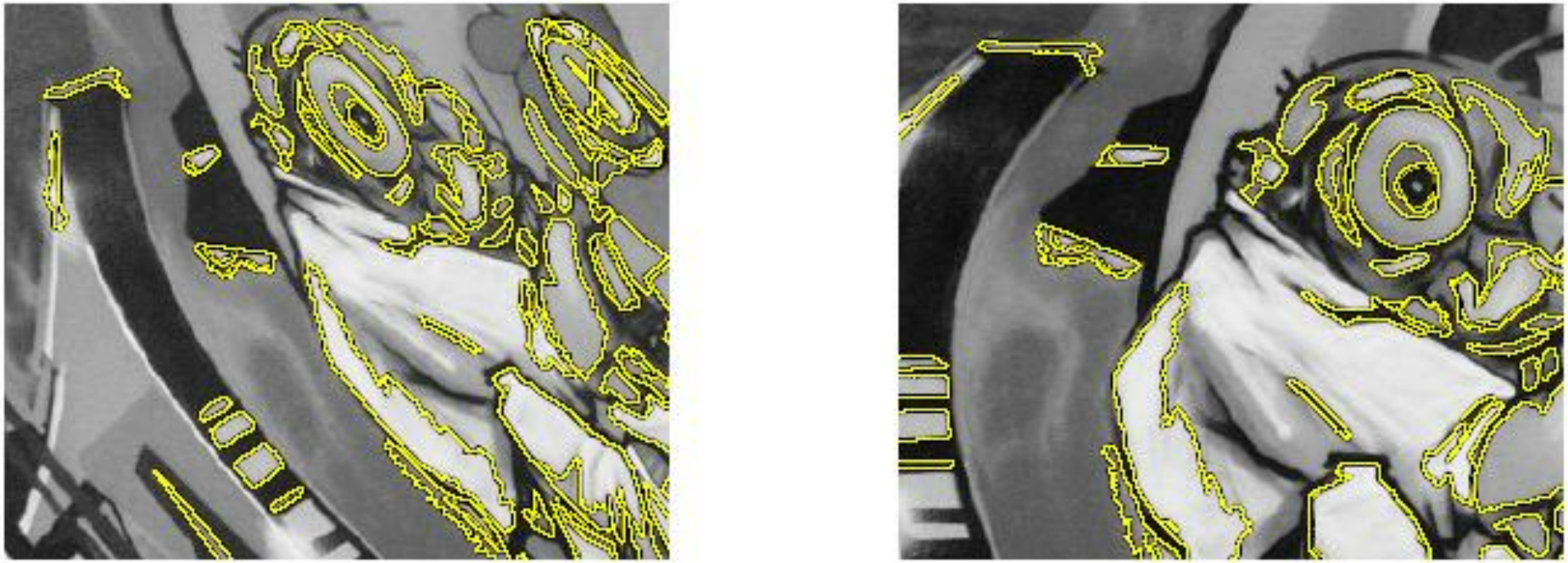
$$Q_1 \subset \ldots \subset Q_i \subset Q_{i+1} \subset \ldots Q_n$$

- *Maximally Stable Extremal Region*: local minimum of

$$q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}| / Q_i$$

Computer Vision

MSER



(a) **MSER**

Example 3: SIFT

SIFT = Scale-Invariant Feature Transform

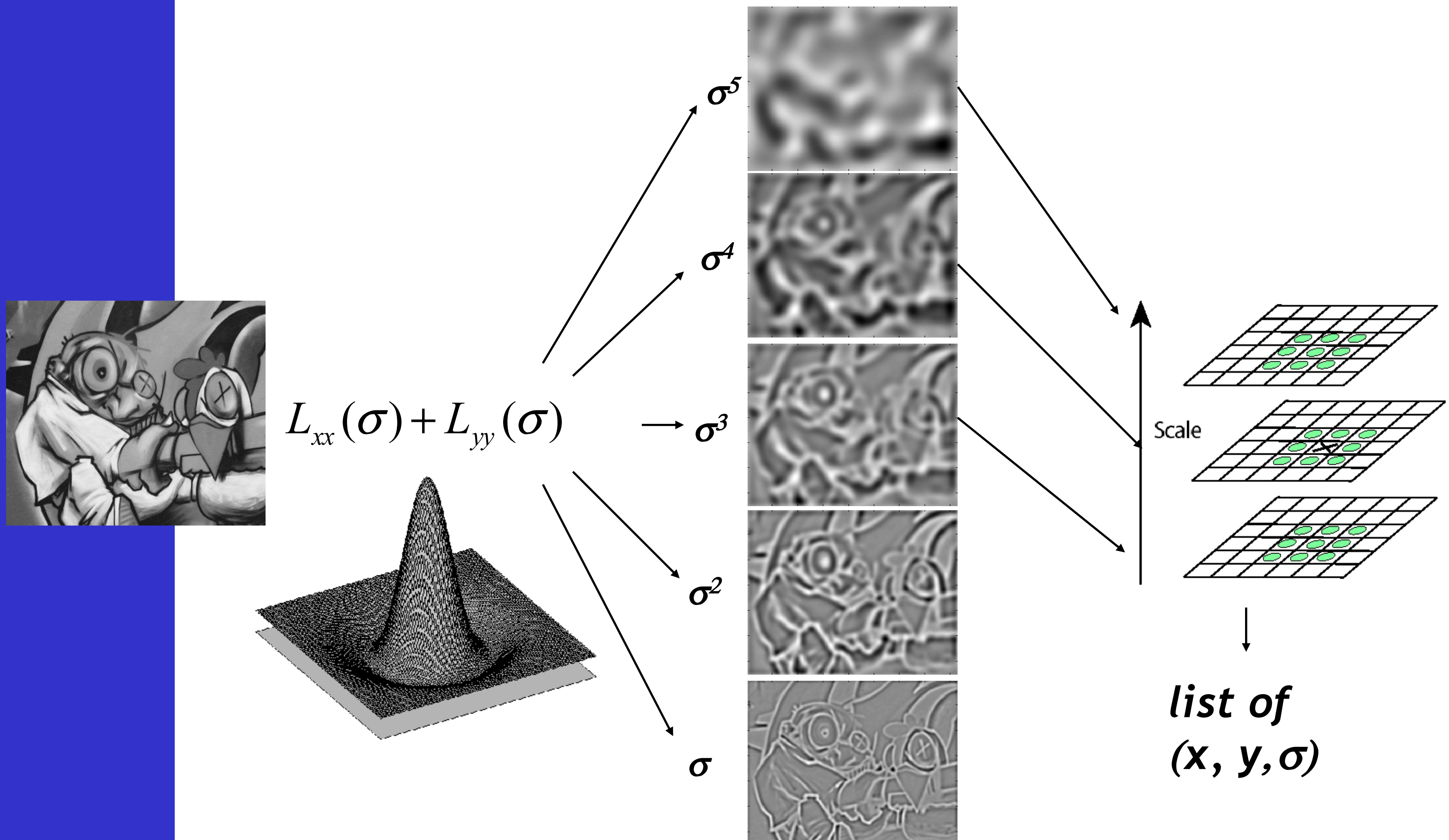
SIFT, developed by David Lowe (Un. British Columbia, Canada), is a carefully crafted interest point detector + descriptor, based on intensity gradients (cf. our comment on photometric invariance) and invariants under similarities, not affine like so far

Our summary is a simplified account!

Computer Vision

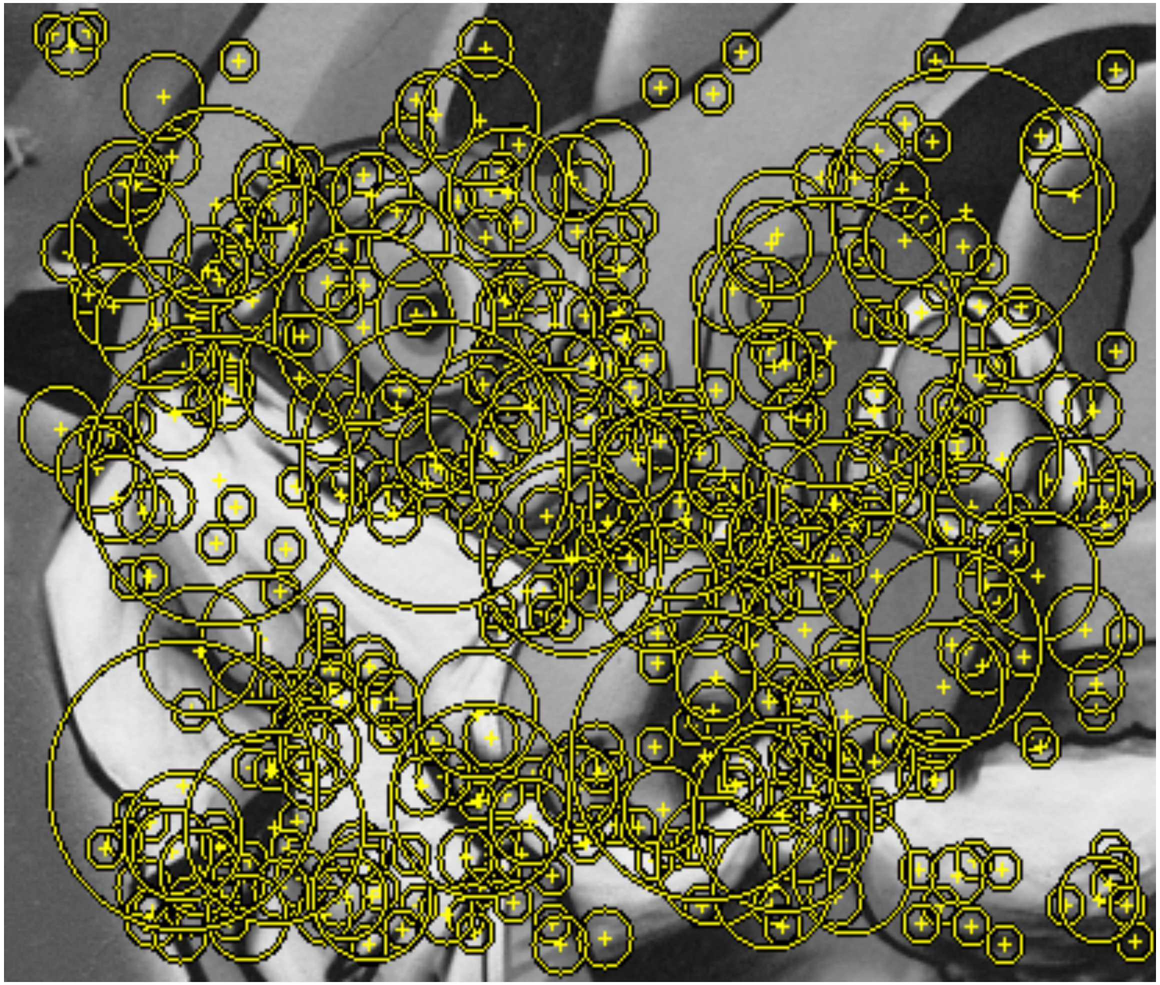
SIFT

Descriptor is based on blob detection, at several scales, i.e. local extrema of the Laplacian-of-Gaussian, or LoG



Computer Vision

SIFT

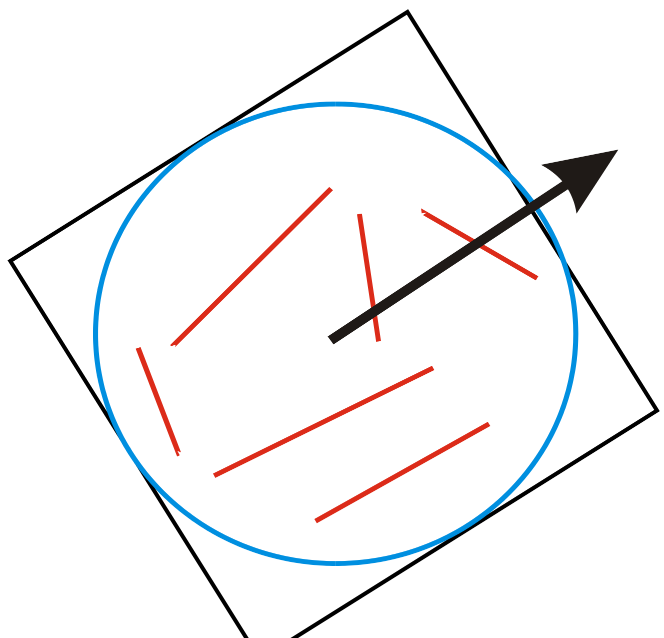


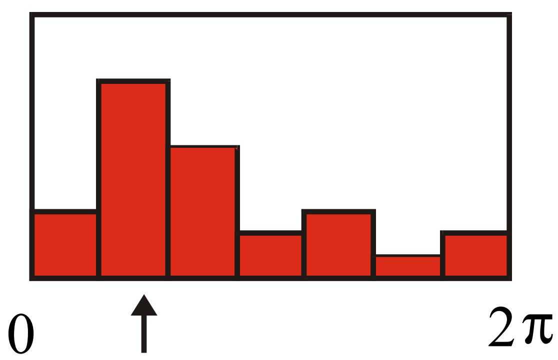
Computer Vision

SIFT

Dominant orientation selection

- Compute image gradients
- Build orientation histogram
- Find maximum

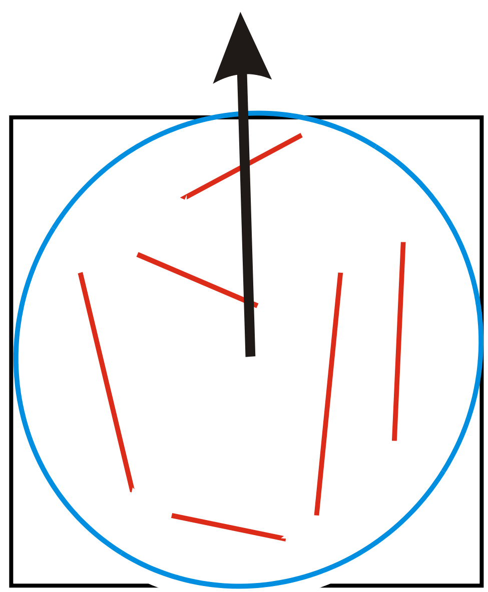


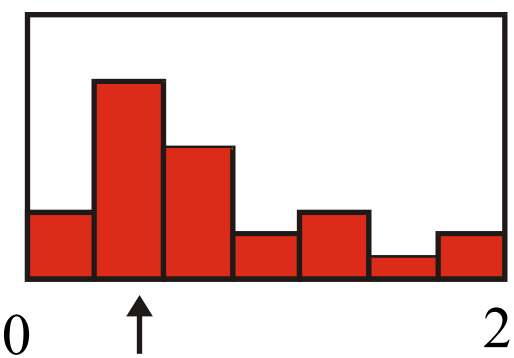


Computer Vision

SIFT

- Dominant orientation selection
  - Compute image gradients
  - Build orientation histogram
  - Find maximum





0 2π

Computer Vision

SIFT

- Thresholded image gradients are sampled over a grid
- Create array of orientation histograms within blocks
- 8 orientations x 4x4 histogram array = 128 dimensions
- Apply weighting with a Gaussian located at the center
- Normalized to unit vector

(Fig. shows 2x2, actually 4x4)

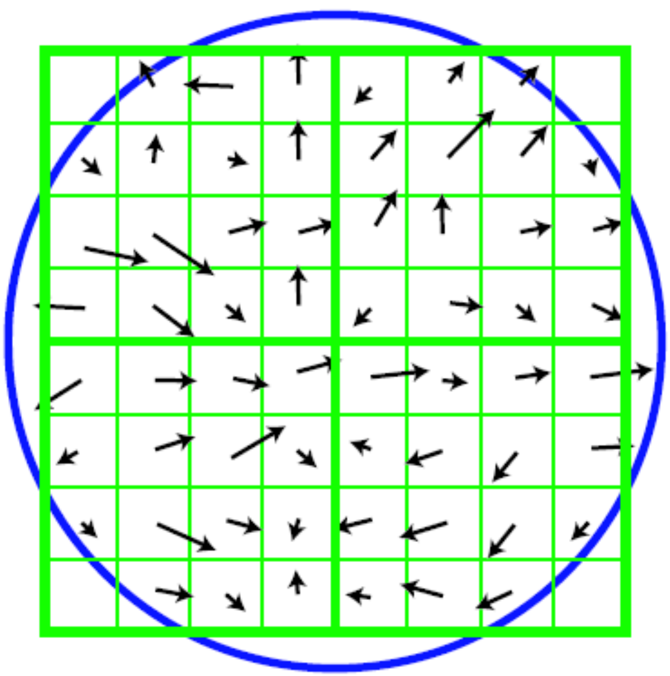
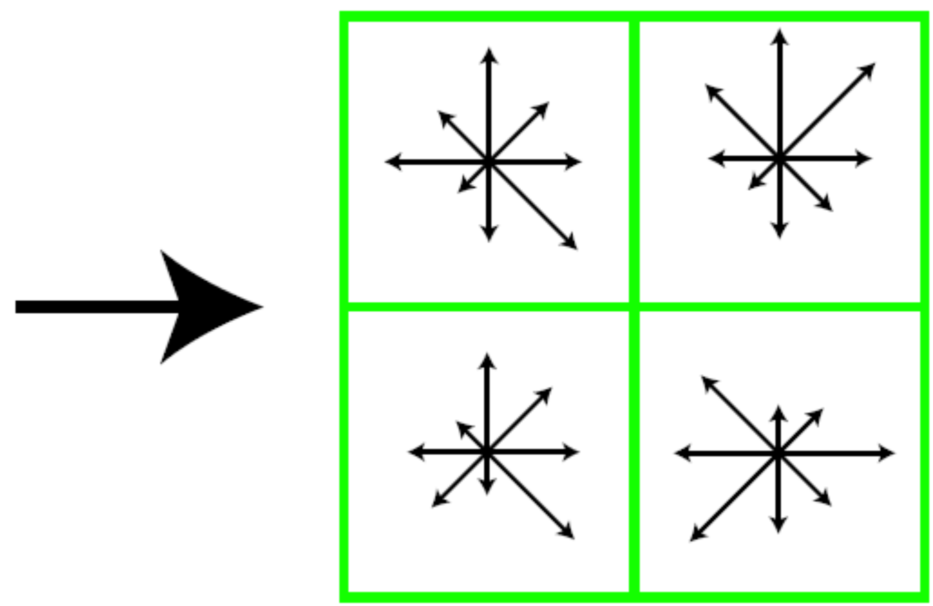


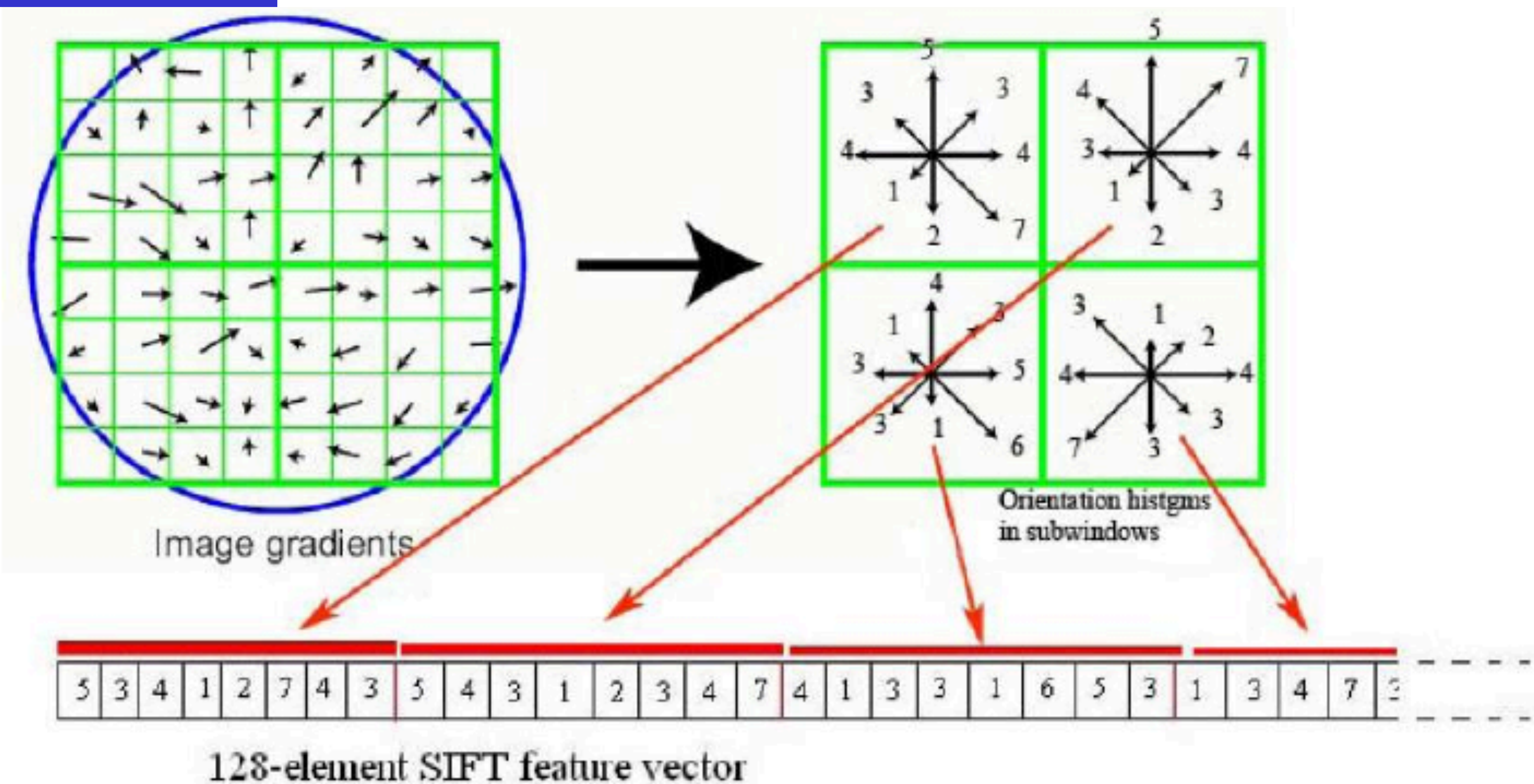
Image gradients



Keypoint descriptor

SIFT

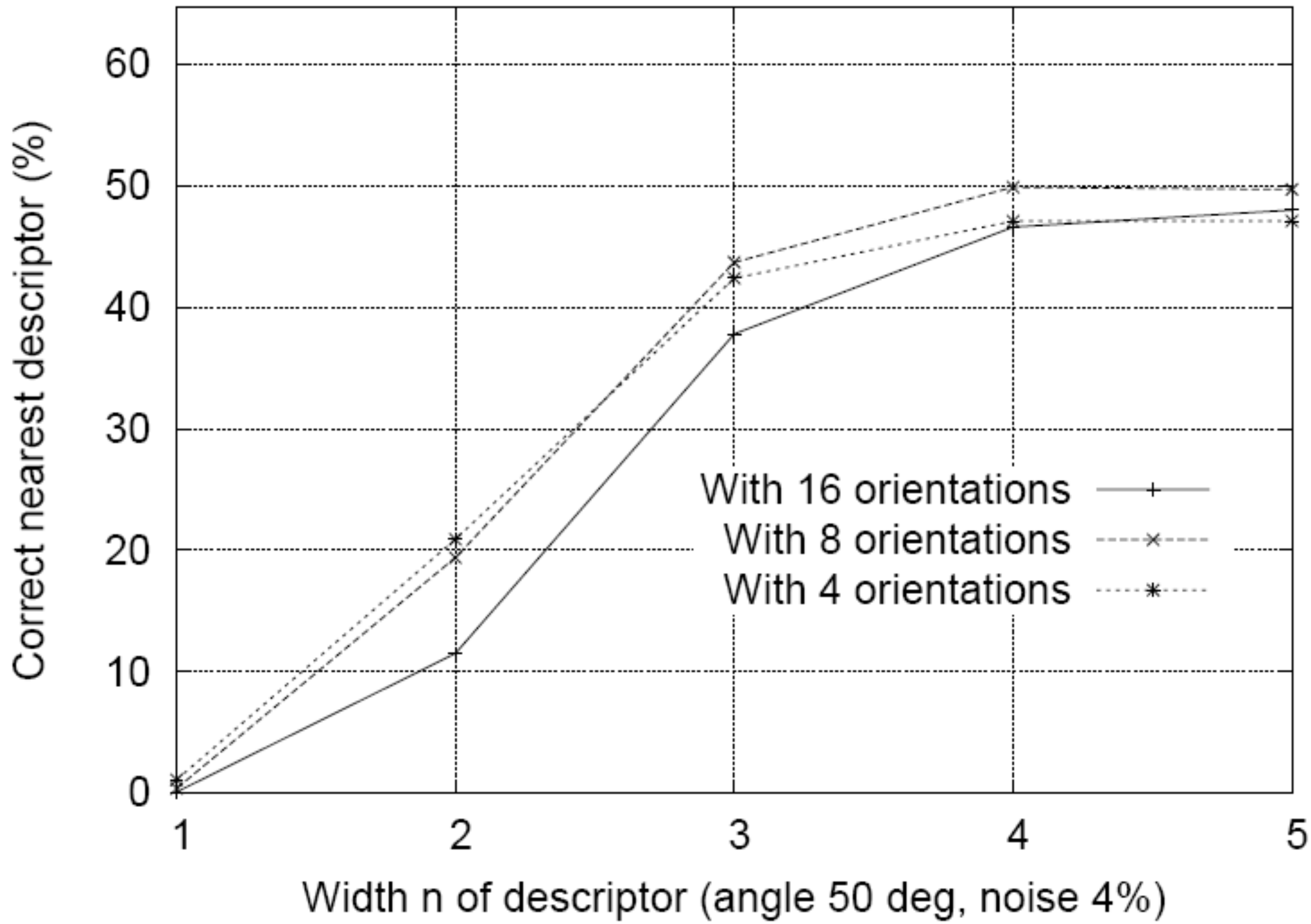
The descriptor is a vector concatenating the cell histograms…
Its total dimension is 128



Computer Vision

SIFT

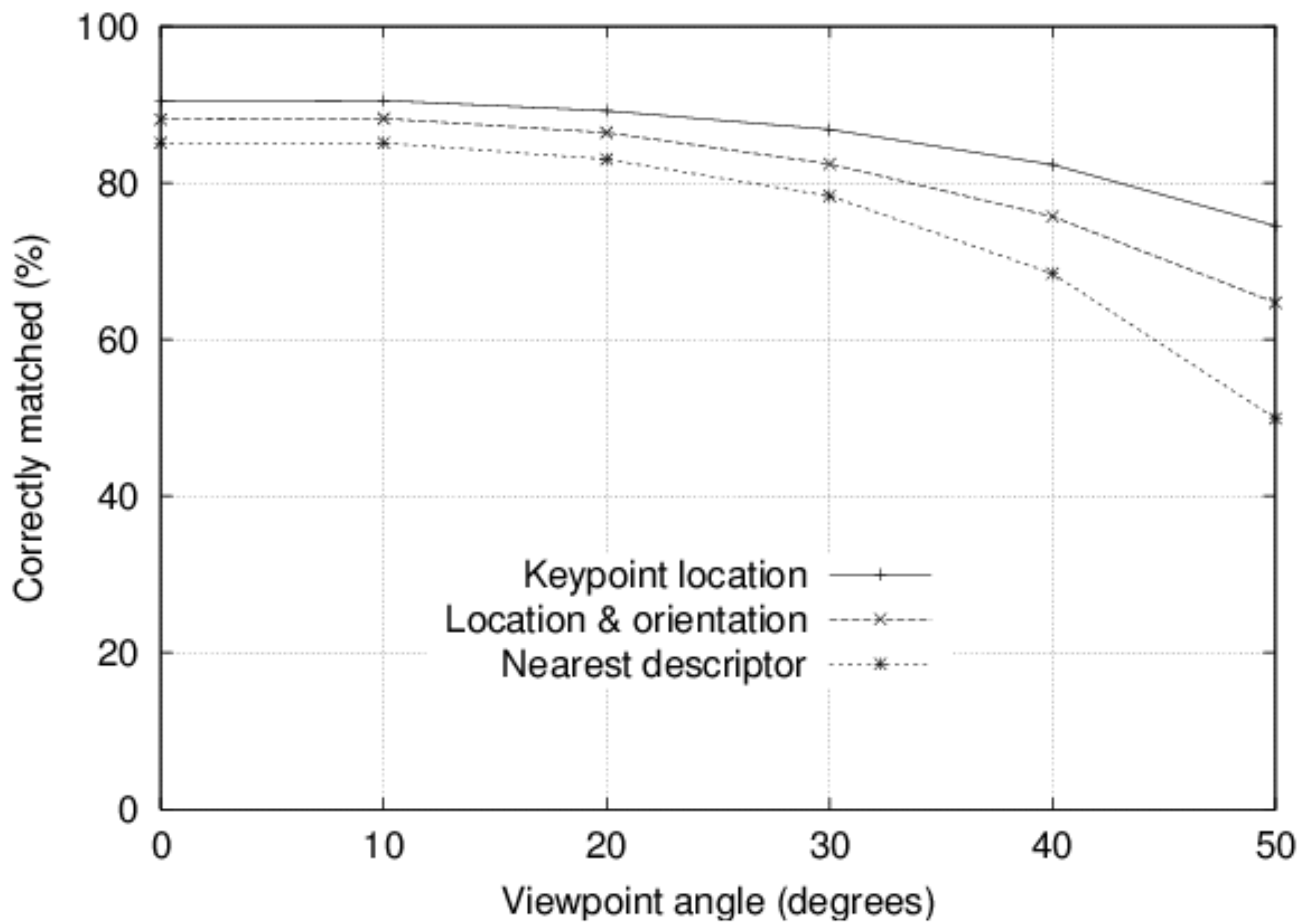
Carefully crafted… e.g. why 4 x 4 x 8 ?



Computer Vision

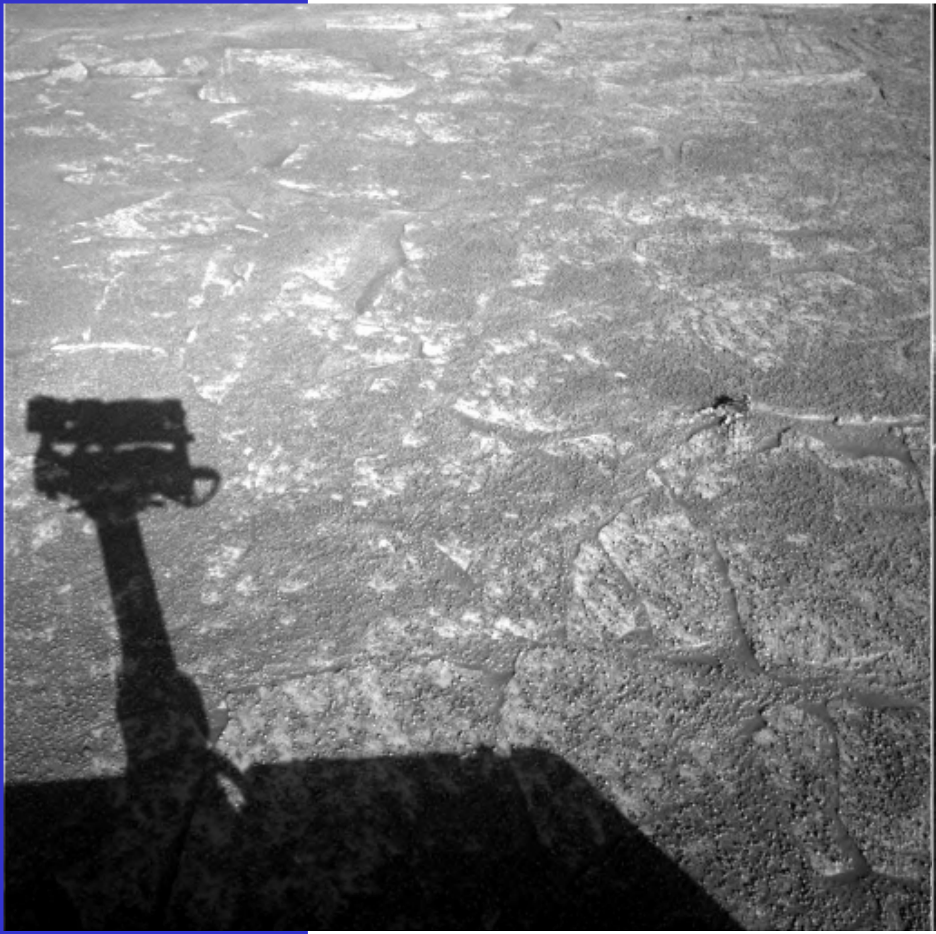
SIFT

Sensitivity to affine changes… quite good !!!



SIFT

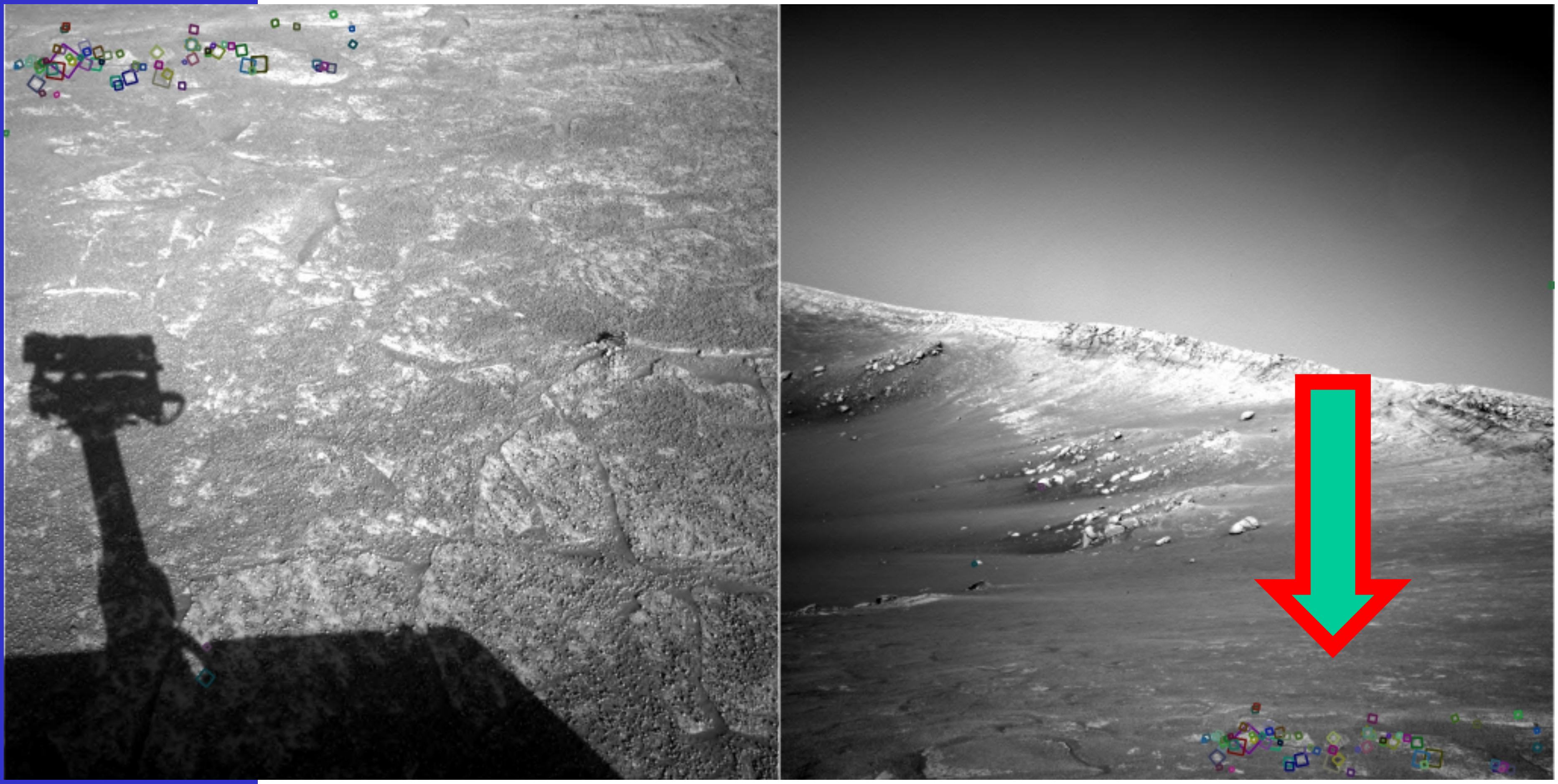
Can you find correspondences ?



NASA Mars Rover images

SIFT

SIFT can…

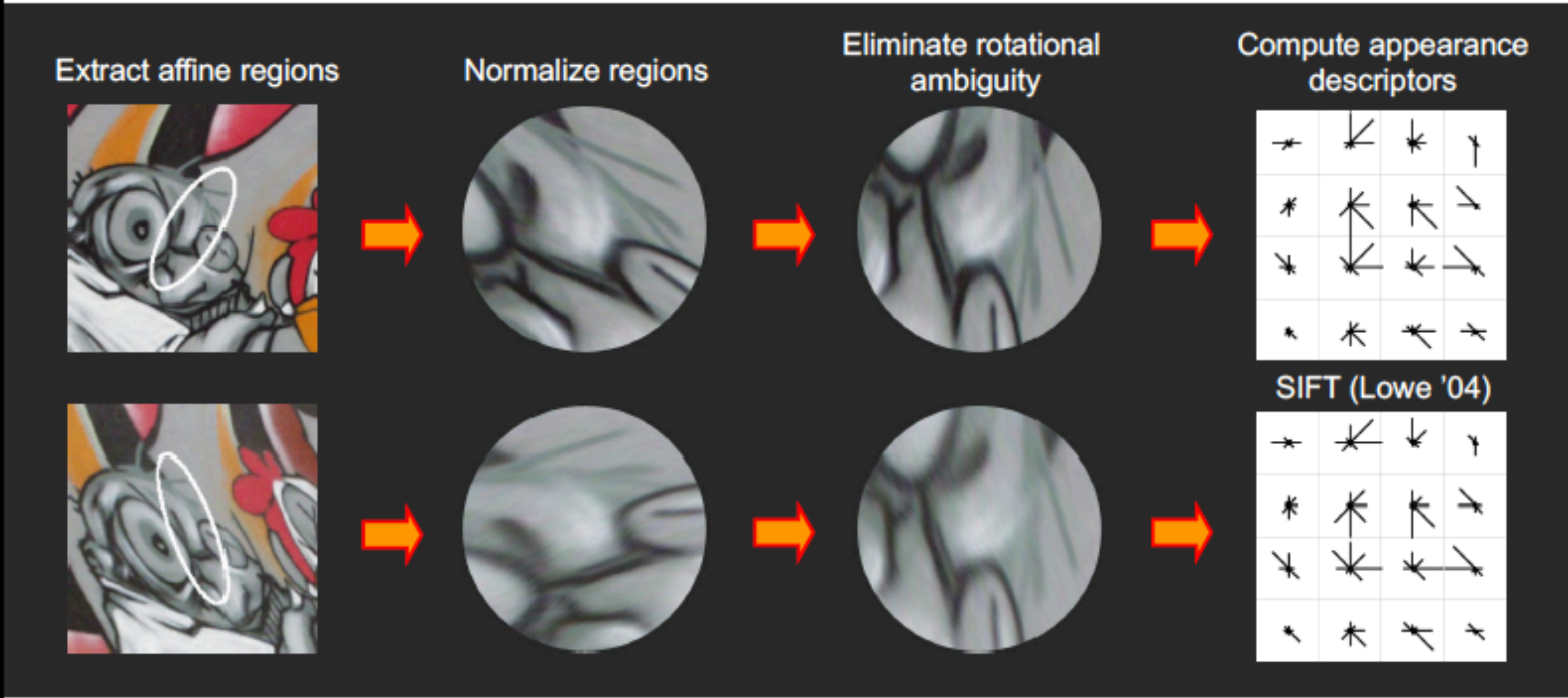


NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

SIFT

NB: detectors and descriptors can be mixed, through normalization

e.g. create an affinely invariant region (affinely inv. detector), then describe content with SIFT:



SURF efficient alternative to SIFT

Application: robot navigation

SURF

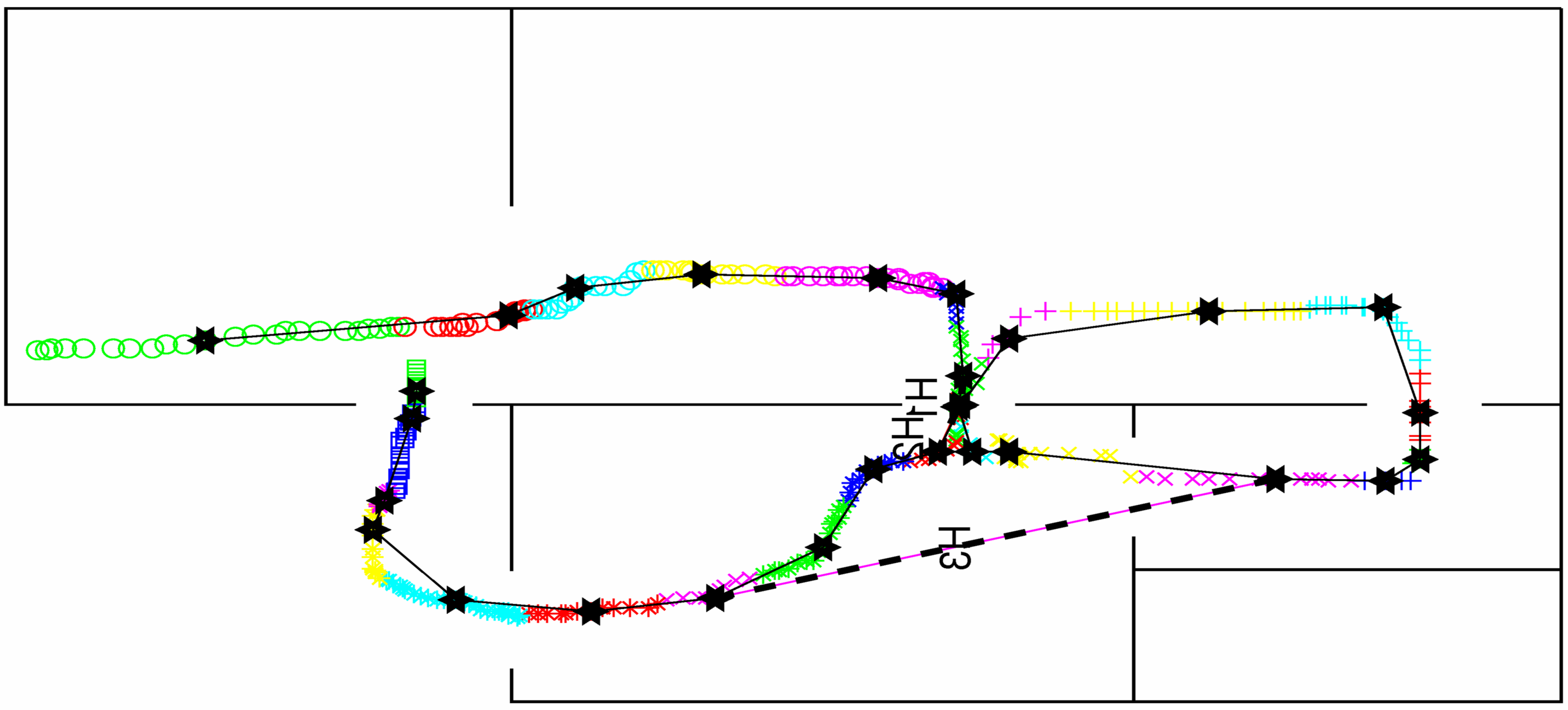
- Robot equipped with omnidirectional camera

Computer Vision

SURF

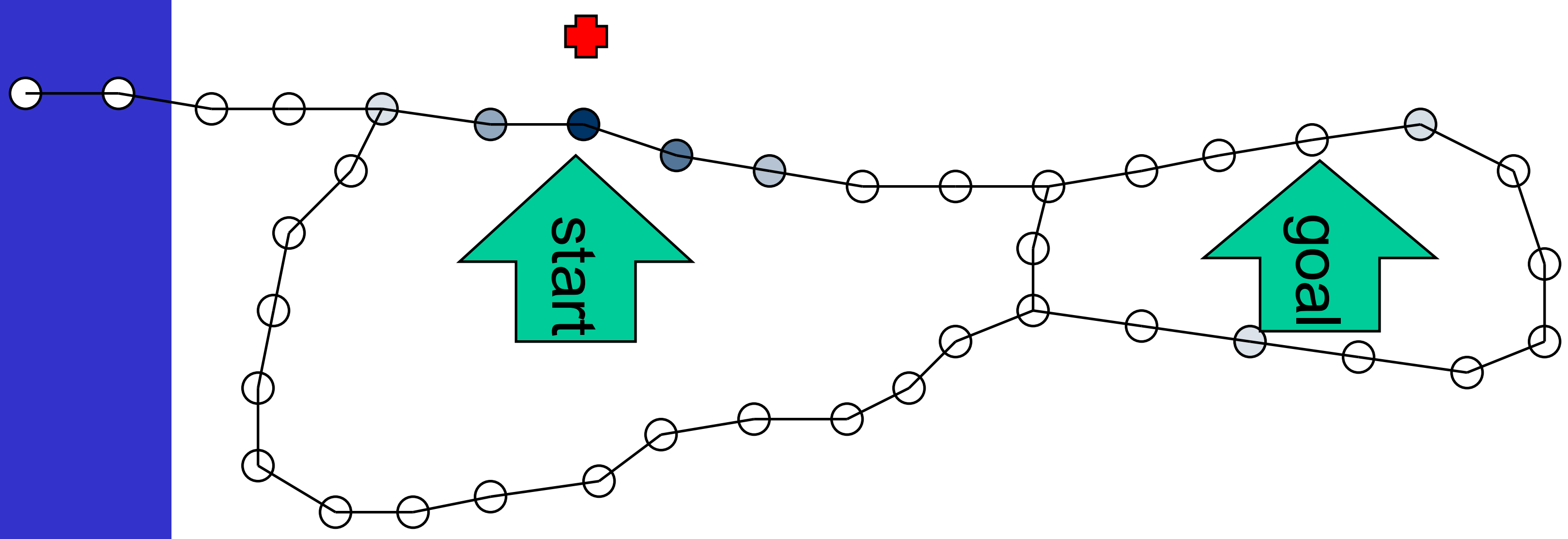
SURF

Robot automatically builds a topological map from SURF features



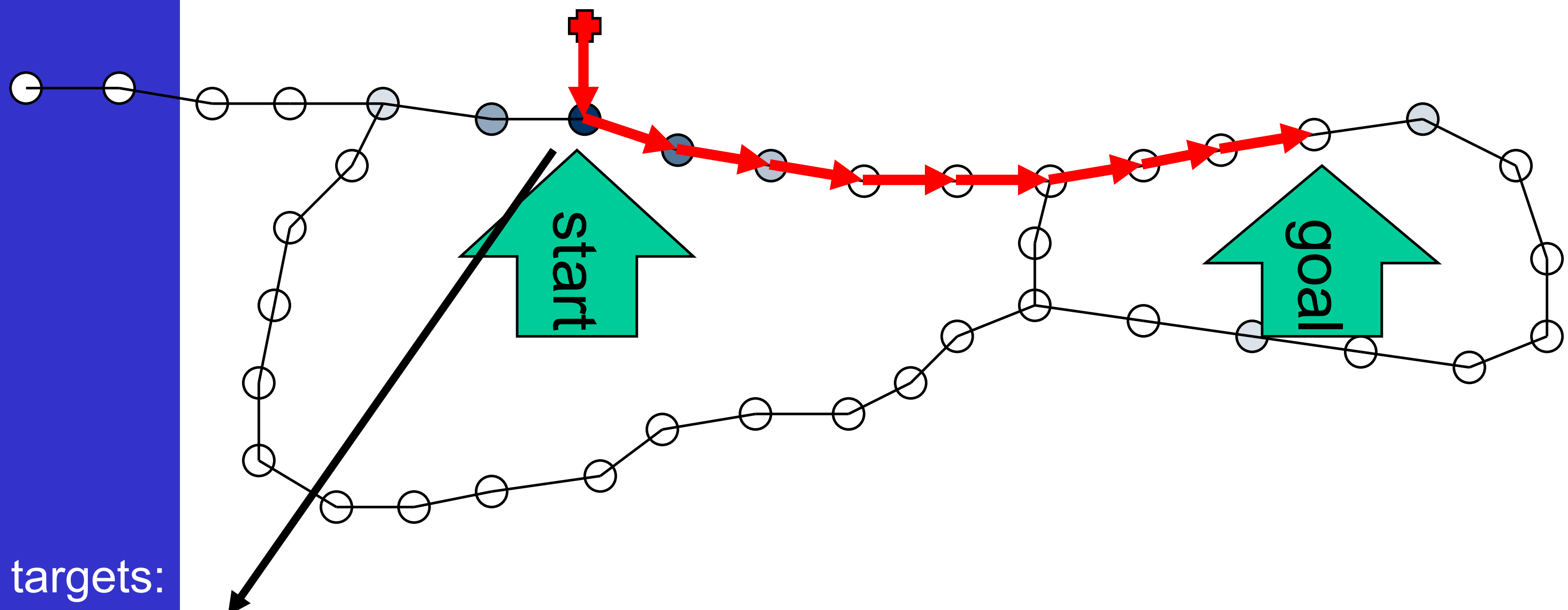
SURF

- Robot automatically builds a topological map, incl. revisited places as link intersections
- Robot can localize itself in the map and navigate to a goal position



Computer Vision

SURF

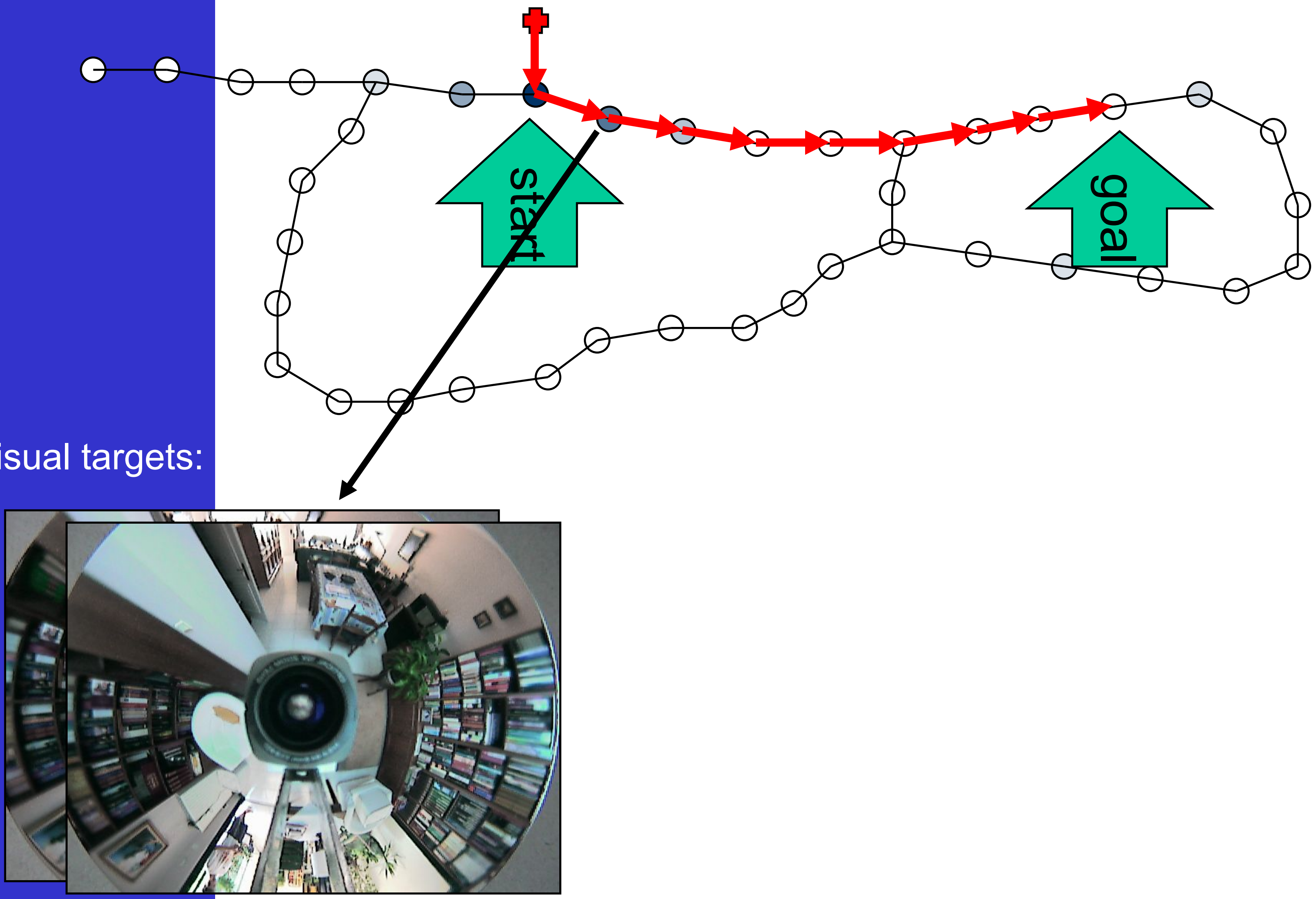


Visual targets:

Navigation becomes an issue
of going to positions with an image
that matches the one taken at the next
target position during teaching

Computer Vision

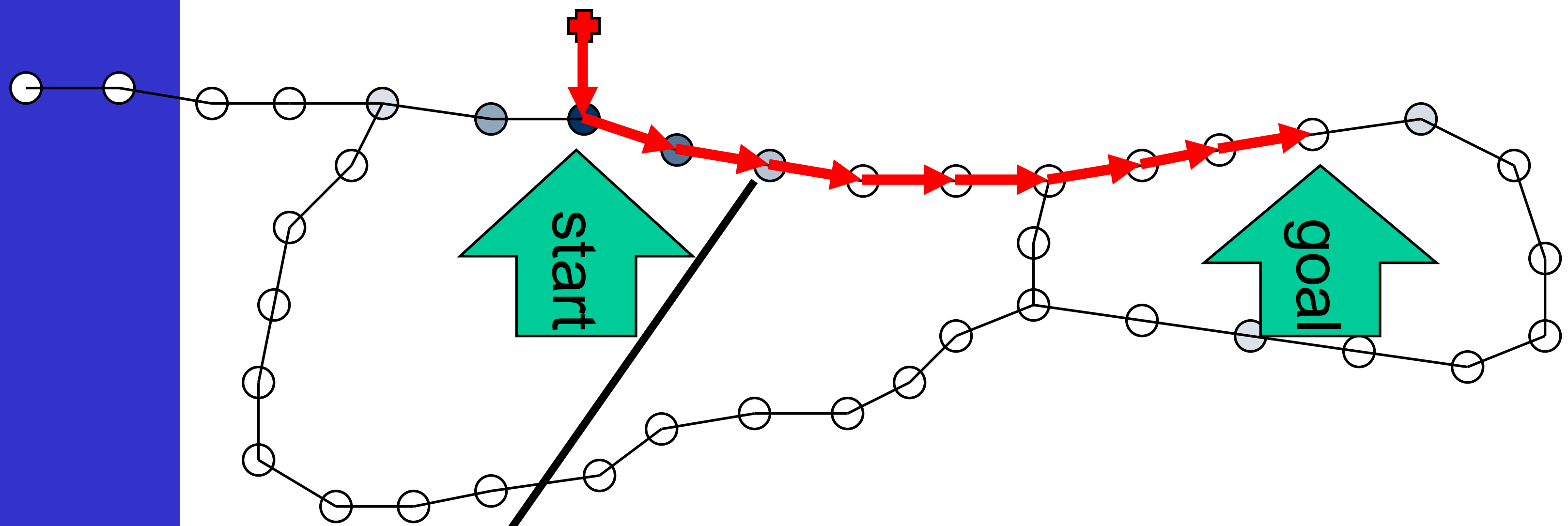
SURF



Visual targets:

Computer Vision

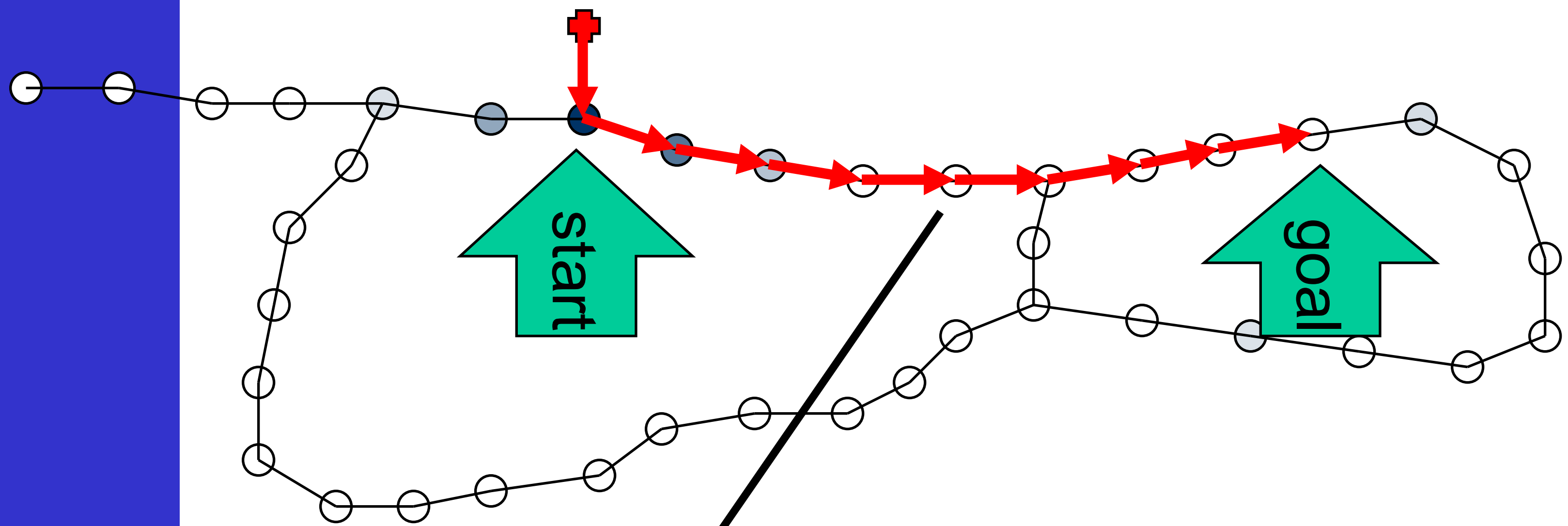
SURF



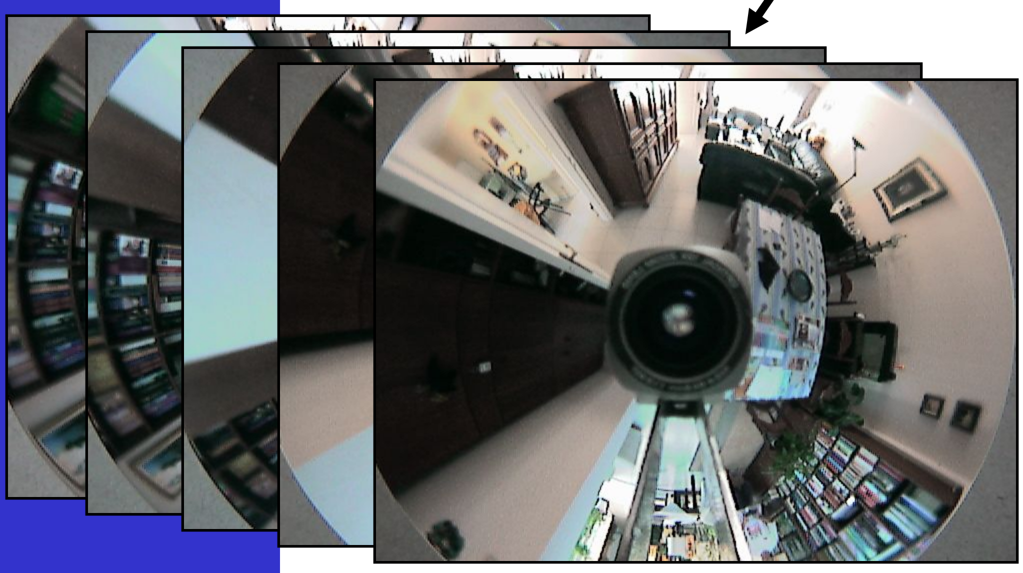
Visual targets:

Computer Vision

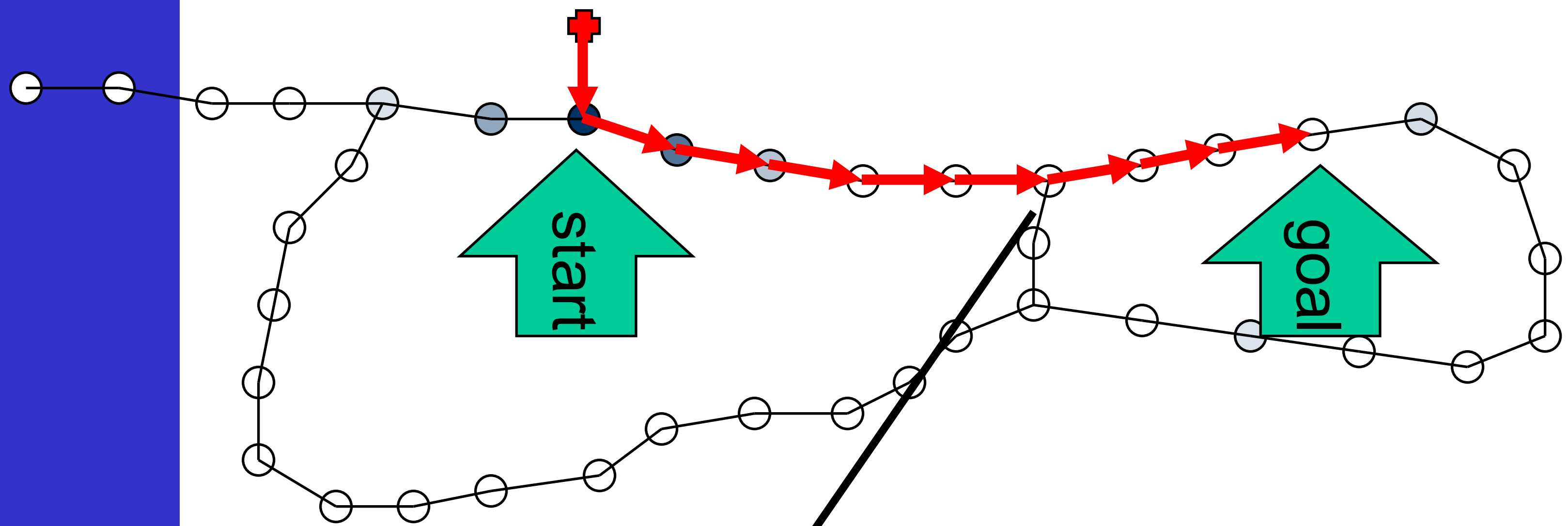
SURF



Visual targets:



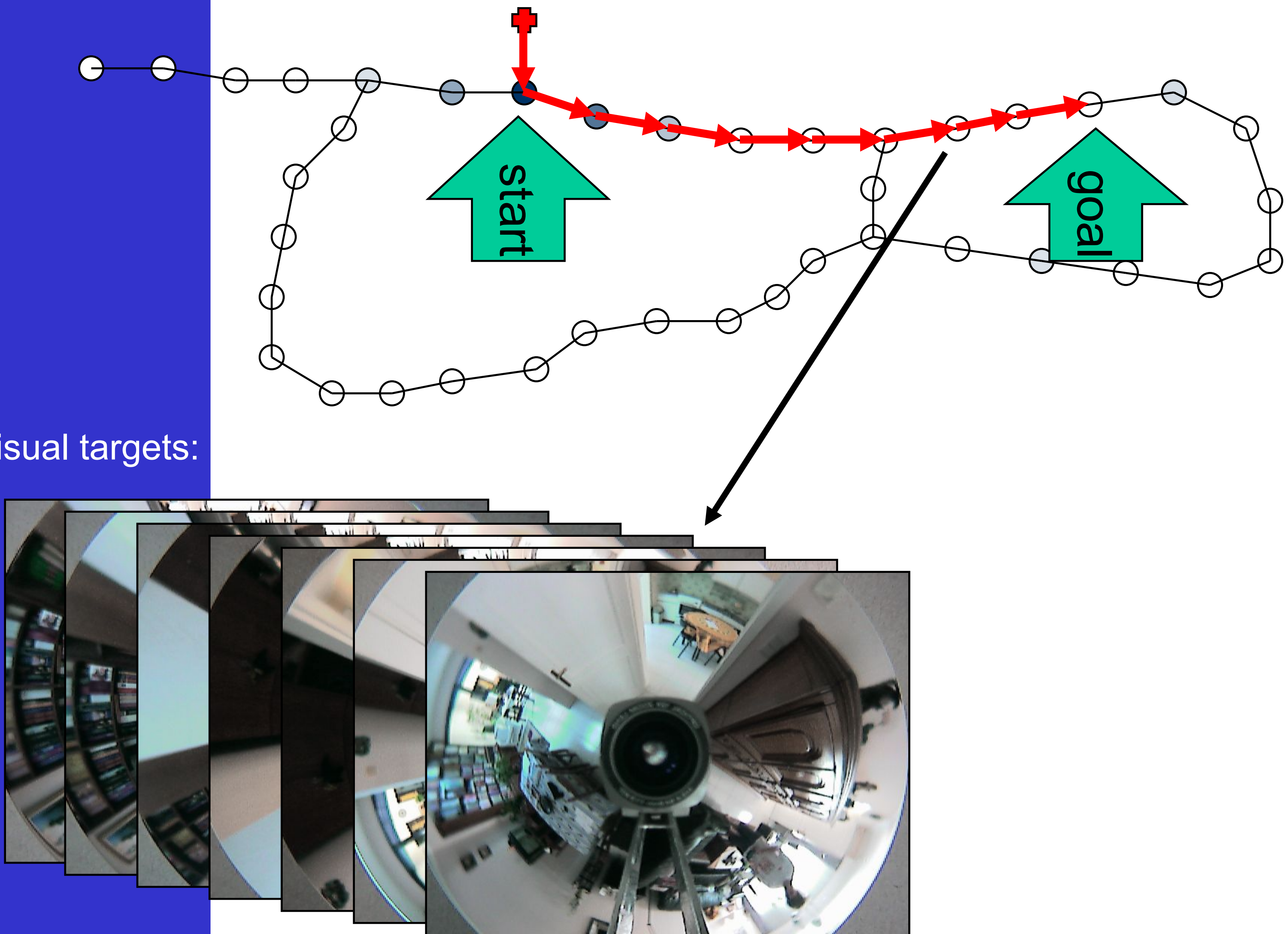
SURF



Visual targets:

Computer Vision

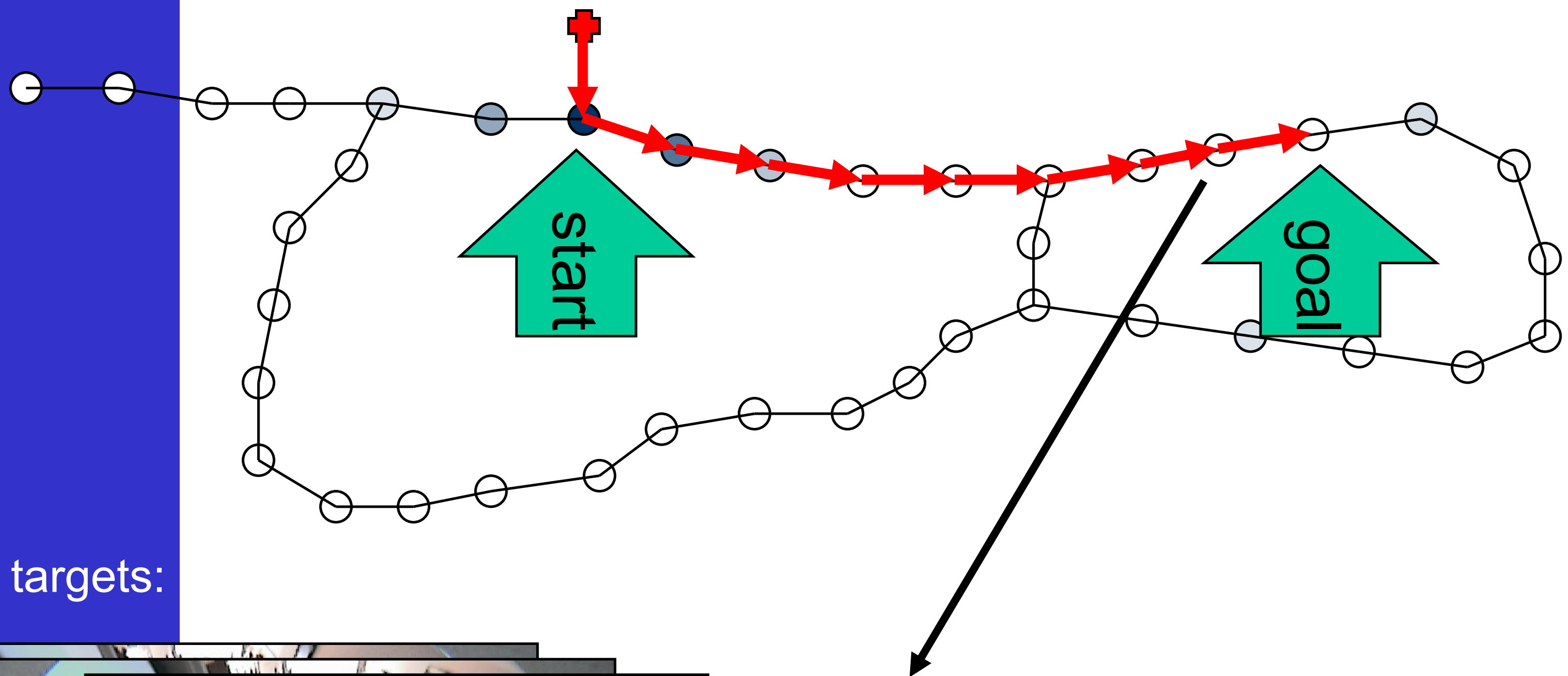
SURF



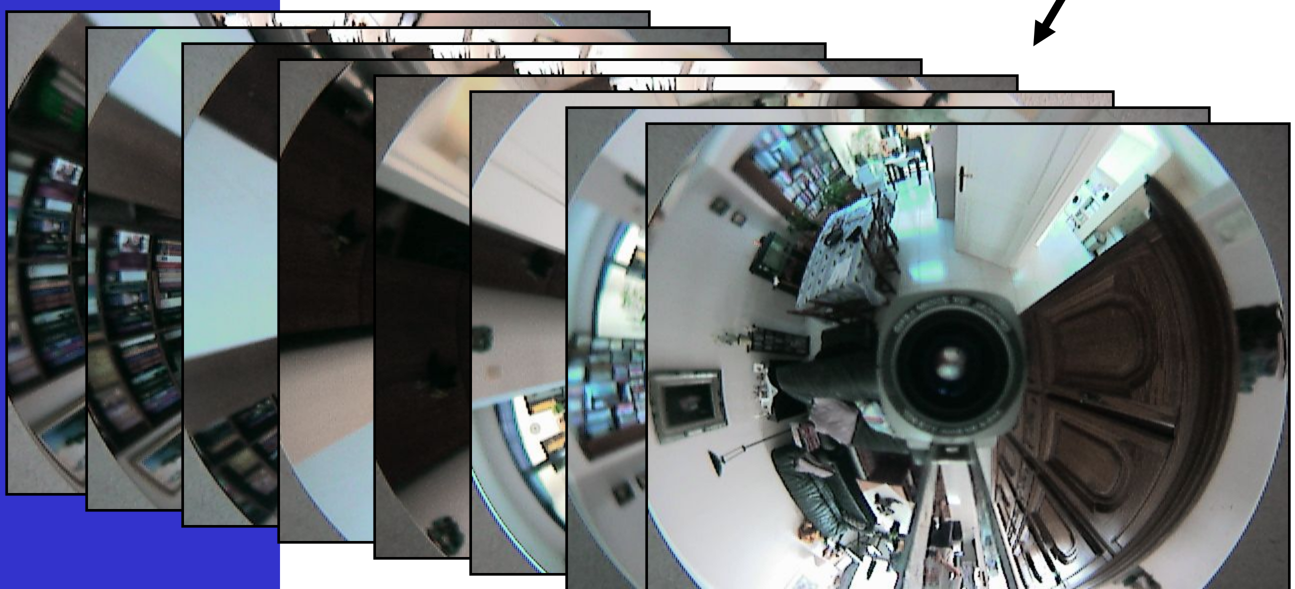
Visual targets:

Computer Vision

SURF

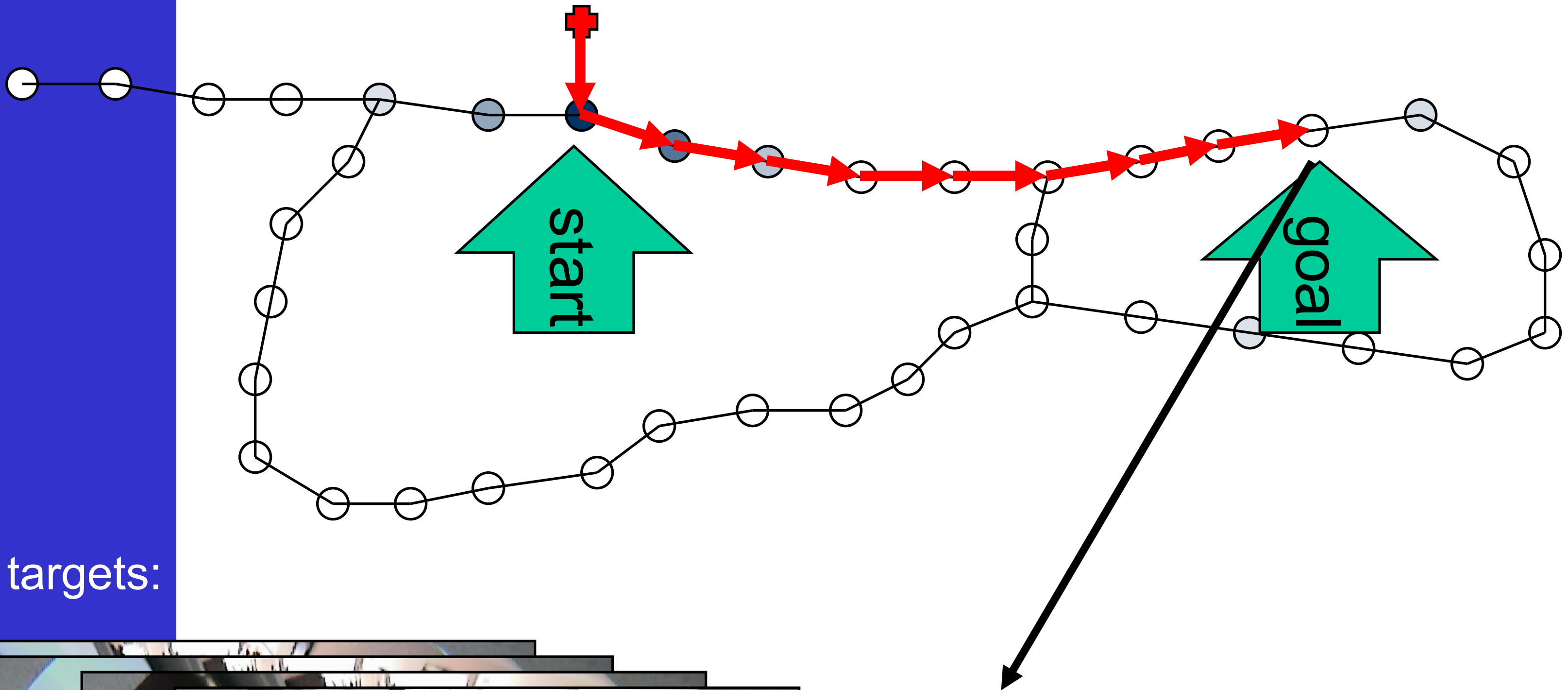


Visual targets:

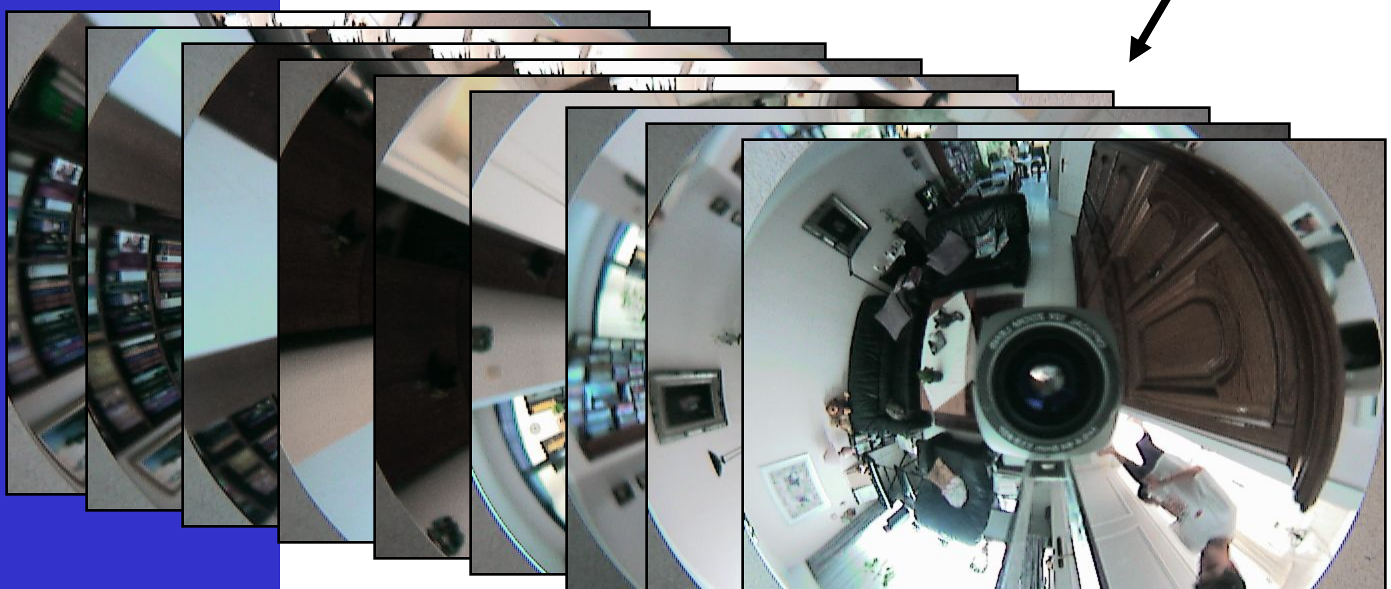


Computer Vision

SURF



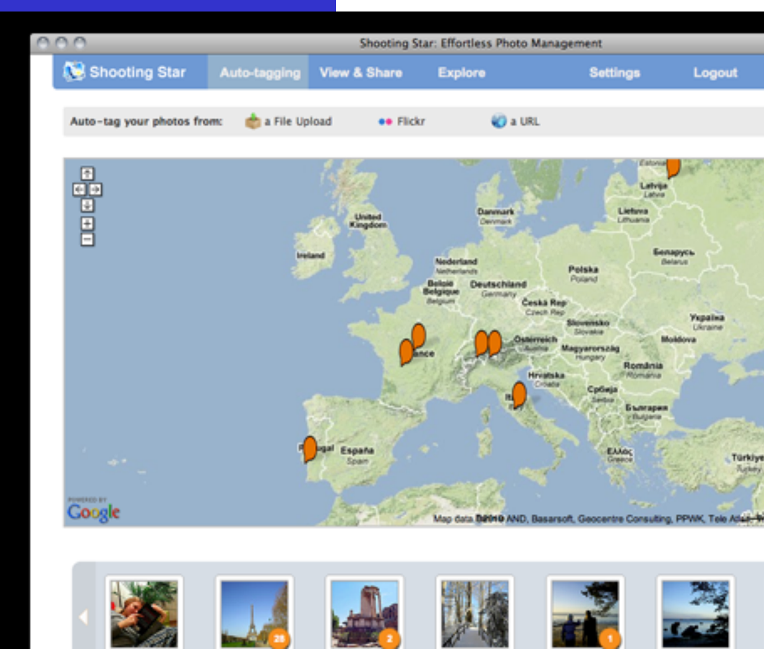
Visual targets:

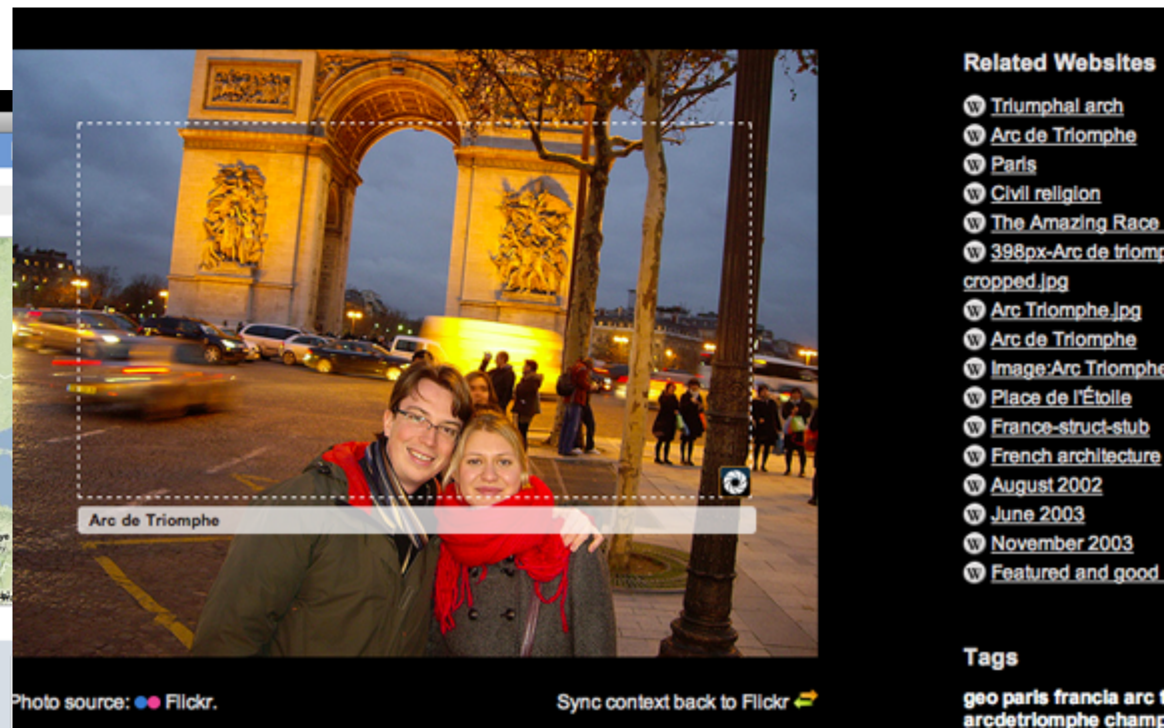


Computer Vision

SURF

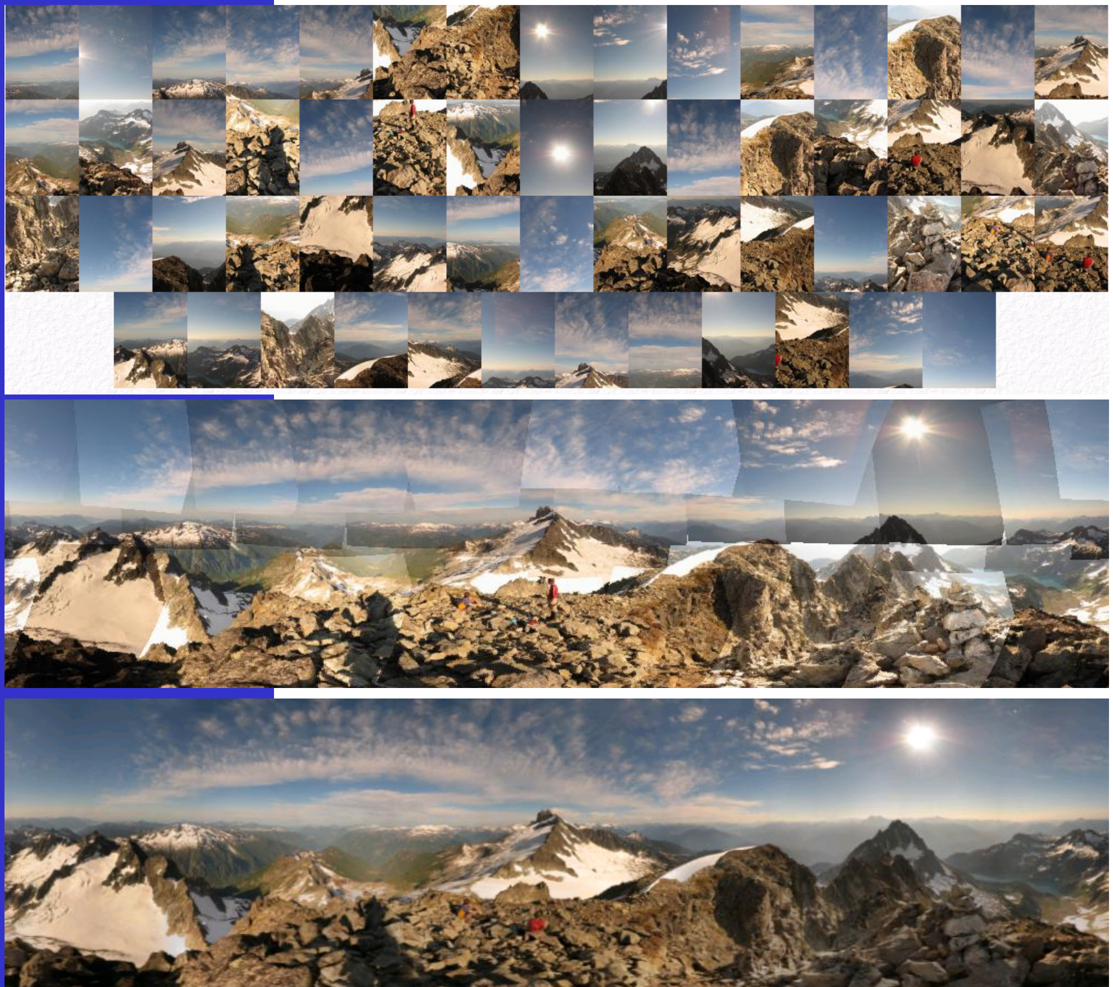
Landmark, label, cover,… recognition also used for prior kooaba Déjà Vu app (then Qualcomm) for the iPhone, Android, and Symbian platforms





Computer Vision

Automatic mosaicing



http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Computer Vision