Introduction to Robotics
for cognitive science

Dr. Andrej Lúčny

KAI FMFI UK

lucny@fmph.uniba.sk

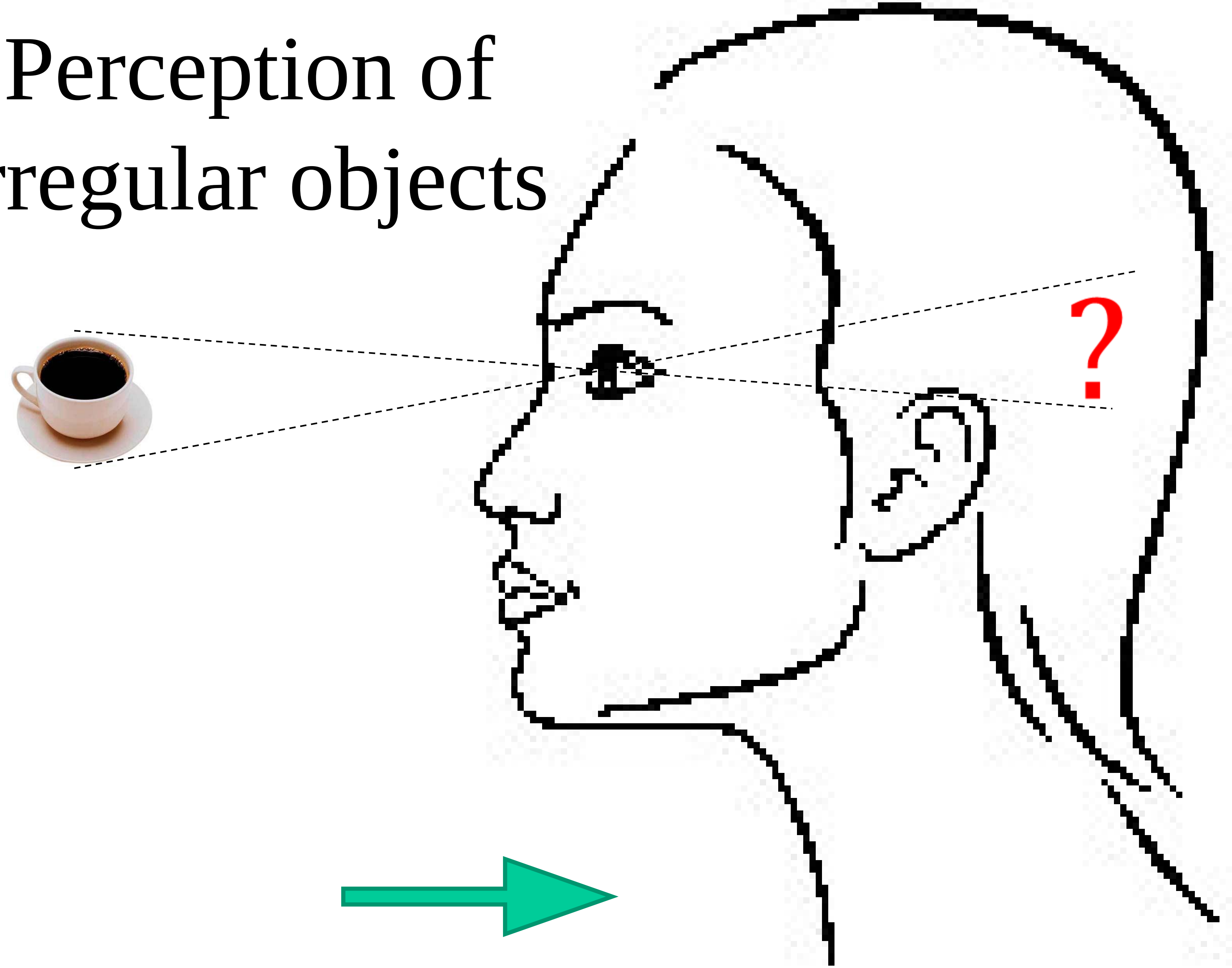
Web page of the subject

www.agentspace.org/kv

Irregular objects

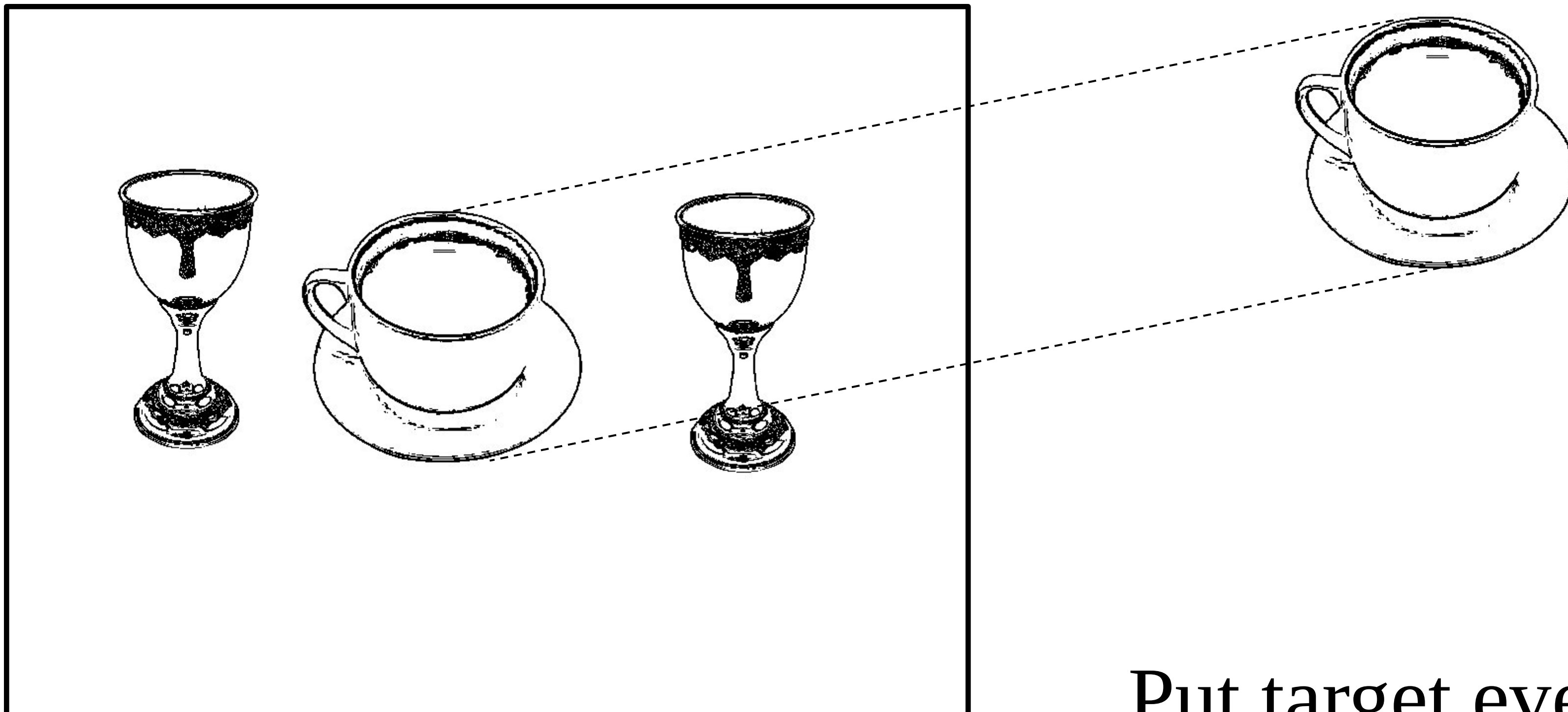
Too many parameters, no rendering algorithm

Perception of irregular objects



Brute force

target

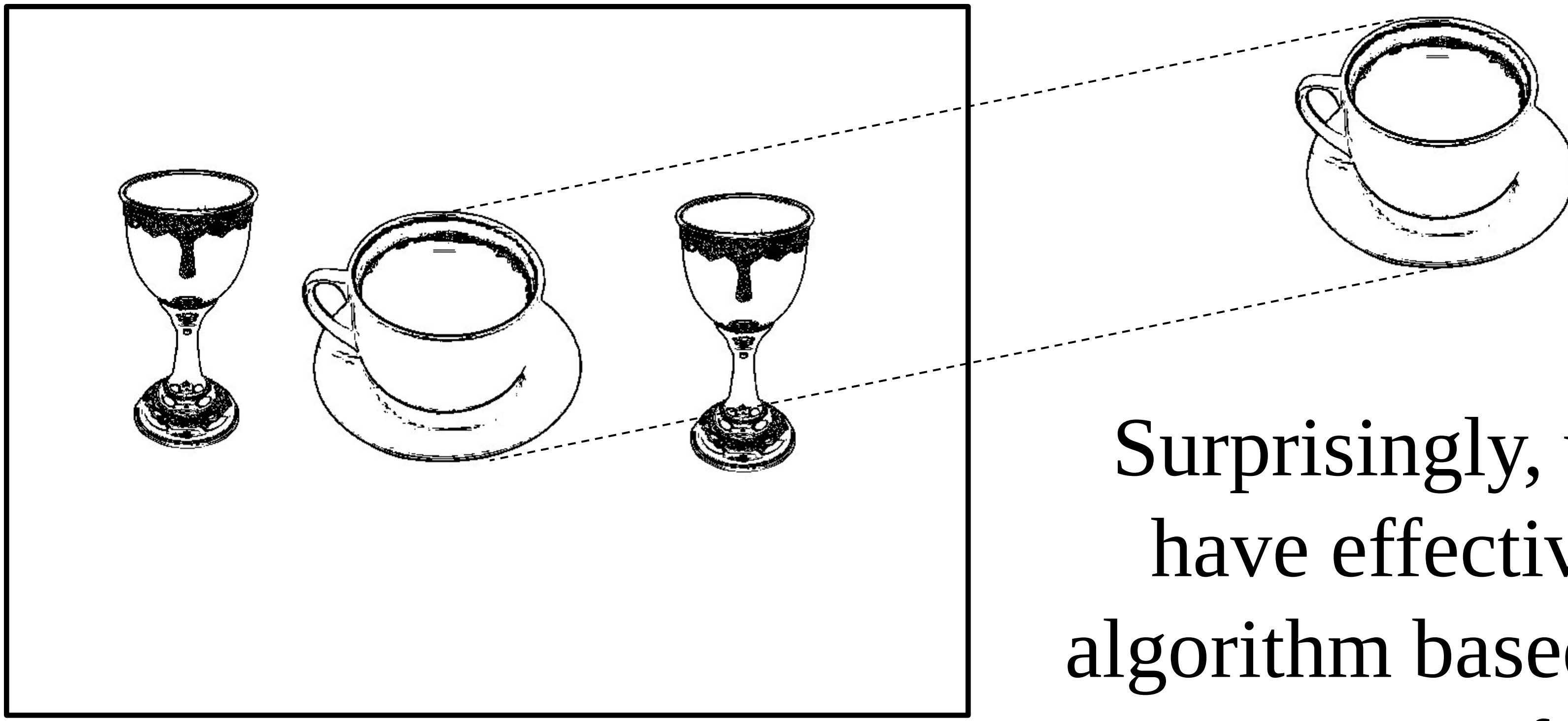


frame

Put target every where, find minimal error

Brute force

reference

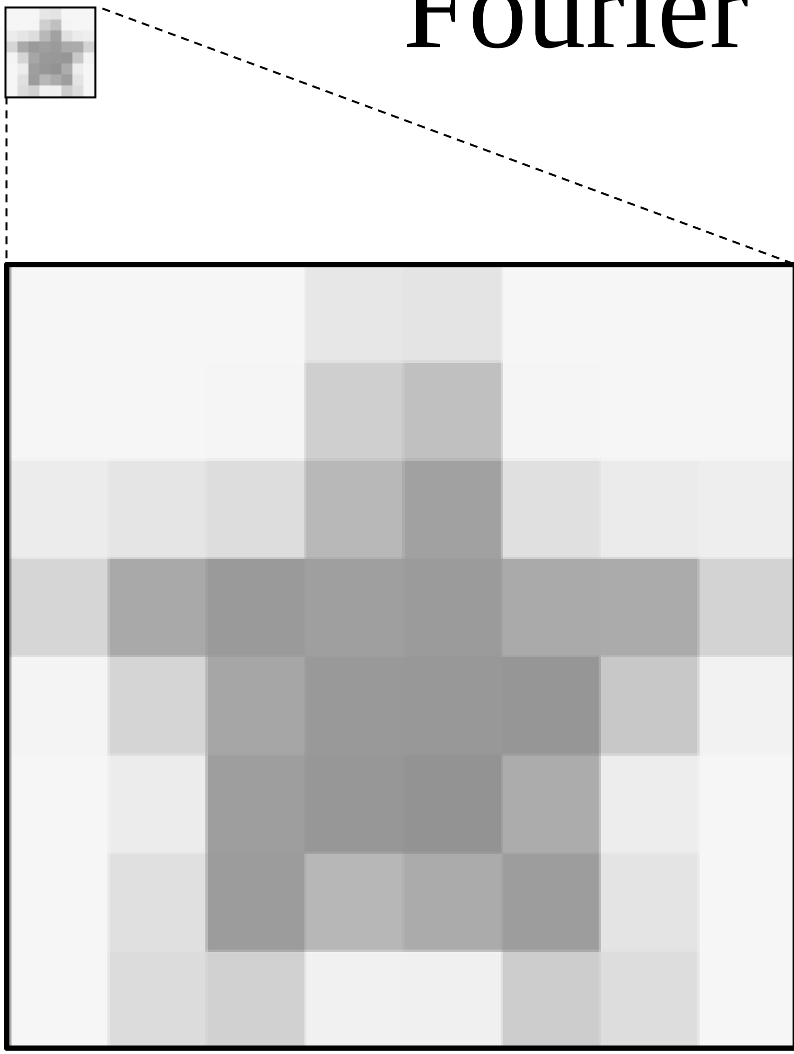


frame

Surprisingly, we have effective algorithm based on Fourier transform O(P log P)

P is number of pixels

Fourier Transform

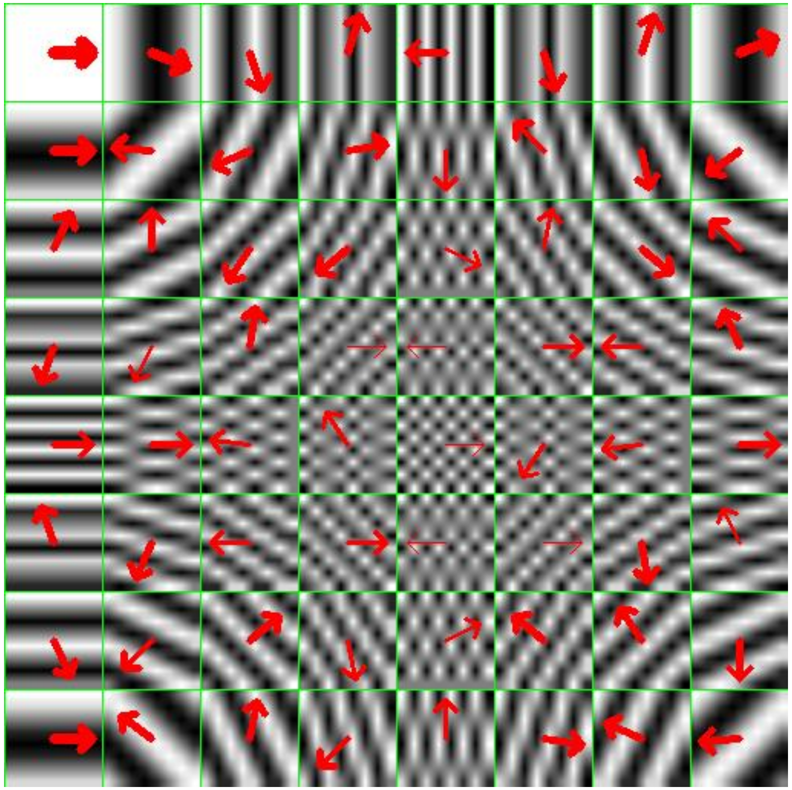


Any image can be effectively expressed as a linear transform of wave images

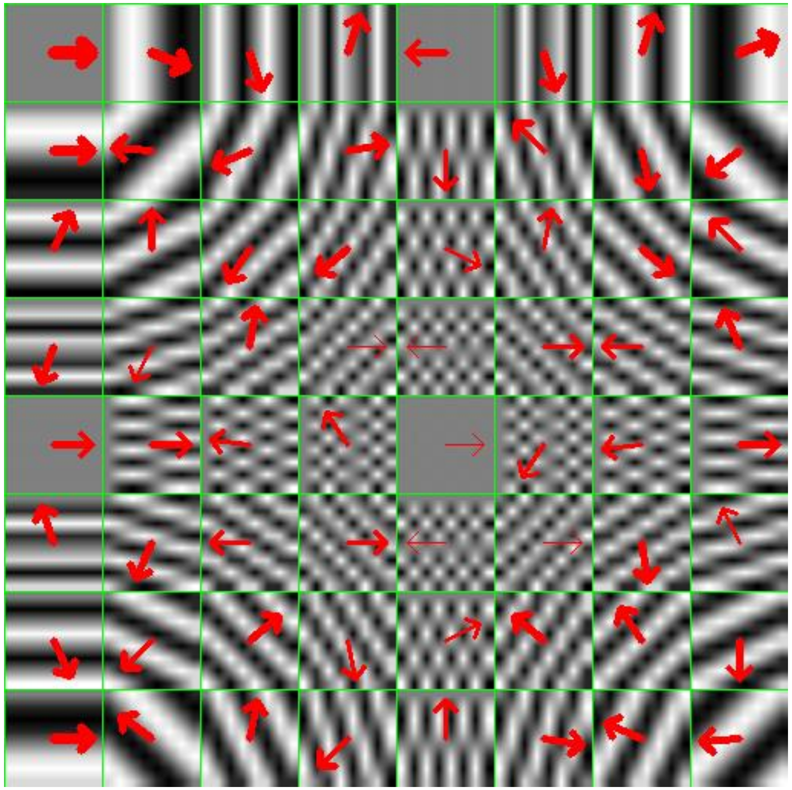
(complex images, complex coefficients)

Fourier Transform

Re



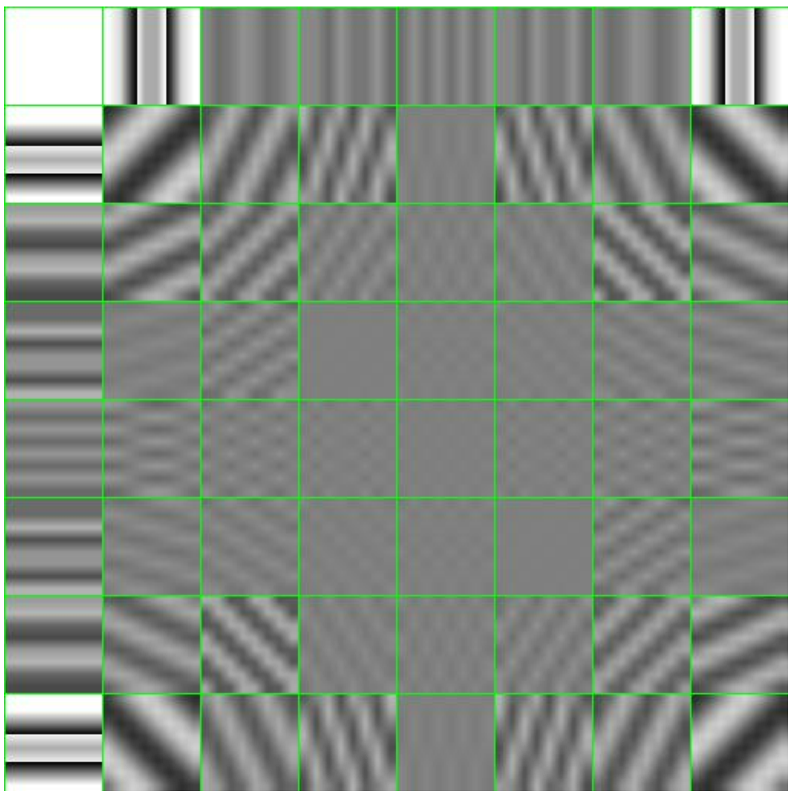
Im



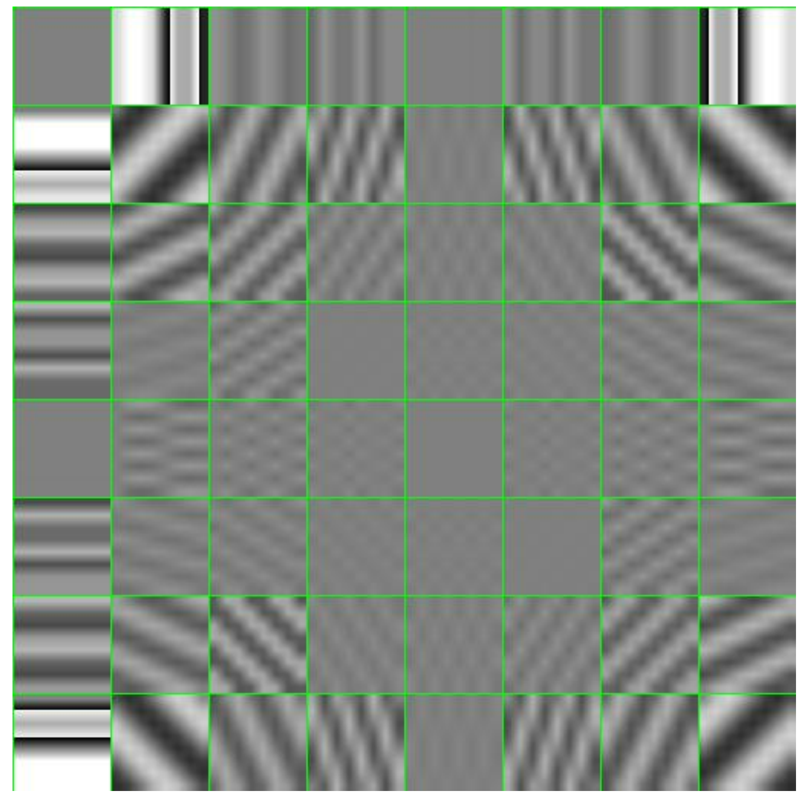
Fourier coefficients are complex numbers, they have amplitude (magnitude) and phase (slope). Phase remains the same when just contrast of image is modified

Fourier Transform

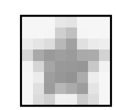
multiplied: Re

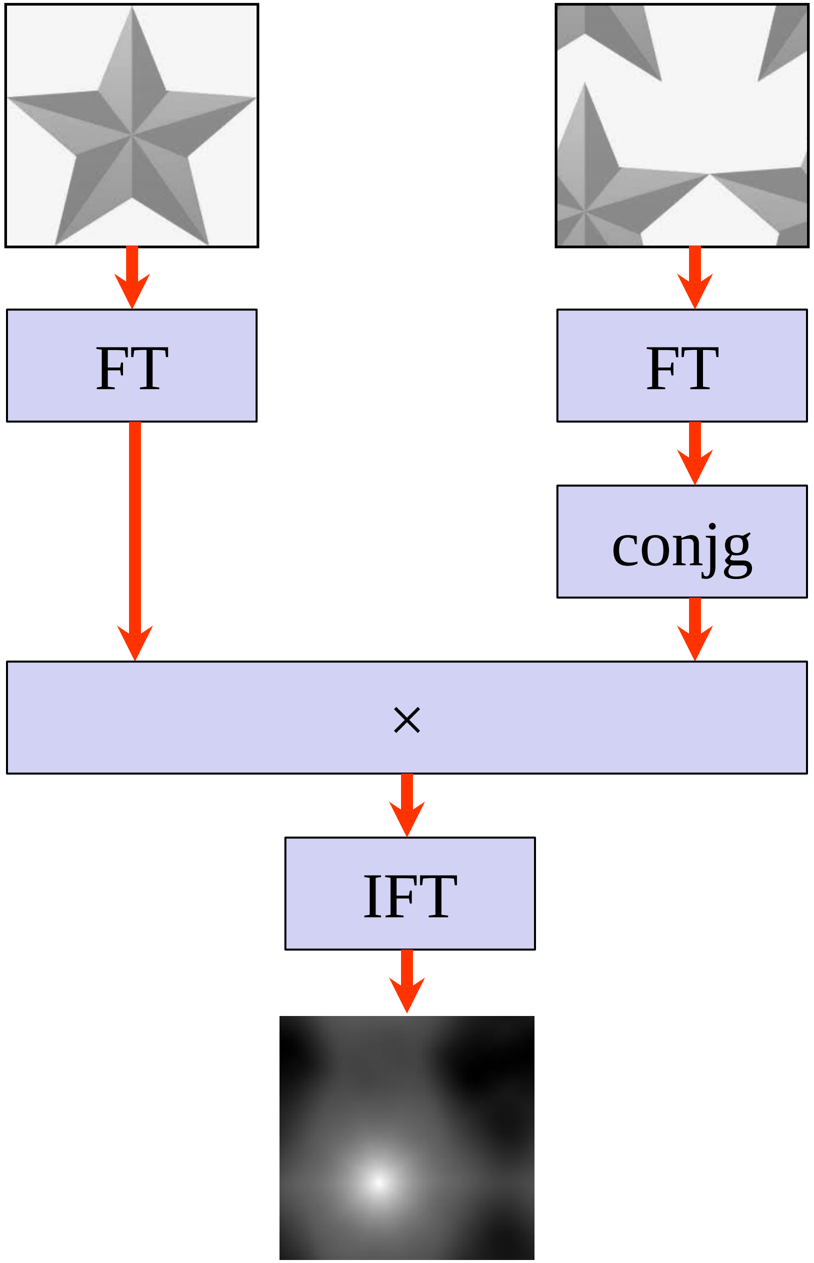


Im



total:

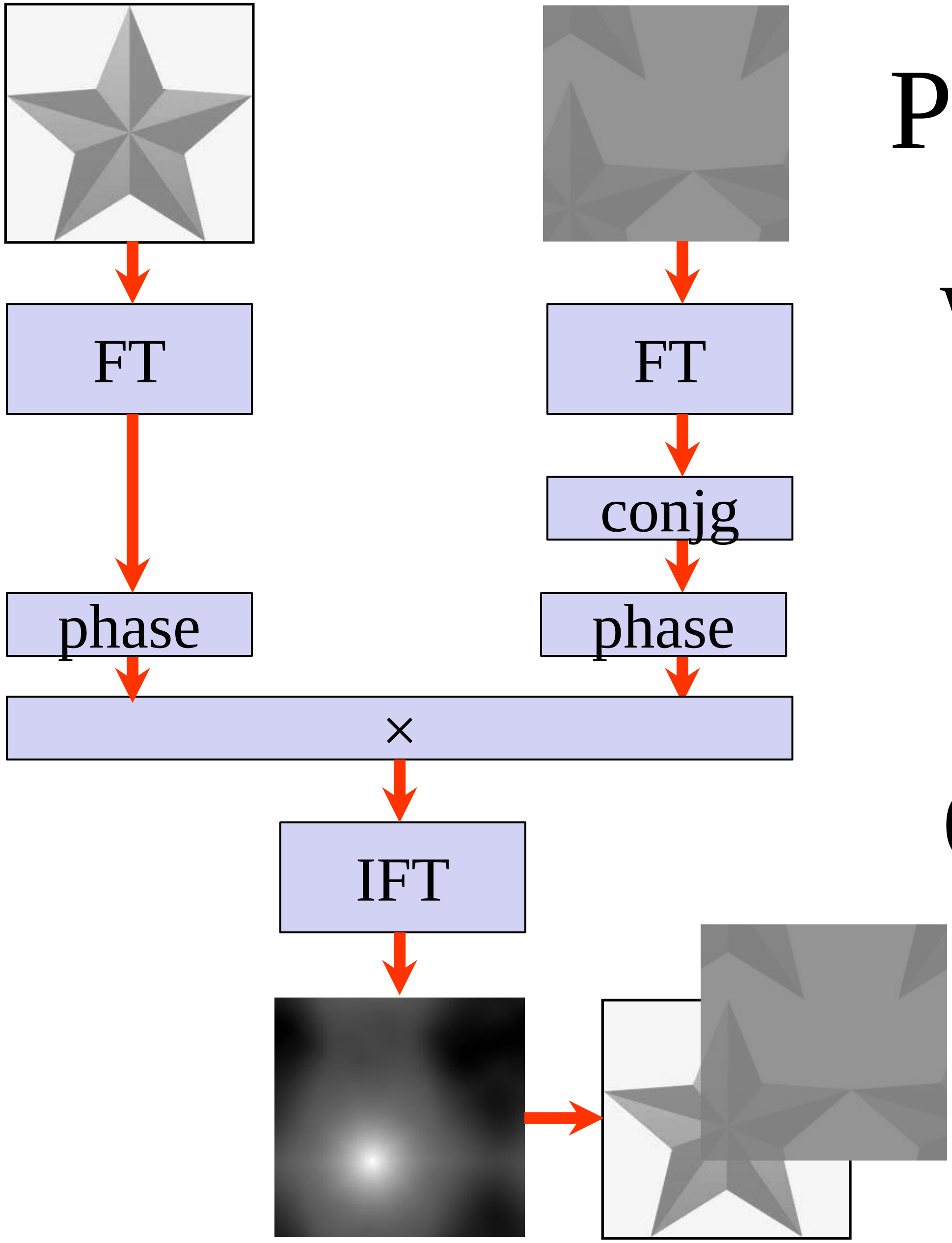




With FT and inverse we can calculate circular convolution of two images.

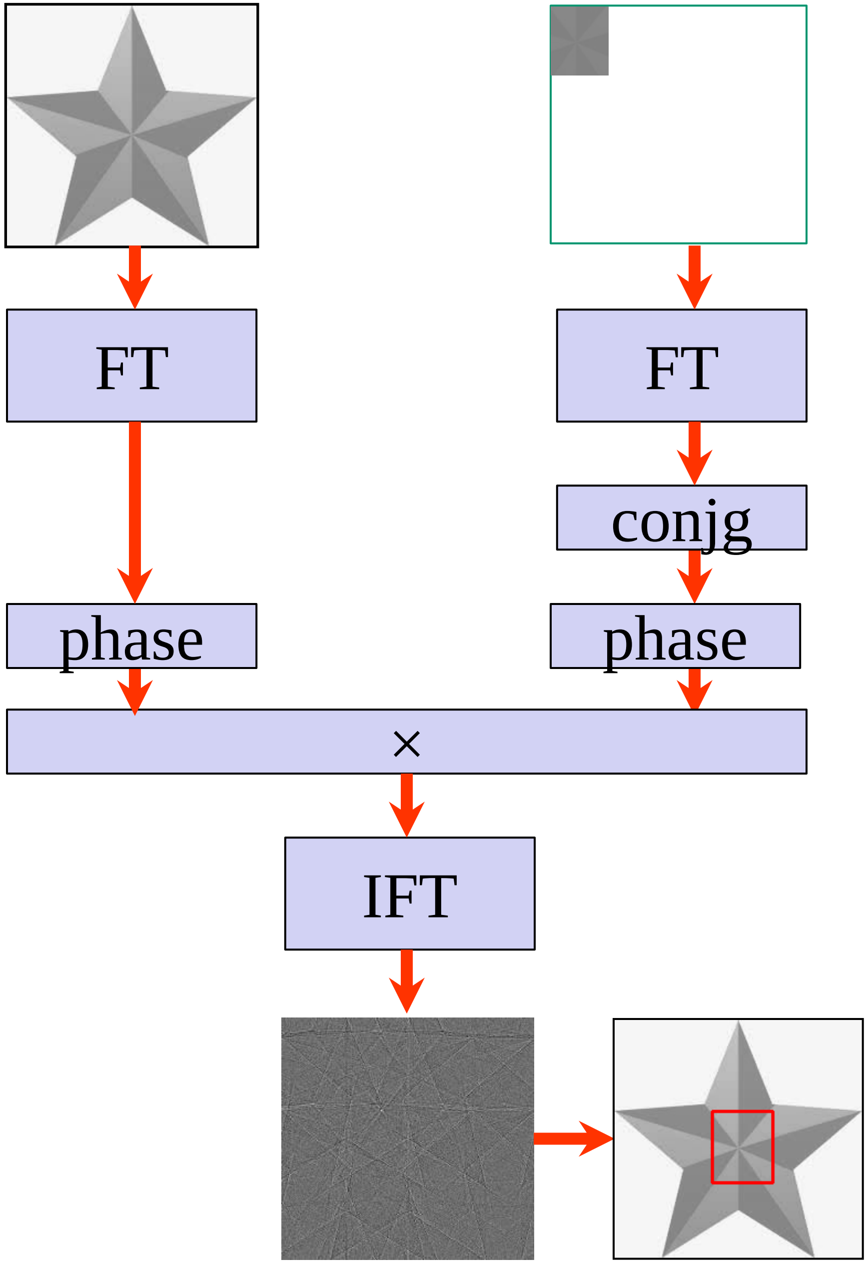
When we reverse one of them and roll by (1,1) we calculated sum of images multiplication for all possible shifts which corresponds to matching error

Phase correlation



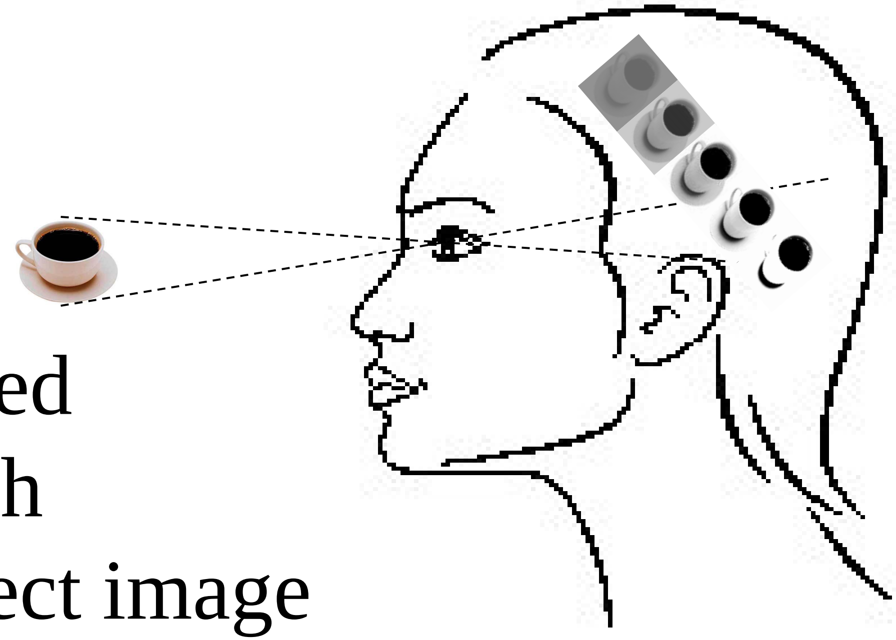
When we pay attention just to the phase, the search becomes independent from lighting condition (contrast modification)

Phase correlation



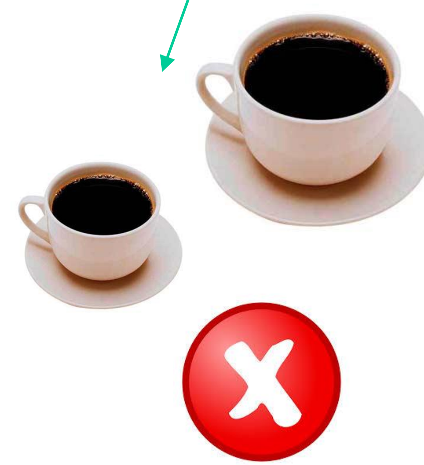
With some risk we can also use padding on part of the image (even with modified contrast) and search for it on the given image

Phase correlation



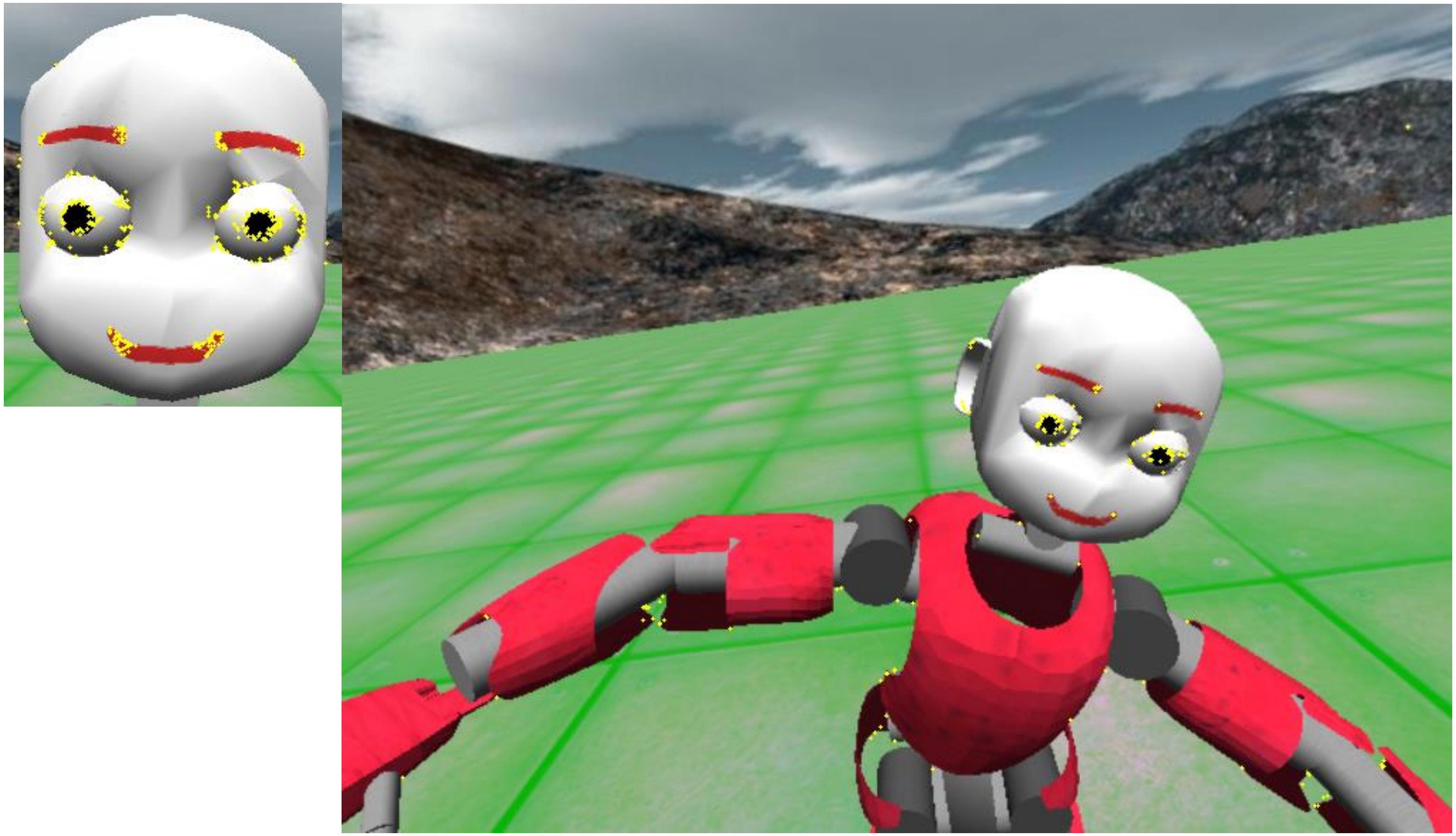
- Object is represented by something which corresponds to object image after application of all possible contrast modifications

could be extended to comply

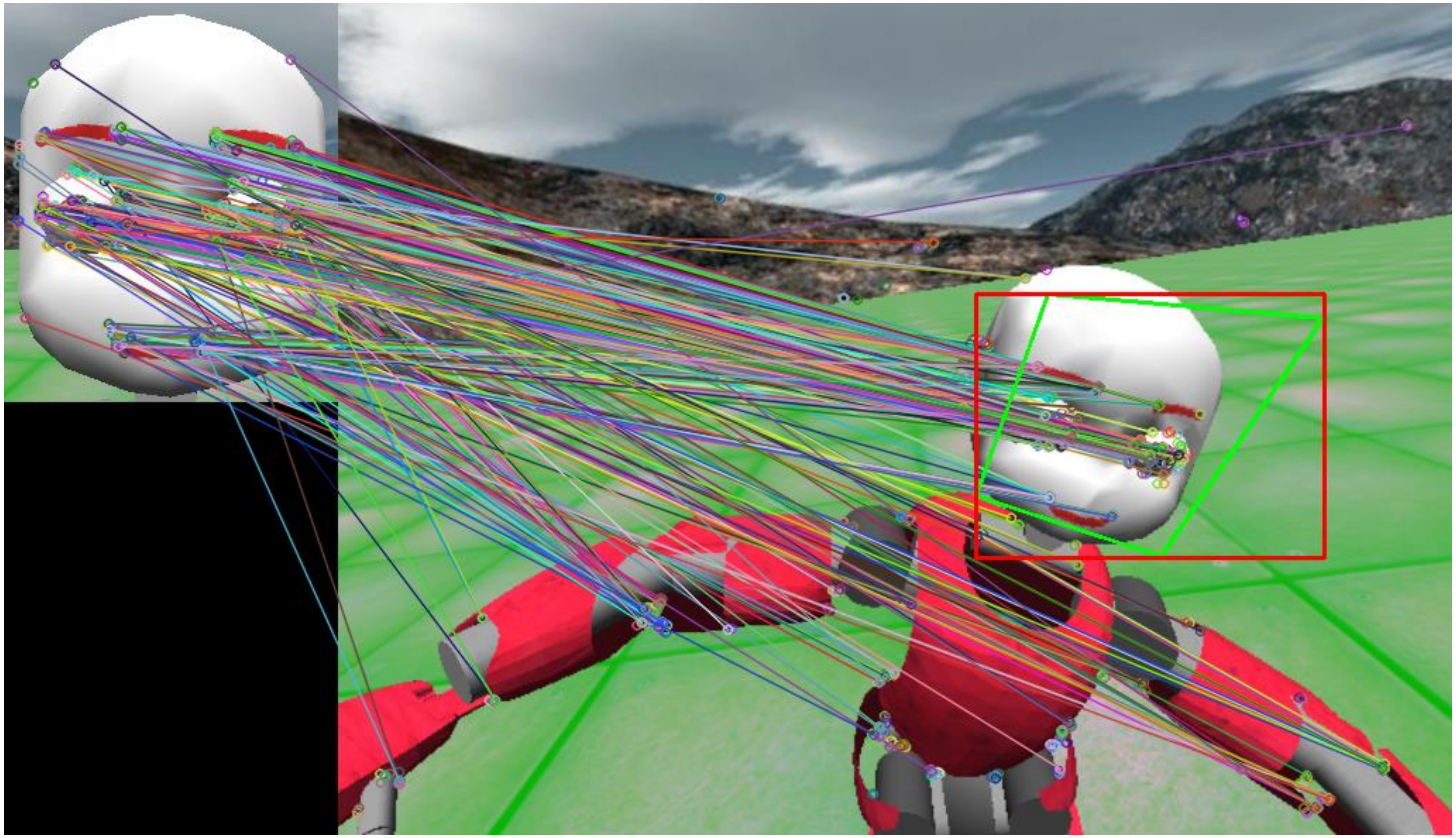


Feature detectors

Feature detector finds set of key points and describe their vicinity by descriptor in such a way that it is almost invariant to translation, scale and rotation



- Then matcher processes keypoint pairs with similar description on a template and an image and calculates a transform
(by the RANSAC algorithm)



Transforms

$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scaling matrix

$$\begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation matrix

$$\begin{bmatrix} s_x & s_{xy} \\ s_{yx} & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Linear transform

$$\begin{bmatrix} s_x & s_{xy} & t_x \\ s_{yx} & s_y & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

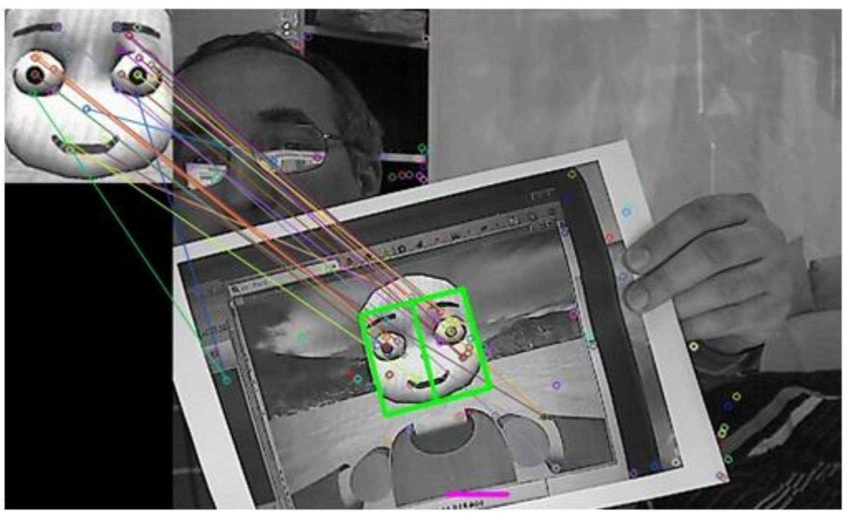
Affine transform

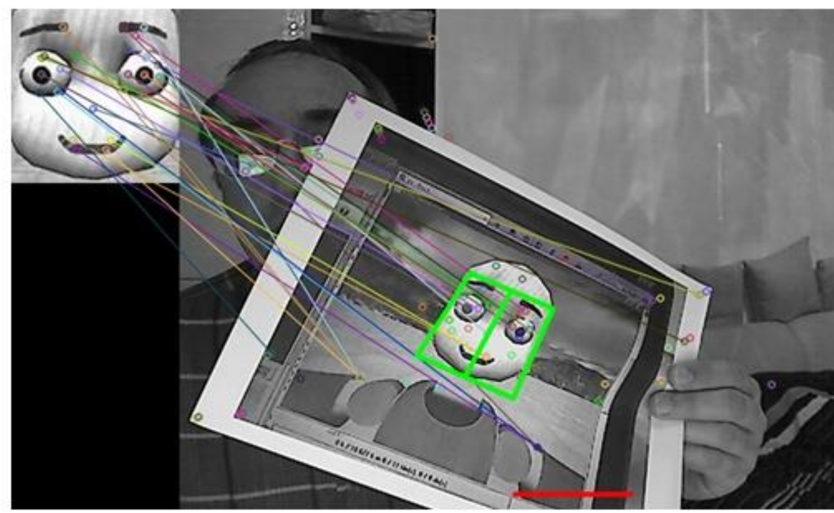
$$\begin{bmatrix} s_x & s_{xy} & t_x \\ s_{yx} & s_y & t_y \\ t_{yx} & t_{xy} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homography

Most known feature detectors

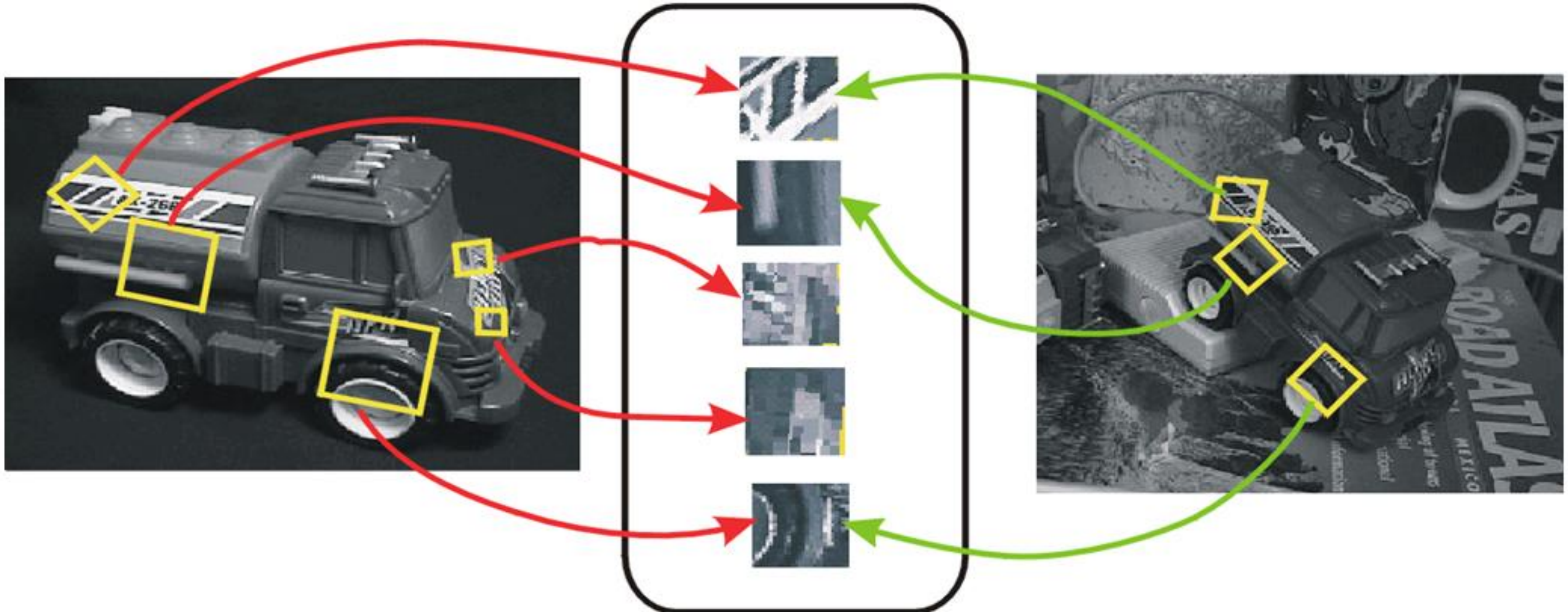
- SIFT (Scale Invariant Feature Transform)
- SURF (Speeded up Robust Features)
- ORB (Oriented FAST and Rotated BRIEF)



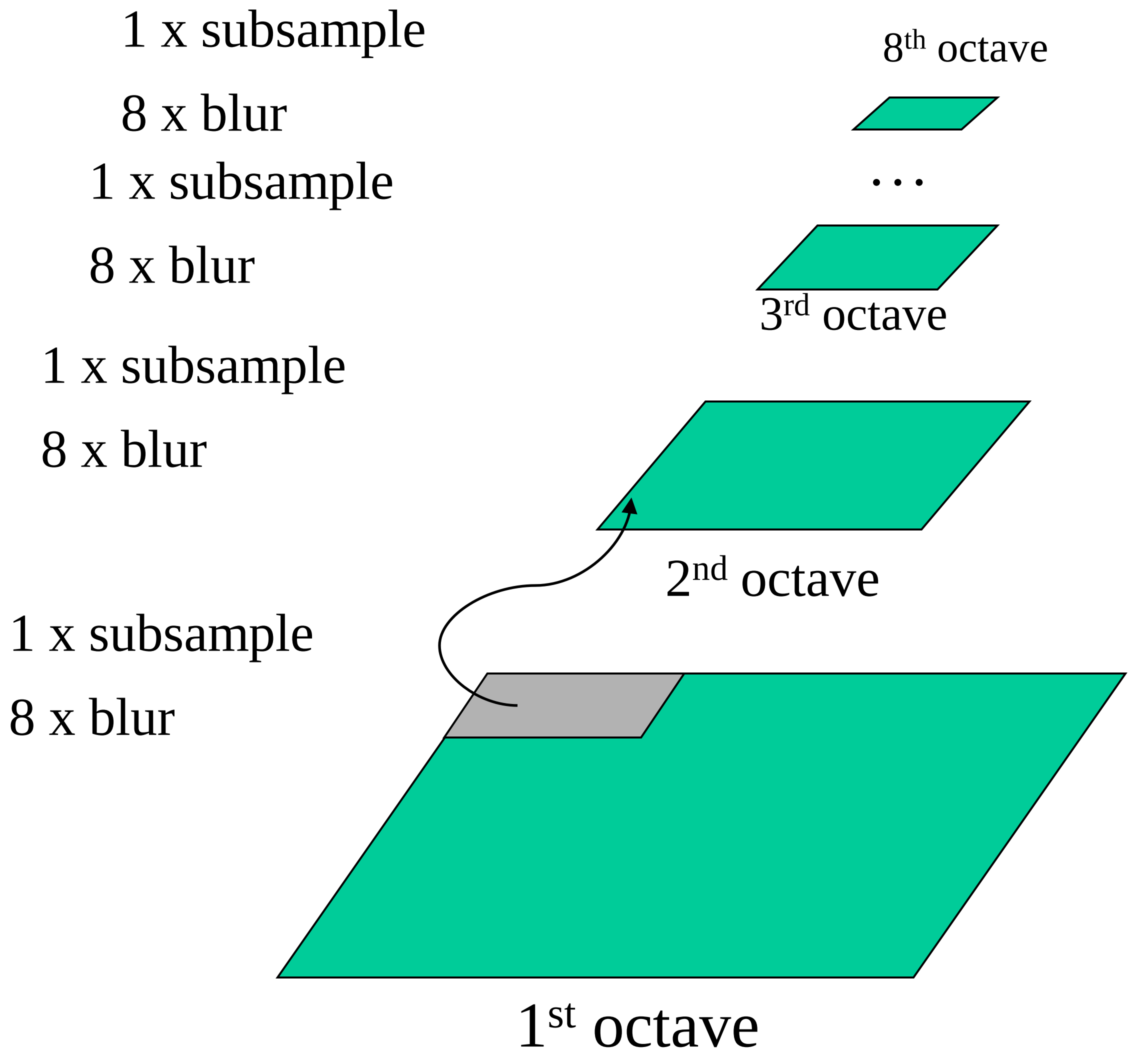


SIFT (Scale Invariant Feature Transform)

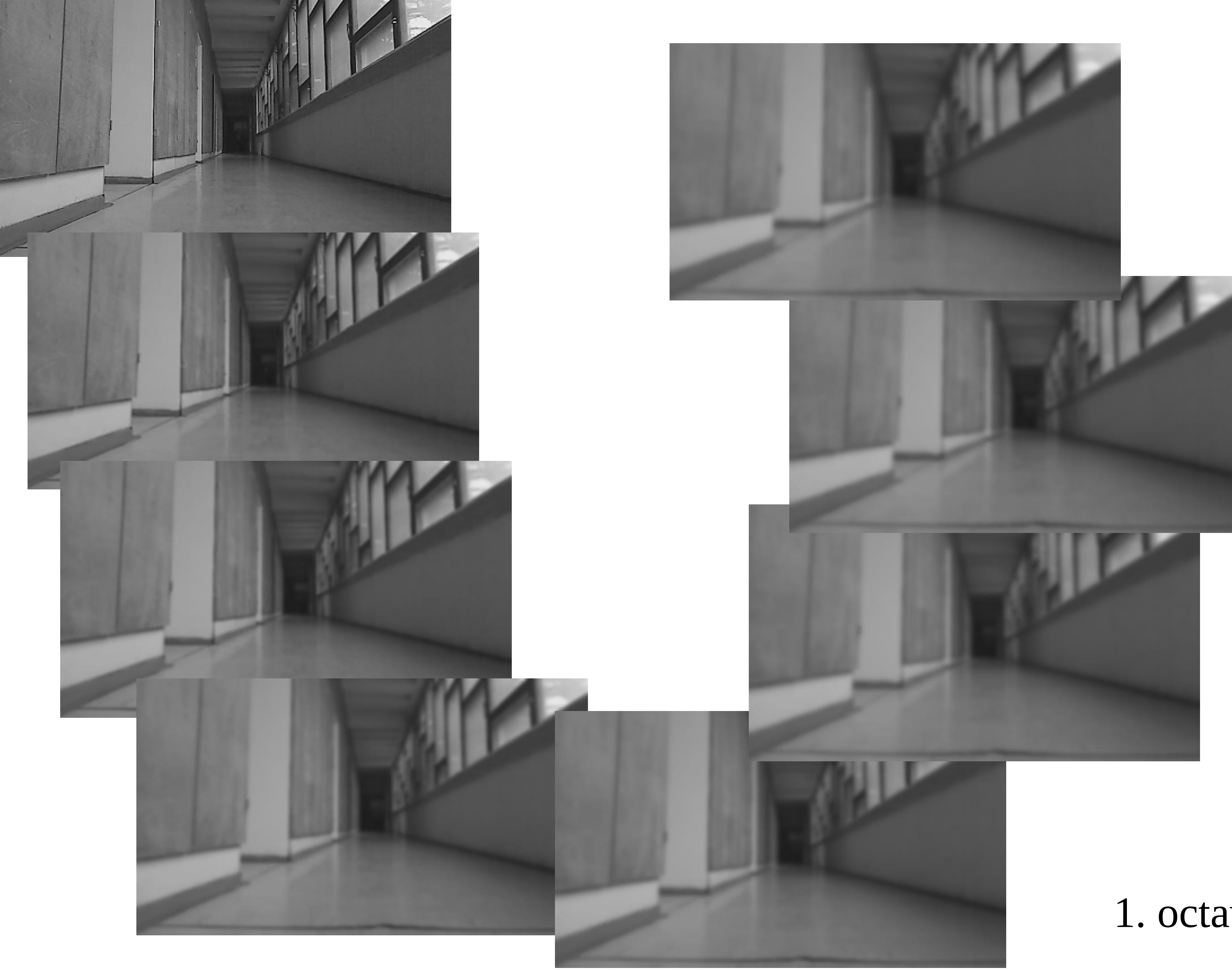
As a result we find points with similar descriptors on two images of the same object and some of them we can pair



SIFT (Scale Invariant Feature Transform)



Keypoints correspond to significant details lost in process of subsampling of the image

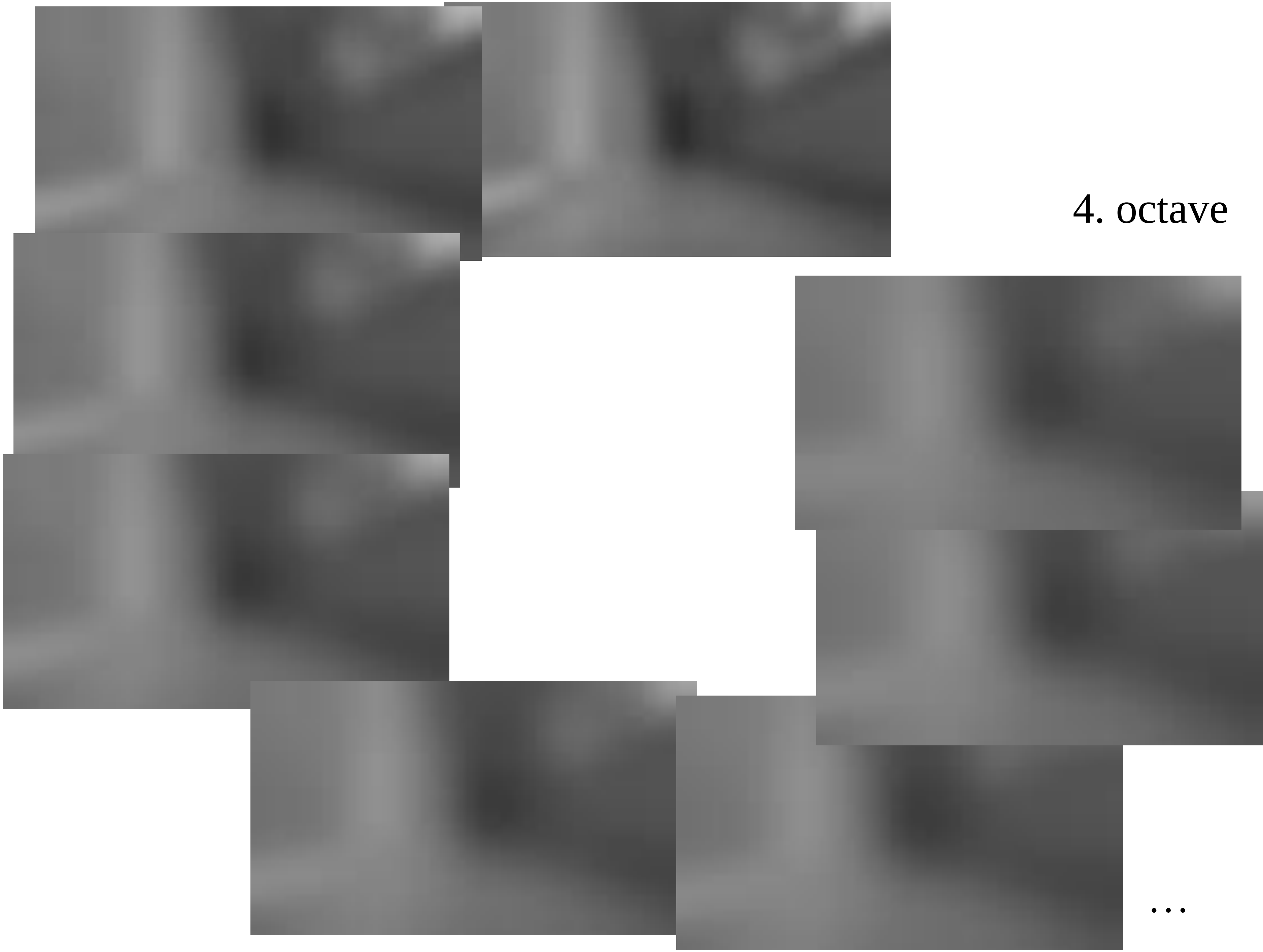


1. octave

2. octave

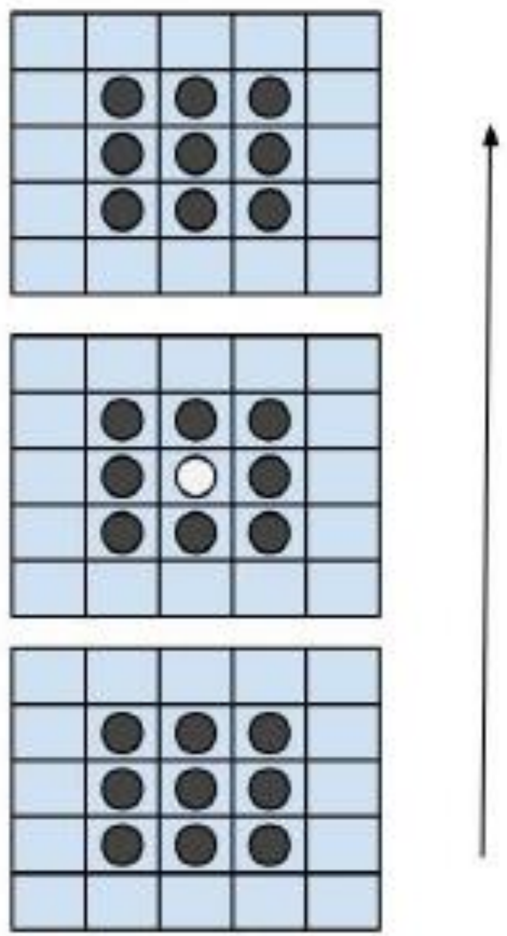
3. octave

4. octave



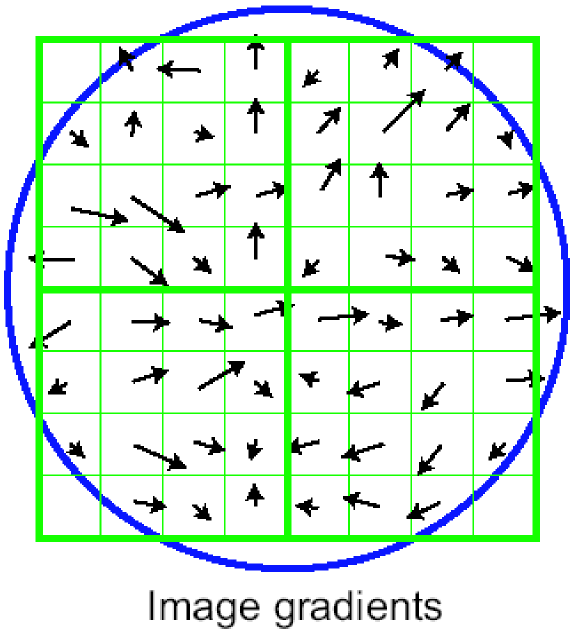
…

SIFT (Scale Invariant Feature Transform)



Keypoint has extremal intensity in 3D pyramid compounded from octaves

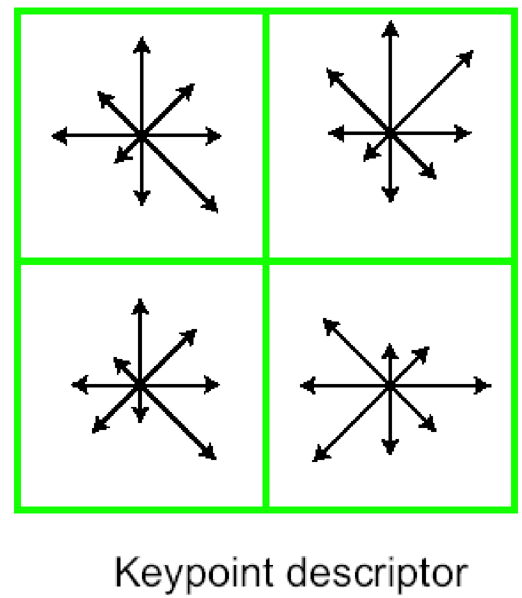
SIFT (Scale Invariant Feature Transform)



orientation

=

prevailing slope

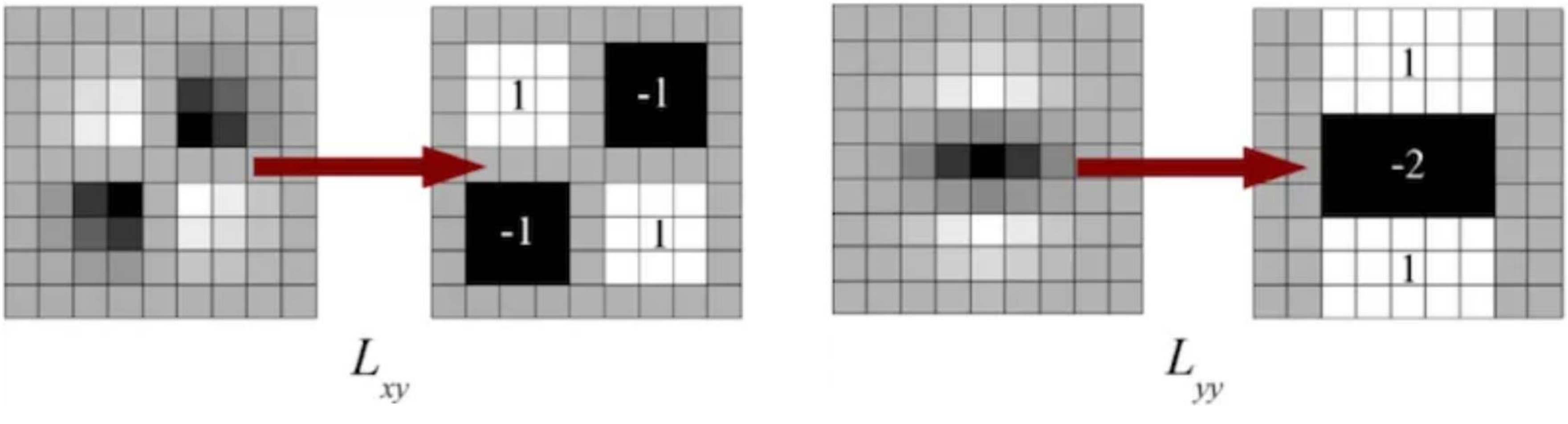


Each keypoint is associated with descriptor of its vicinity: normalized distribution of gradients in quadrants

Descriptors are compared by the Euclidean distance

SURF (Speeded up Robust Features)

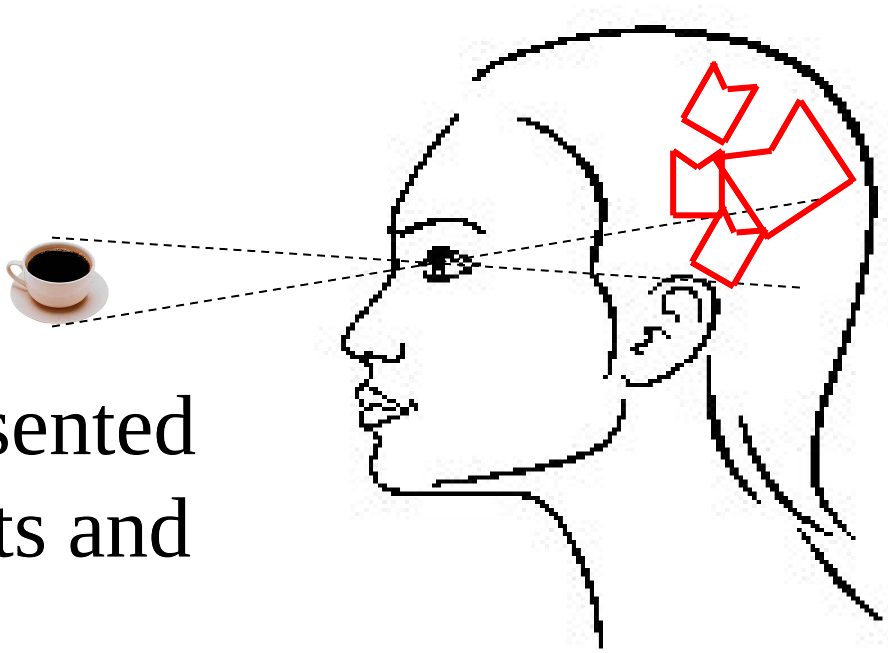
- a faster and less precise clone of SIFT, which employs an approximation of Gaussians that we can calculate by integral images



ORB (Oriented FAST and Rotated BRIEF)

- FAST – keypoint detector (detects corners by Harris detector and apply non-maximum suppression)
- BRIEF – keypoint descriptor (describes vicinity of the keypoint by a binary feature vector indicated comparison of the keypoint intensity and intensities of the points in its vicinity) called for several possible rotations from the orientation of keypoints and several scales
- Descriptors are compared by the Hamming distance

Feature detectors



- The object is represented by a set of keypoints and their descriptors
- Problem when object is not present

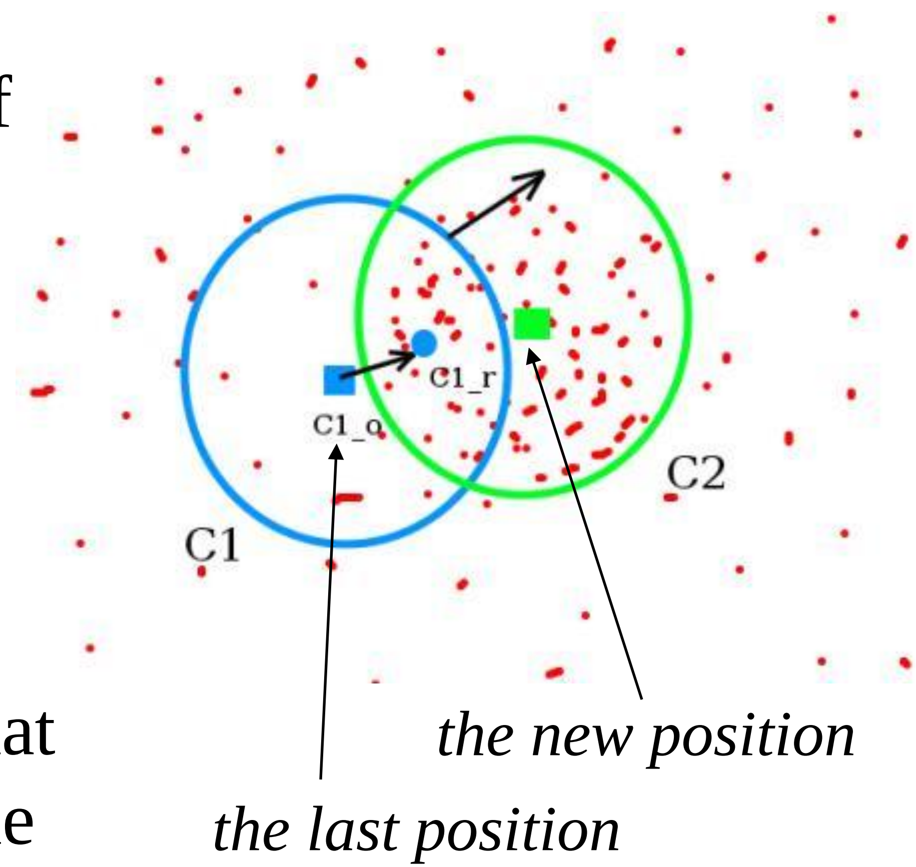
Trackers

- Detector process each image separately
- Tracker works with video and can employ information from previous images
- Simplest tracker: detector + outlier filtering (e.g. by Kalman filter)
- Advanced solutions: the tracker tries to move a window containing the object in such direction which keeps some features invariant CamShift, MIL

The color-based tracking

MeanShift

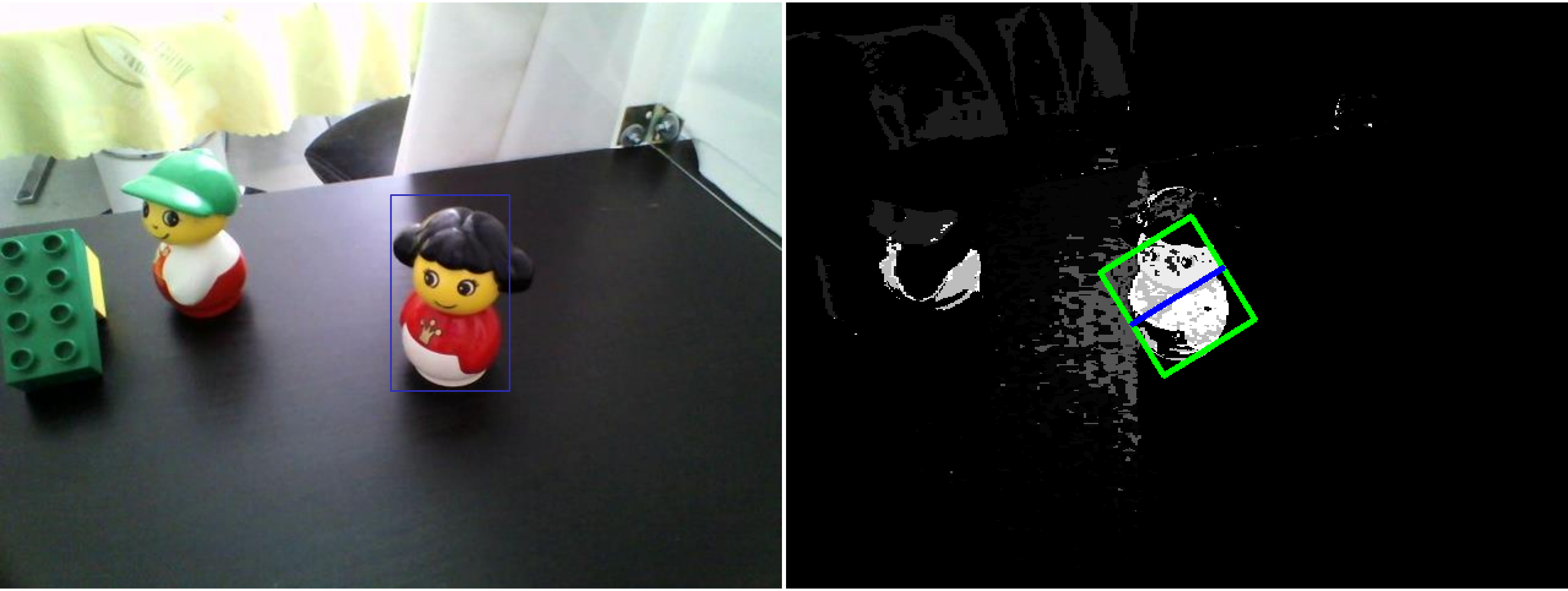
We have a set of points that belongs to an object. One of them represents the object's position. Within a given spatial radius, we calculate their center of mass and move position there. We repeat the process until reaching the fixed point. That is the updated position of the tracked object.



MeanShift / CamShift tracker

1. How can we define which points belong to an object? The Meanshift tracker calculates the color histogram of the given template and the rest of image.
2. Then, it calculates for each pixel of a given image the color distance between the pixel color and the histograms - the so-called back projection image.
3. Then, the Meanshift moves the tracking rectangle to a new position.
4. The Camshift also tries to rotate and scale it.

MeanShift / CamShift tracker

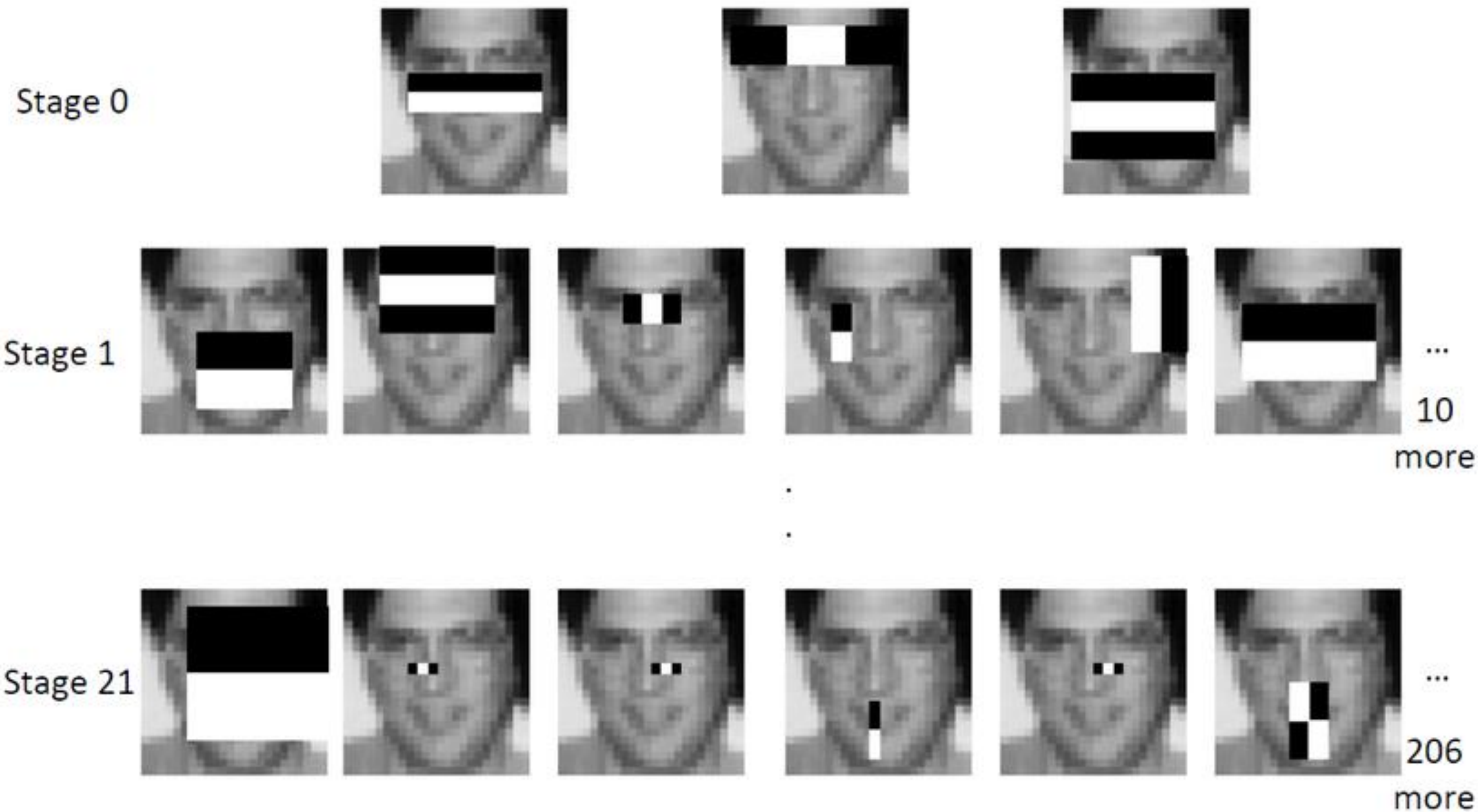


the backprojection image

The shape-based tracking

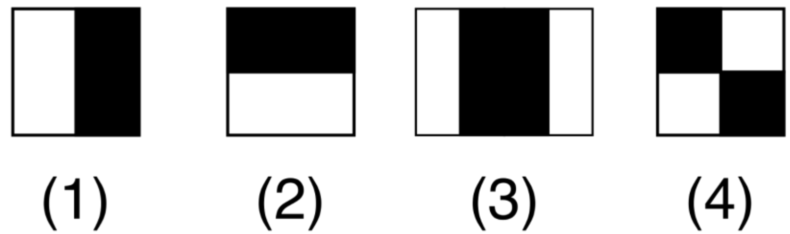
Haar Features

$$\sum c.\,\square < \sum c.\,\blacksquare \;?$$

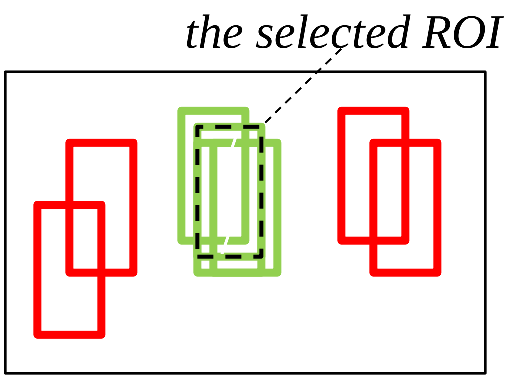


Multiply instance learning (MIL)

- Haar features



we put on ROI set of 2, 3 or 4 rectangles and calculate difference of white and black areas totals and compare them to zero

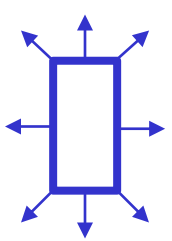


- Initial Region of Interest (ROI)

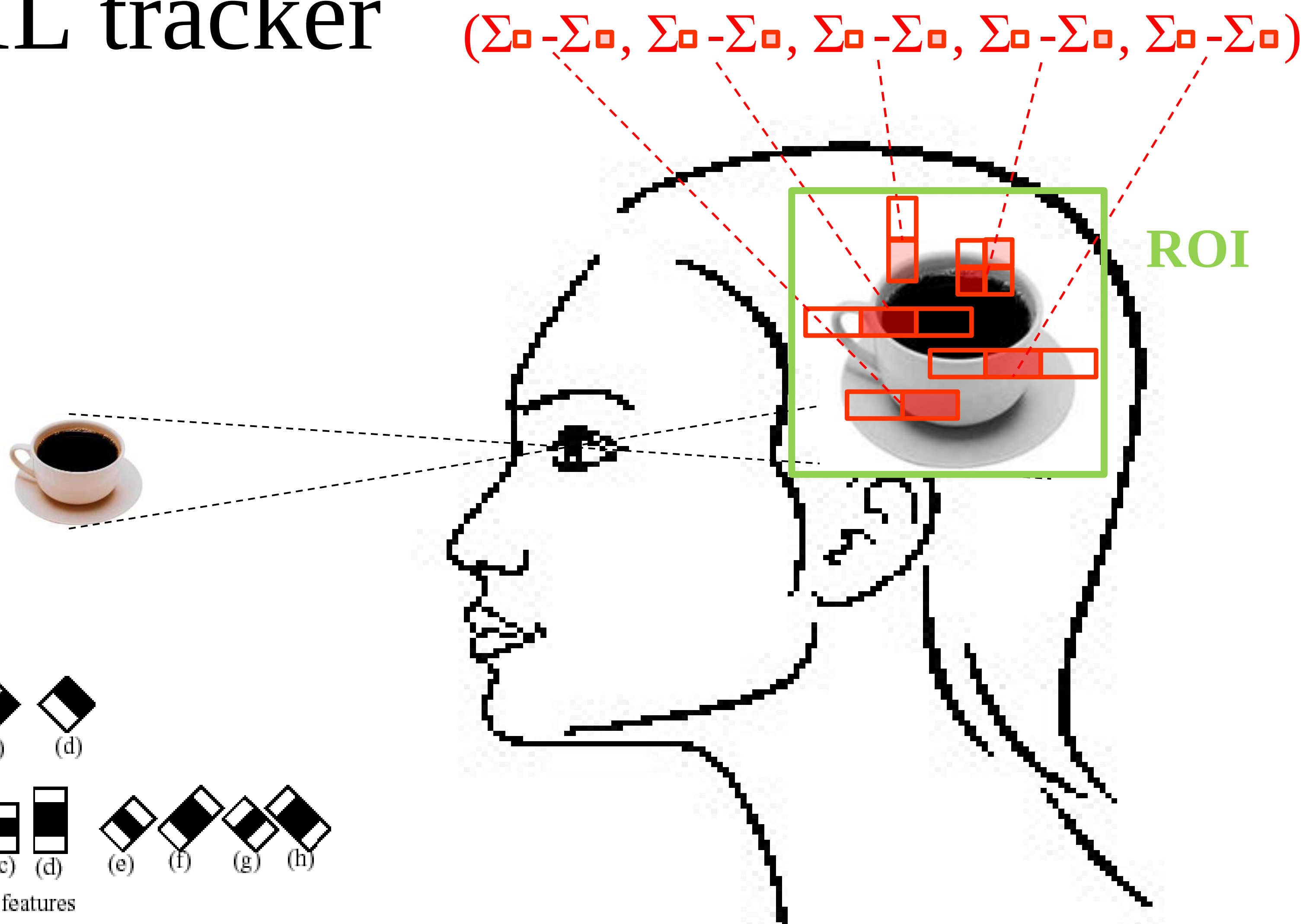
Then we teach Bayes classifier of bags of positive (ROIs close to the current ROI) and negative (far ROIs) examples

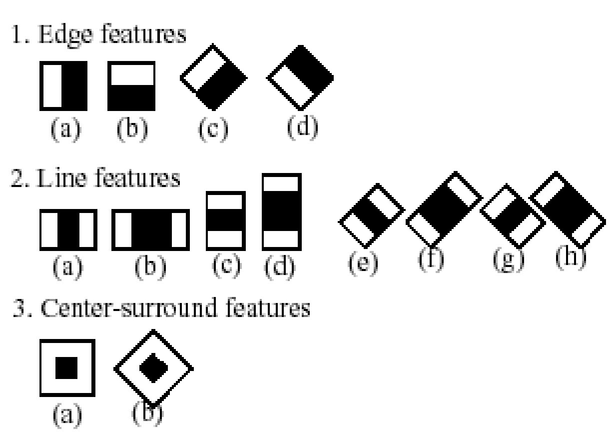
- Motion model: neighborough ROI

we move to neighborough ROI which has the highest ranging provided by the classifier



MIL tracker


(Σ□-Σ□, Σ□-Σ□, Σ□-Σ□, Σ□-Σ□, Σ□-Σ□)
ROI


1. Edge features
(a) (b) (c) (d)
2. Line features
(a) (b) (c) (d) (e) (f) (g) (h)
3. Center-surround features
(a) (b)