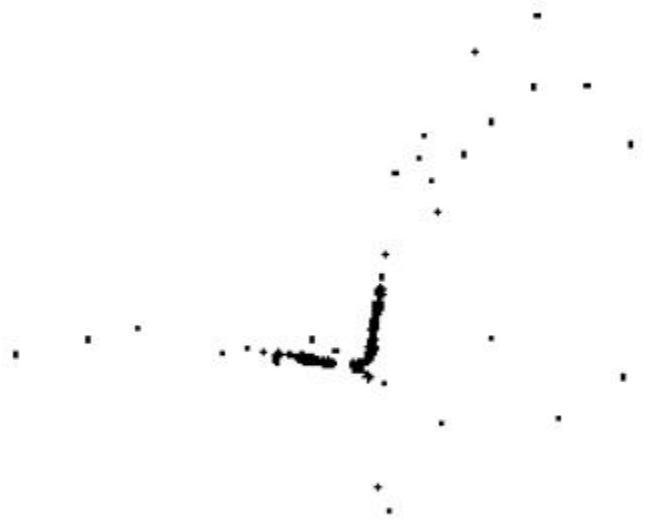
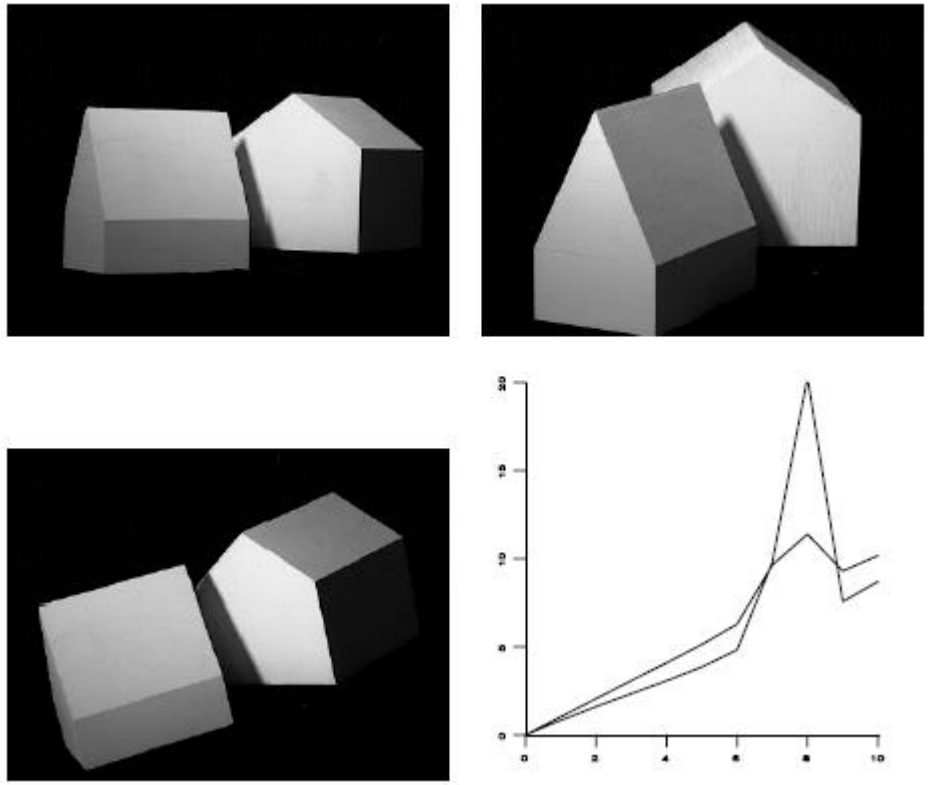
Lecture 4B – Feature detection and matching

Reconstruction circa 1994



Possibly the first Euclidean reconstruction with uncalibrated cameras.



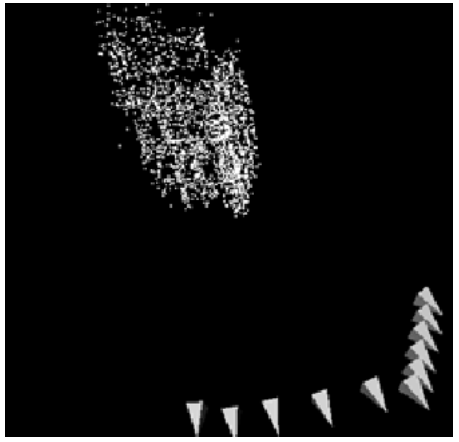
The trifocal tensor gave relationships between three views of an object.

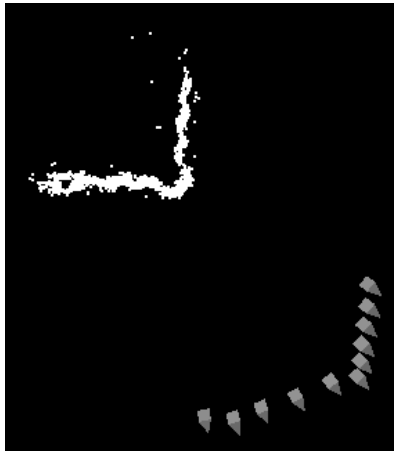
Image Stitching, 1994

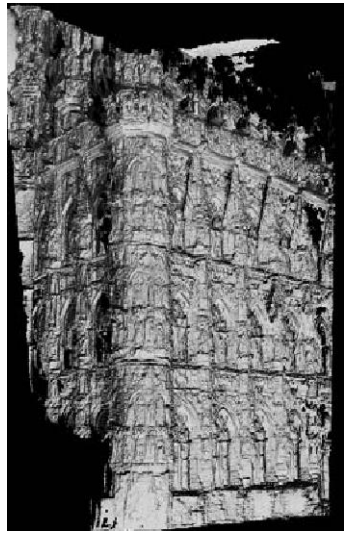
Scene Reconstruction and Model Building, ca 1999

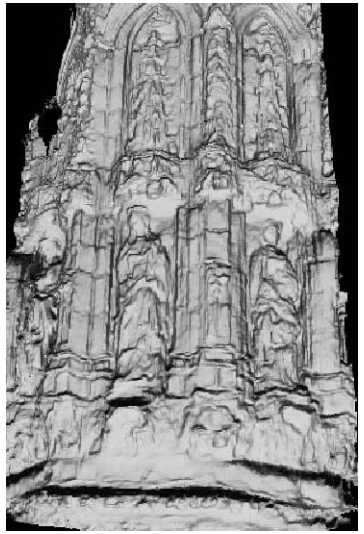
Courtesy Marc Pollefeys

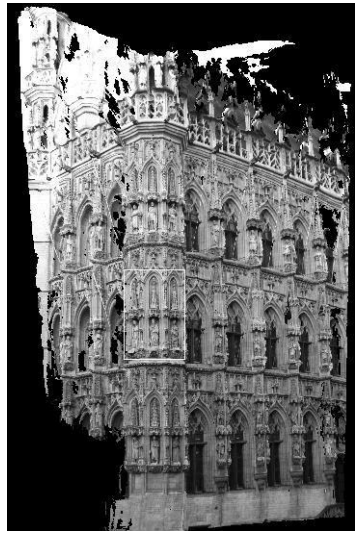
Courtesy Marc Pollefeys

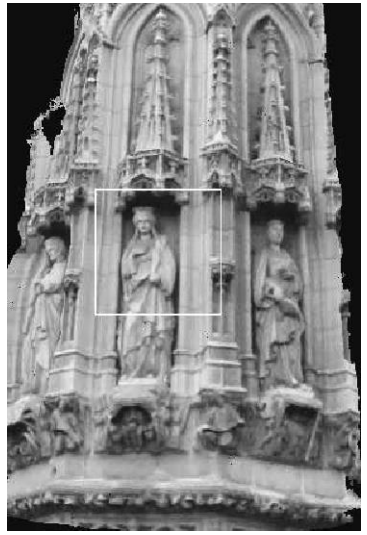






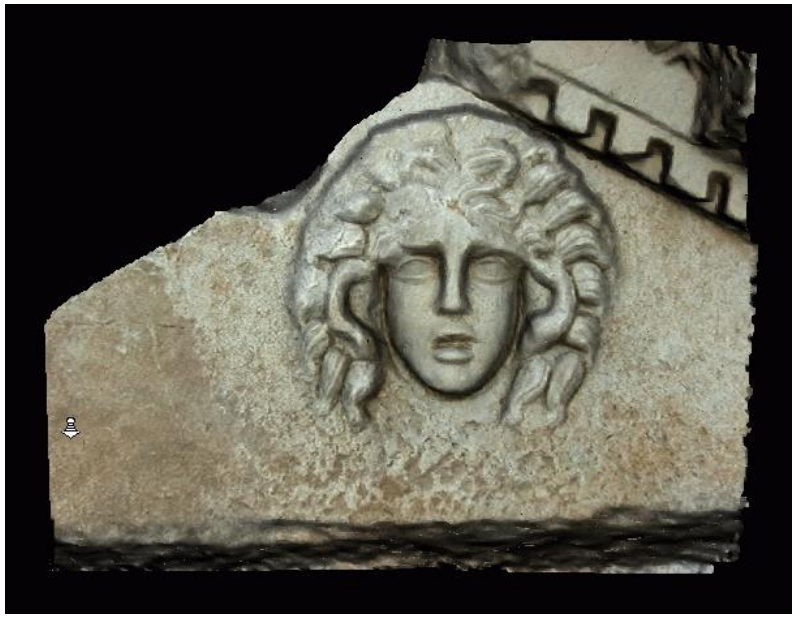






Example: DV video → 3D model

(Pollefeys et al. IJCV04)



Recorded at archaeological site of Sagalassos in Turkey

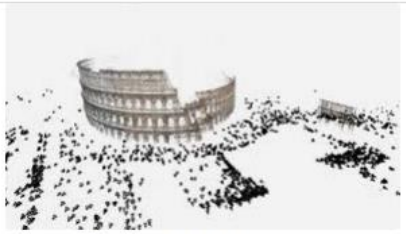
accuracy ~1/500 from DV video (i.e. 140kb jpegs 576x720)

3D reconstruction, circa 2006

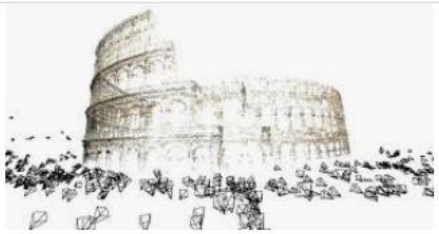
Building Rome in a Day – c 2011

Building Rome in a Day
grail.cs.washington.edu

Building Rome in a Day | O...
m-cacm.acm.org

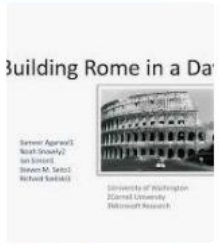


Rome was built in a day, with hundreds ...
washington.edu

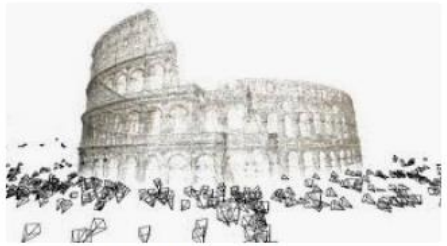


Building Rome in a Day
vision.cs.utexas.edu

Building Rome in a Day
grail.cs.washington.edu



PPT - Building Rome in a...
slideserve.com



Computers Turn Flat Photos into 3-D ...
nytimes.com

Building Rome in a Day
grail.cs.washington.edu

Building Rome in a Day Sameer A...
slideplayer.com

Building Rome in a Day
grail.cs.washington.edu

Building Rome In A Day (Computer Vision...
youtube.com

Building Rome on a Cloudless Day - YouTube
youtube.com

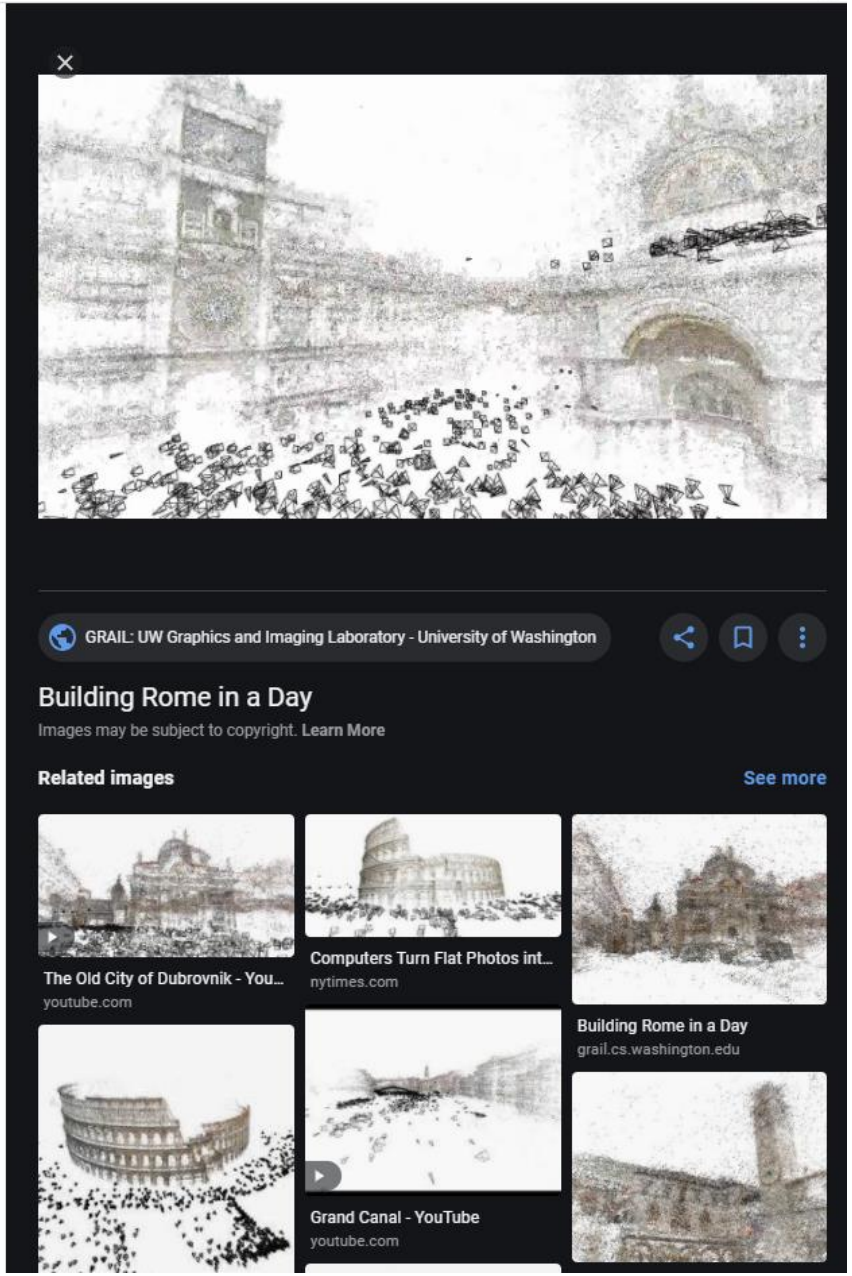
How to build Rome in a day
slideshare.net

Noah Snavely | MIT Technology Review
technologyreview.com

Building Rome in a Day
vision.cs.utexas.edu

Rome Wasn't Built in a Day
graylinerome.com

Building rome in_a_day_yas-nyan
slideshare.net



So what happened?

David Lowe

Computer Science Dept., University of British Columbia
Verified email at cs.ubc.ca - Homepage

Computer Vision Object Recognition

FOLLOW

TITLE

Distinctive in
DG Lowe
International jo

Object recog
DG Lowe
International C

Fast Approxi
M Muja, DG Lo
VISAPP (1) 2,

Automatic pa
M Brown, DG L
International Jo

Perceptual C
DG Lowe
Kluwer Academ

Three-dimer
DG Lowe
Artificial intelli

Recognising
M Brown, DG L
International C

Shape index
JS Beis, DG Lo
Computer Visio

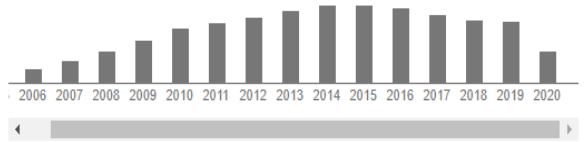
A boosted pa
K Okuma, A Tal
Computer Visio

Fitting paran
DG Lowe
IEEE Transacti

×

Distinctive image features from scale-invariant keypoints

Authors	David G Lowe
Publication date	2004/11/1
Journal	International journal of computer vision
Volume	60
Issue	2
Pages	91-110
Publisher	Springer Netherlands
Description	This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through ...
Total citations	Cited by 57911



Scholar articles Distinctive image features from scale-invariant keypoints
DG Lowe - International journal of computer vision, 2004
Cited by 57911 Related articles All 168 versions

Local Grayvalue Invariants for Image Retrieval

Cordelia Schmid, Roger Mohr

$$\exp^{-\frac{x^2+y^2}{2\sigma^2}} \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

The eigenvectors of this matrix are the principal curvatures of the auto-correlation function. Two significant values indicate the presence of an interest point.

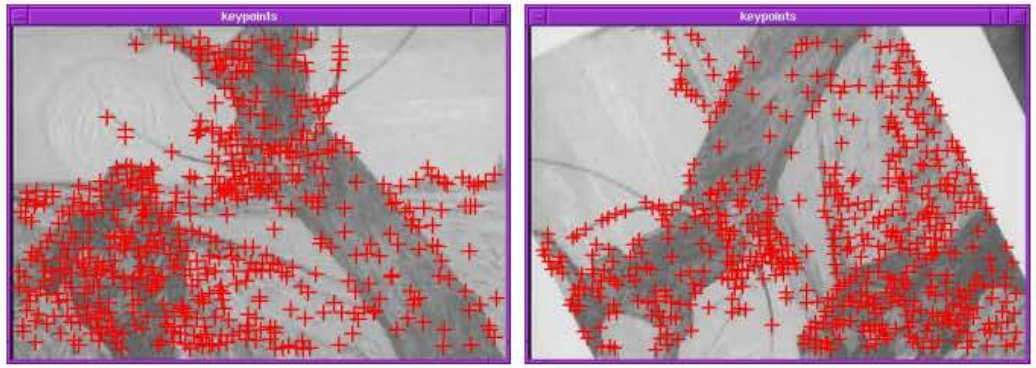


Fig. 2. Interest Points detected on the same scene under rotation. The image rotation between the left image and the right image is 155 degrees. The repeatability rate is 92%.

Figure 2 shows interest points detected on the same scene under rotation. The repeatability rate is 92% which means that 92% of the points detected in the first image are detected in the second one. Experiments with images taken under different conditions show that the average repeatability rate is about 90%. Moreover, 50% repeatability is sufficient for the remaining process if we use robust methods.

B. Complete set of differential invariants

In order to obtain invariance under the group $SO(2)$, differential invariants are computed from the local jet. Differential invariants have been studied theoretically by Kœnderink [20] and Romeny et al.[28], [29], [30]. A complete set of invariants can be computed that locally characterises the signal. The set of invariants used in this work is limited to third order. This set is stacked in a vector, denoted by $\mathcal{V}$. In equation 1 vector $\mathcal{V}$ is given in tensorial notation – the so-called Einstein summation convention. Notice that the first component of $\mathcal{V}$ represents the average luminance, the second component the square of the gradient magnitude and the fourth the Laplacian.

$$\mathcal{V}[0..8] = \begin{bmatrix} L \\ L_i L_i \\ L_i L_{ij} L_j \\ L_{ii} \\ L_{ij} L_{ji} \\ \varepsilon_{ij}(L_{jkl} L_i L_k L_l - L_{jkk} L_i L_l L_l) \\ L_{iij} L_j L_k L_k - L_{ijk} L_i L_j L_k \\ -\varepsilon_{ij} L_{jkl} L_i L_k L_l \\ L_{ijk} L_i L_j L_k \end{bmatrix} \quad (1)$$

with L_i being the elements of the "local jet" and ε_{ij} the 2D antisymmetric Epsilon tensor defined by $\varepsilon_{12} = -\varepsilon_{21} = 1$ and $\varepsilon_{11} = \varepsilon_{22} = 0$.

SIFT:

Scale Invariant Feature Transform

Overview

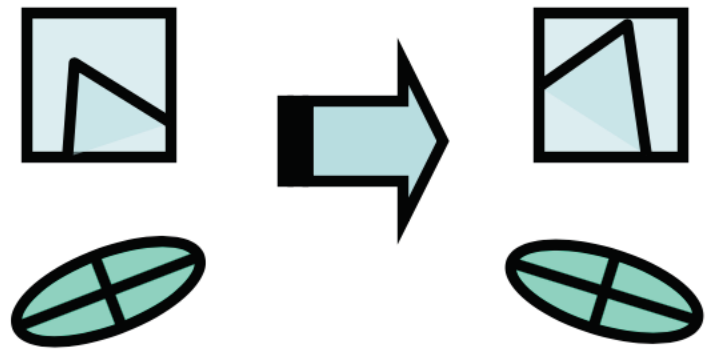
- Motivation for SIFT
- SIFT Feature Detector
- SIFT Descriptor
- Application

Overview

- Motivation for SIFT
- SIFT Feature Detector
- SIFT Descriptor
- Application

Harris Corner: Properties

- Harris corner is rotation-invariant.

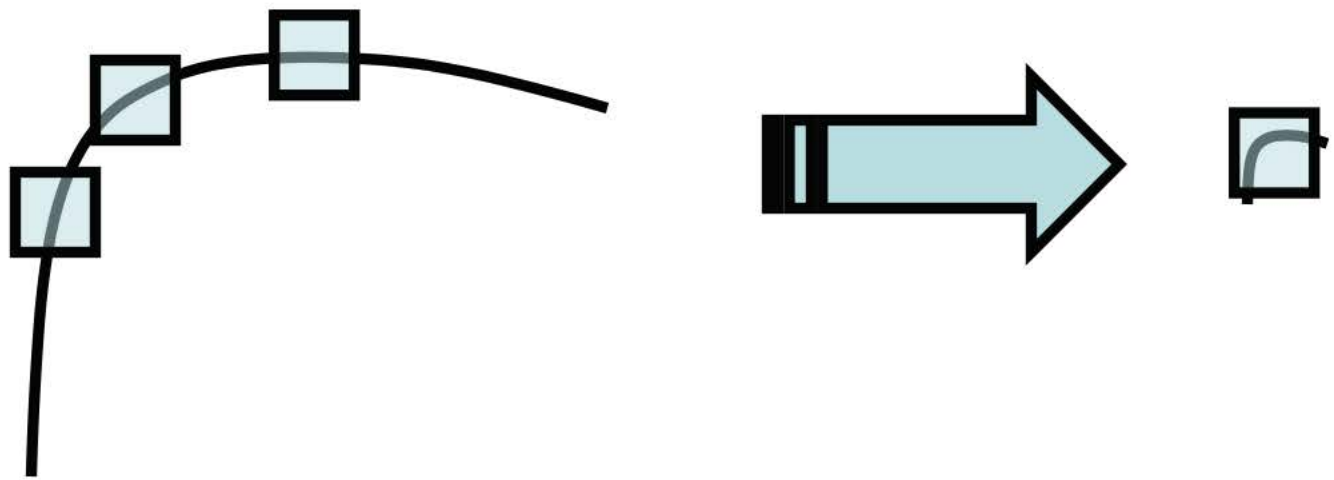


Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response R is invariant to image rotation

Harris Corner: Properties

- But: it is **not invariant** to *image scale change* !



All points will be classified as edges

Corner !

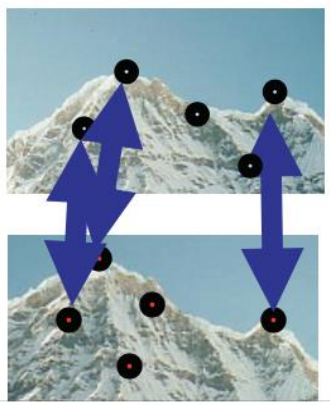
Example of scale change

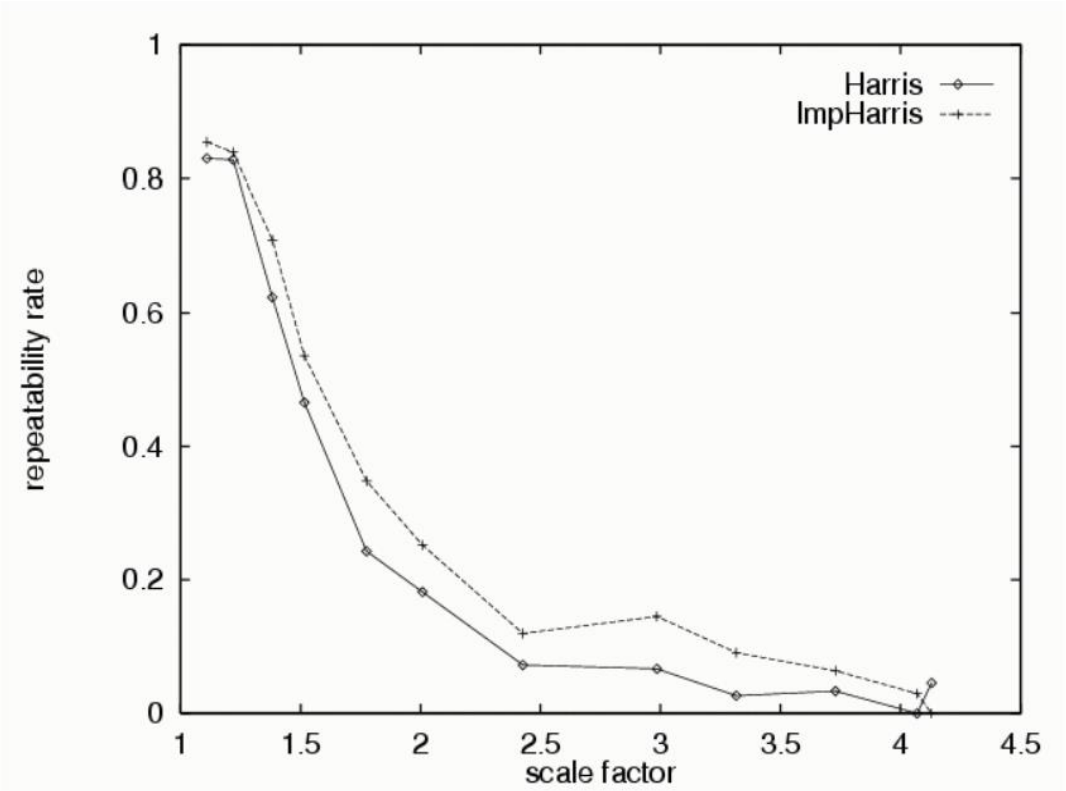
From left to right: Reference image, scale change for middle one is 1.5, scale change for right one is 4.1.

Harris Corner is not *Scale Invariant*

ε Repeatability rate:

$$\frac{\#\text{ correspondences}}{\#\text{ possible correspondences}}$$





Imp.Harris is a variant of Harris corner detector, which uses derivative of Gaussian instead of standard template used by Harris et al

C.Schmid et.al. "Evaluation of Interest Point Detectors". IJCV 2000

How to adapt to scale change ?

Goal

- To detect the same interest points regardless of image scale changes

How to adapt to scale change ?

- David Lowe's ***SIFT**** is a very efficient implementation of scale invariant distinctive image features detector.

*D. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60 (2), pp. 91-110, 2004.

SIFT: Motivation

- The Harris operator is not invariant to scale change.
- For better (more reliable, robust) image matching, Lowe's goal was to develop an interest operator that is invariant to scale and rotation.
- Also, Lowe aimed to create a descriptor that is robust to the variations in images, corresponding to typical viewing conditions.

Advantages of SIFT

- **Locality:** features are local, so robust to occlusion and clutter (no prior segmentation)
- **Distinctiveness:** individual features can be matched to a large database of objects
- **Quantity:** many features can be generated for even small objects
- **Efficiency:** close to real-time performance
- **Extensibility:** can easily be extended to wide range of differing feature types, with each adding robustness

Overall Procedure at a High Level

1. Scale-space extrema detection

   Search over *multiple scales* and *image locations*.

2. Keypoint localization

   Fit a model to determine location and scale. Select keypoints based on a measure of stability.

3. Orientation assignment

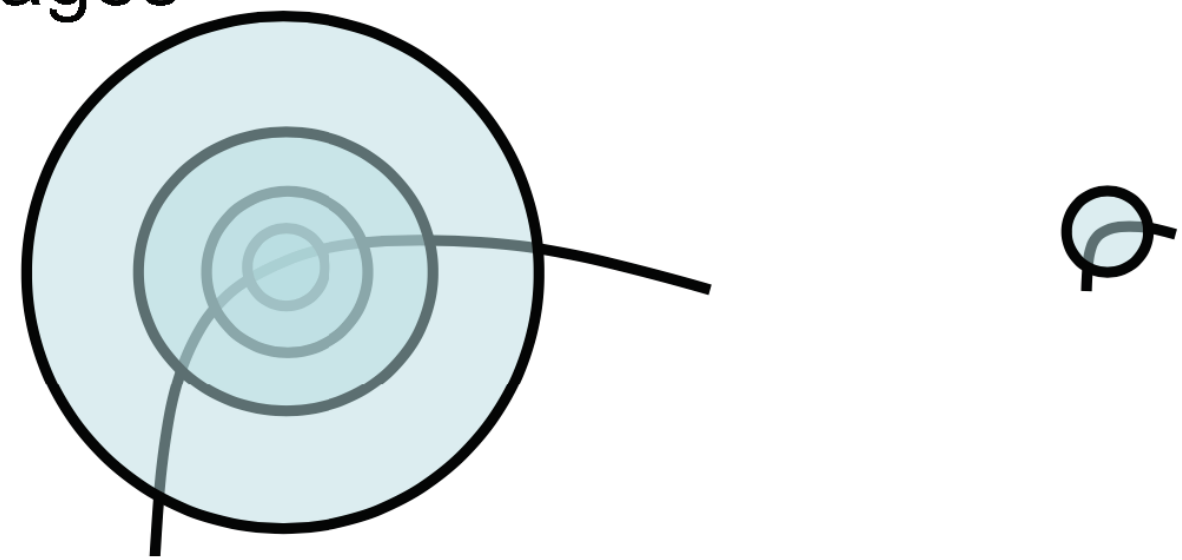
   Compute best orientation(s) for each keypoint region.

4. Keypoint description

   Use local image gradients at selected scale and rotation to describe each keypoint region.

Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes (scales) around a point
- Regions of corresponding (suitable) sizes (‘optimal scales’) will *look the same* in both images



Automatic Scale Selection principle [Lindeberg98*]

- Optimal scale selection principle:

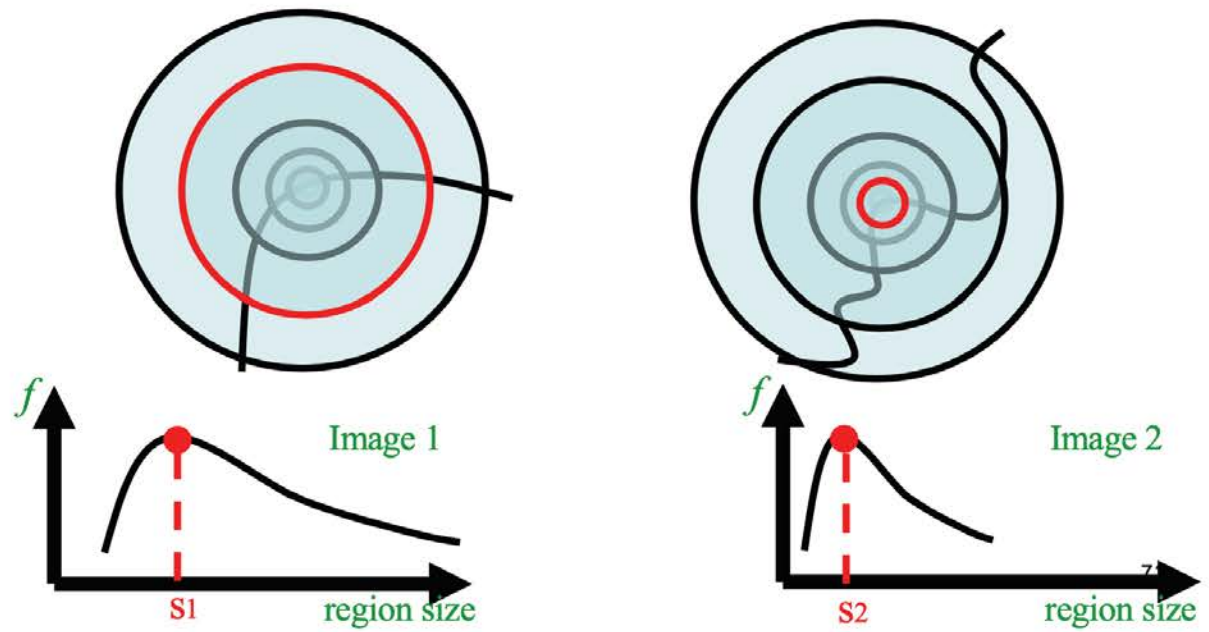
  – The scale at which ‘some function’ (e.g. the normalized derivative) assumes an extreme value indicates a feature containing interesting pattern/structure.

*T. Lindeberg, "Feature detection with automatic scale selection," International Journal of Computer Vision 30 (2), pp. 77-116, 1998.

Automatic Scale Selection

- **Principle:**

Find scale that gives local maxima of some function *f* in both spatial position and scale-space.



- What is this ‘signature function’ f ?

Scale Invariant Function

- Functions for determining scale

$$f = \text{Kernel} * \text{Image}$$

Kernels:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

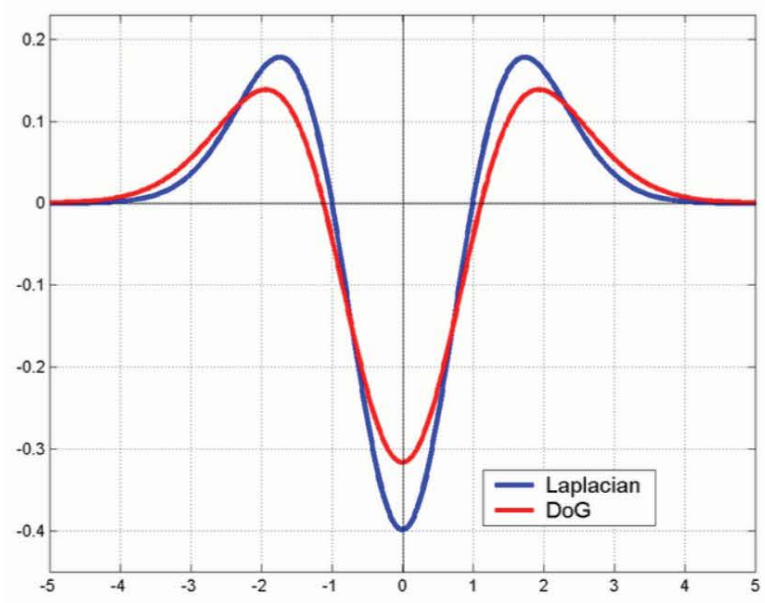
(Laplacian of Gaussian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

where Gaussian

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

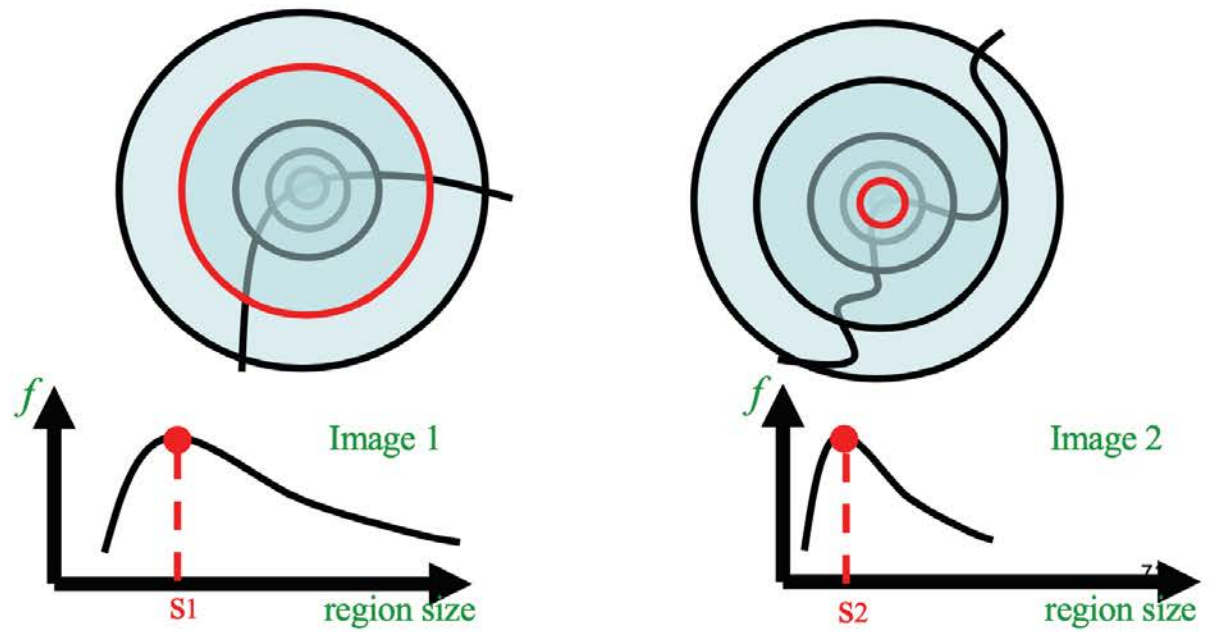


Note: both Kernels are invariant to *scale* and *rotation*

Recall: Automatic Scale Selection

- **Principle:**

Find scale that gives local maxima of some function *f* in both spatial position and scale-space.



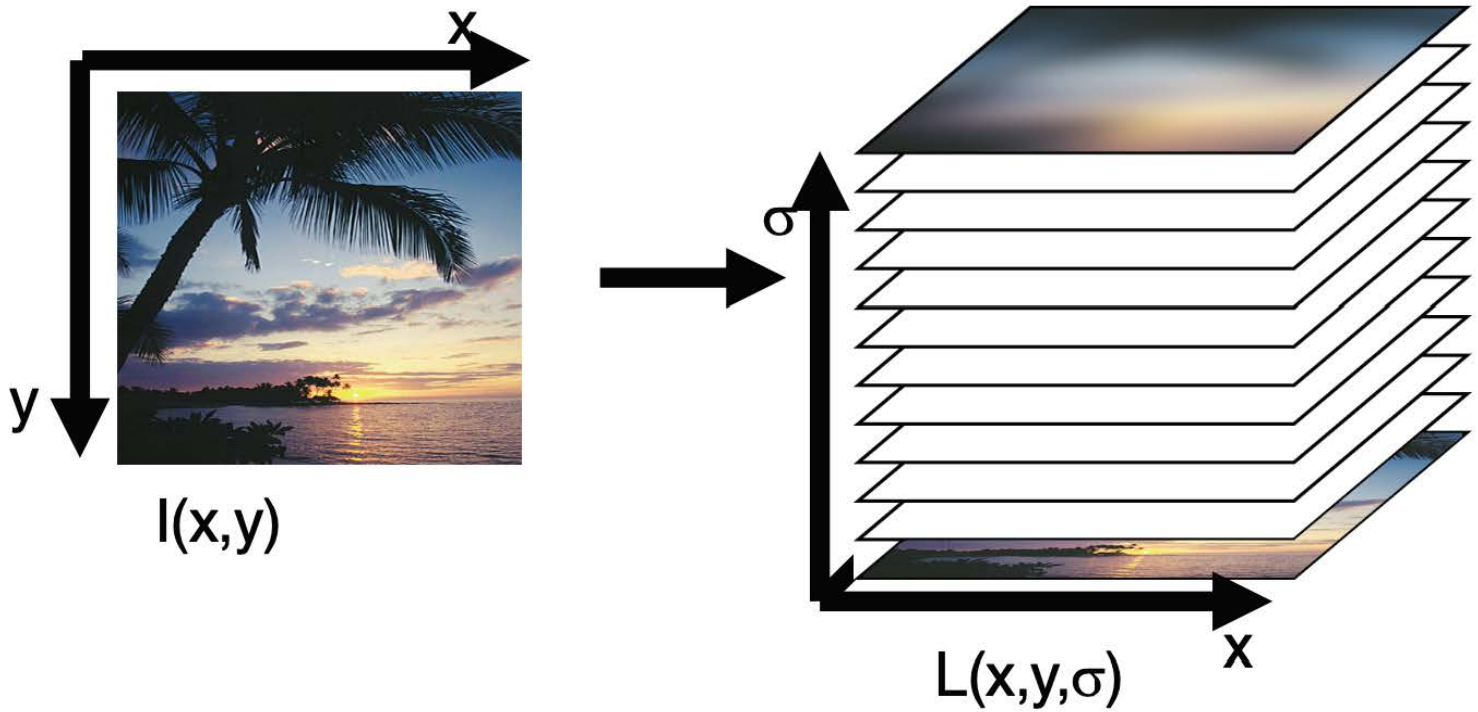
Scale space representation

$$L: R^2 \times R \rightarrow R$$

$$L(x, y; \sigma) = G(\sigma) * I(x, y)$$

- L is the scale space representation of I(x,y).
- L is obtained by smoothing I with Gaussian kernel G(σ).
- G(σ) is the natural choice for building up a scale space.

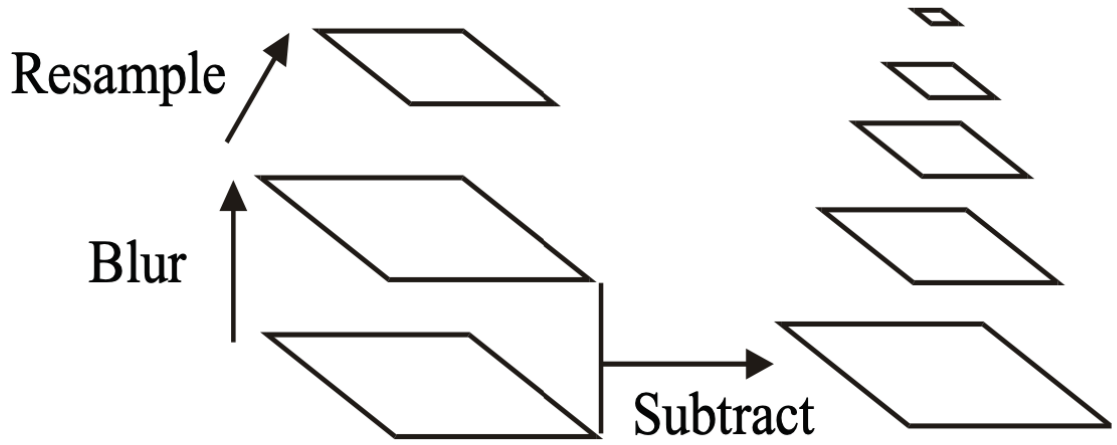
Scale space representation



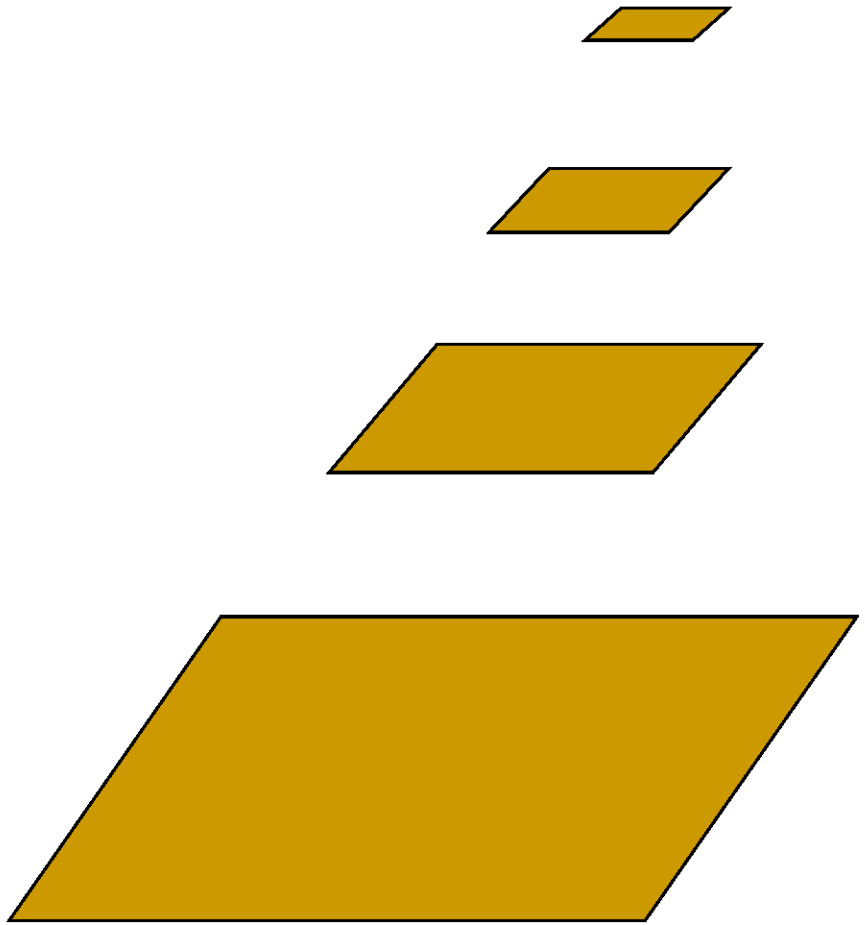
Scale-space extremum indicates the occurrence of the “optimal scale”

Build Scale-Space Pyramid

- All scales are examined to identify scale-invariant features.
- An efficient function is to compute the Difference of Gaussian (DOG) pyramid.



Aside: Image Pyramids



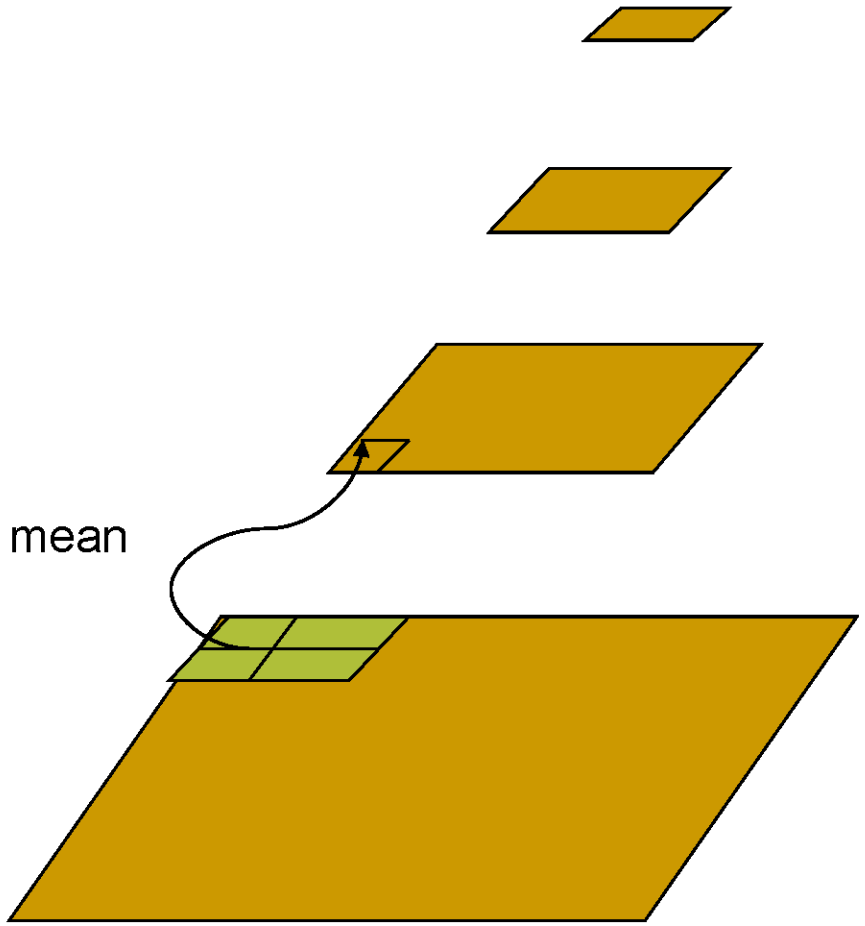
And so on.

3rd level is derived from the 2nd level according to the same funtion

2nd level is derived from the original image according to some function

Bottom level is the original image.

Aside: Mean Pyramid



And so on.

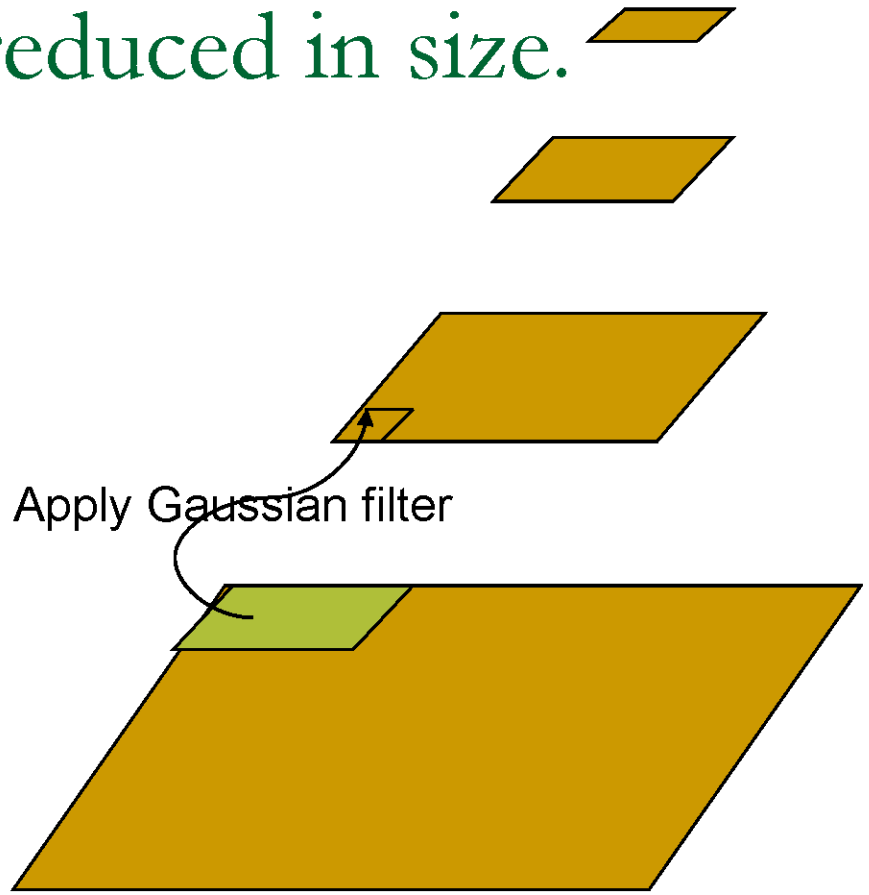
At 3rd level, each pixel is the mean of 4 pixels in the 2nd level.

At 2nd level, each pixel is the mean of 4 pixels in the original image.

Bottom level is the original image.

Aside: Gaussian Pyramid

At each level, image is smoothed and reduced in size.

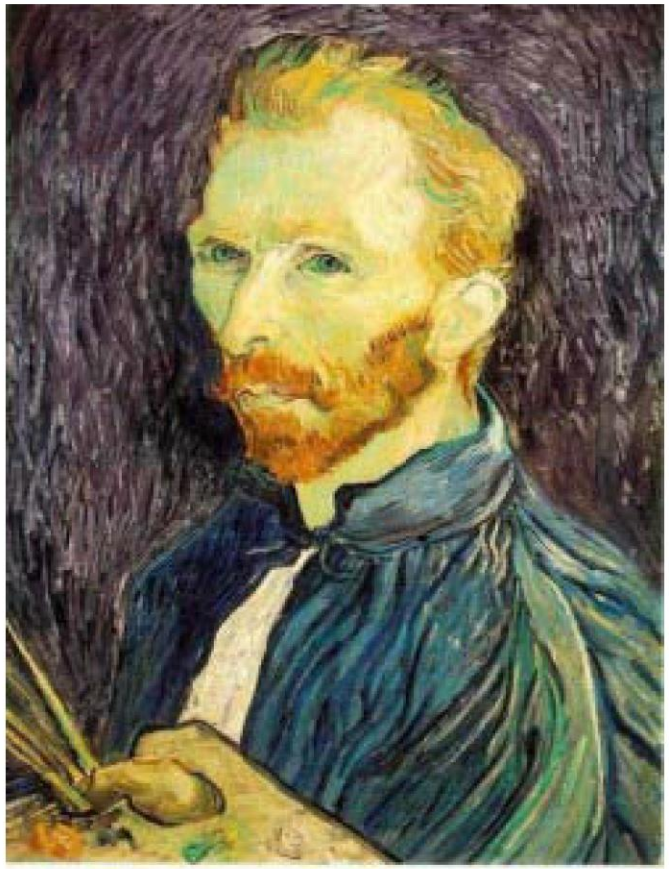


And so on.

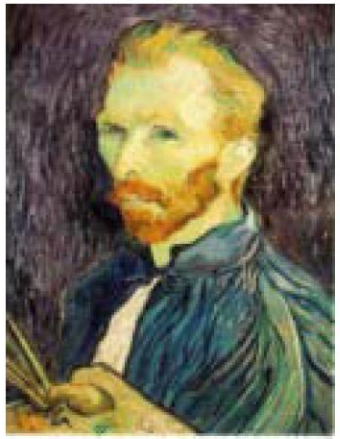
At 2nd level, each pixel is the result of applying a Gaussian mask to the first level and then subsampling to reduce the size.

Bottom level is the original image.

Example: Subsampling with Gaussian pre-filtering



Gaussian 1/2

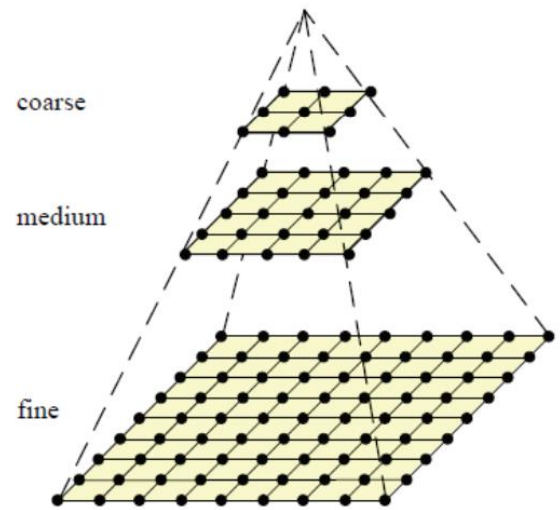


G 1/4

G 1/8

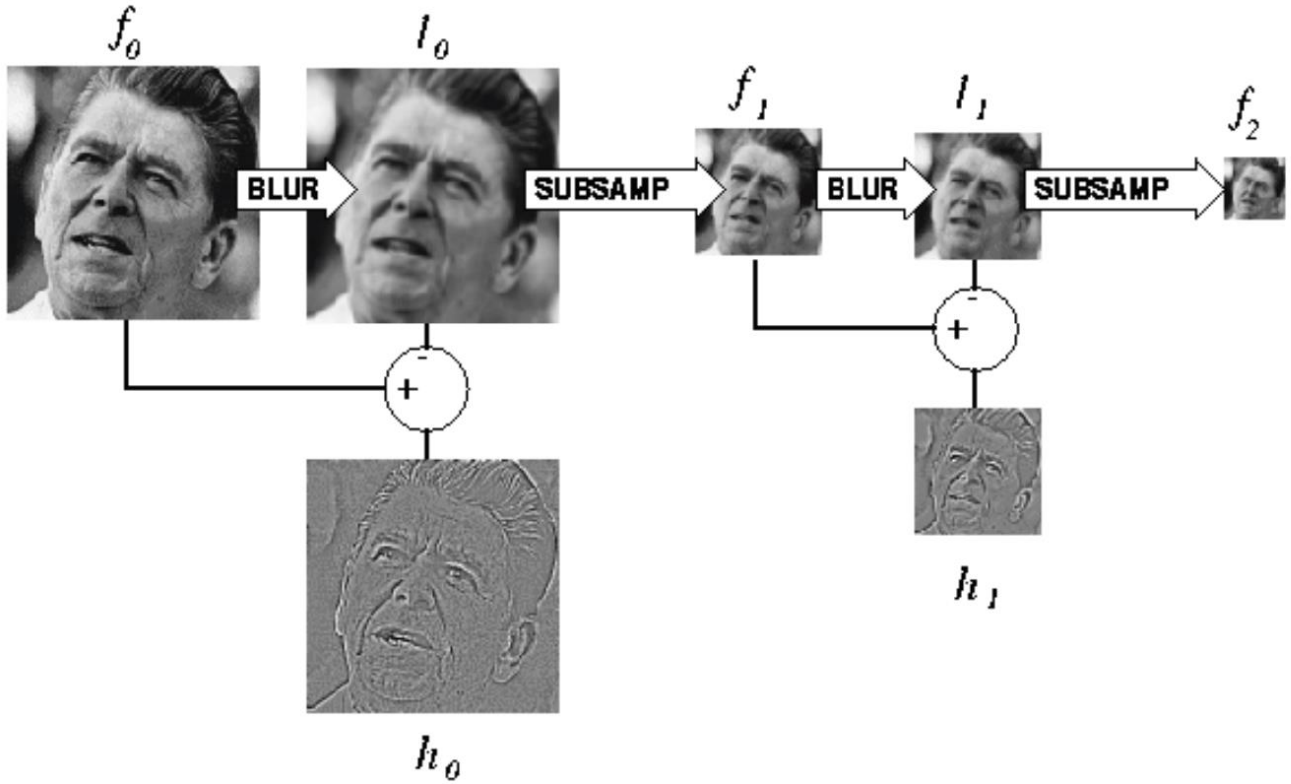
Gaussian Pyramid

512 256 128 64 32 16 8



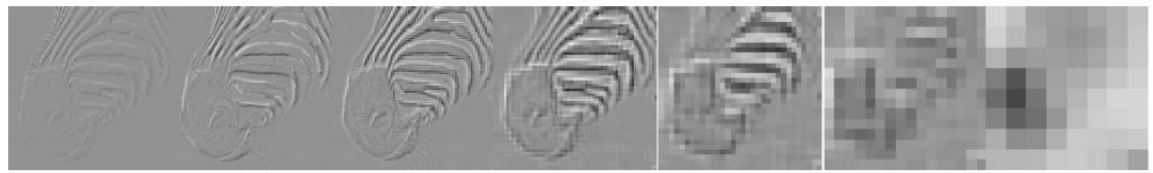
From **Forsyth**

Compute Gaussian and Laplacian Pyramid

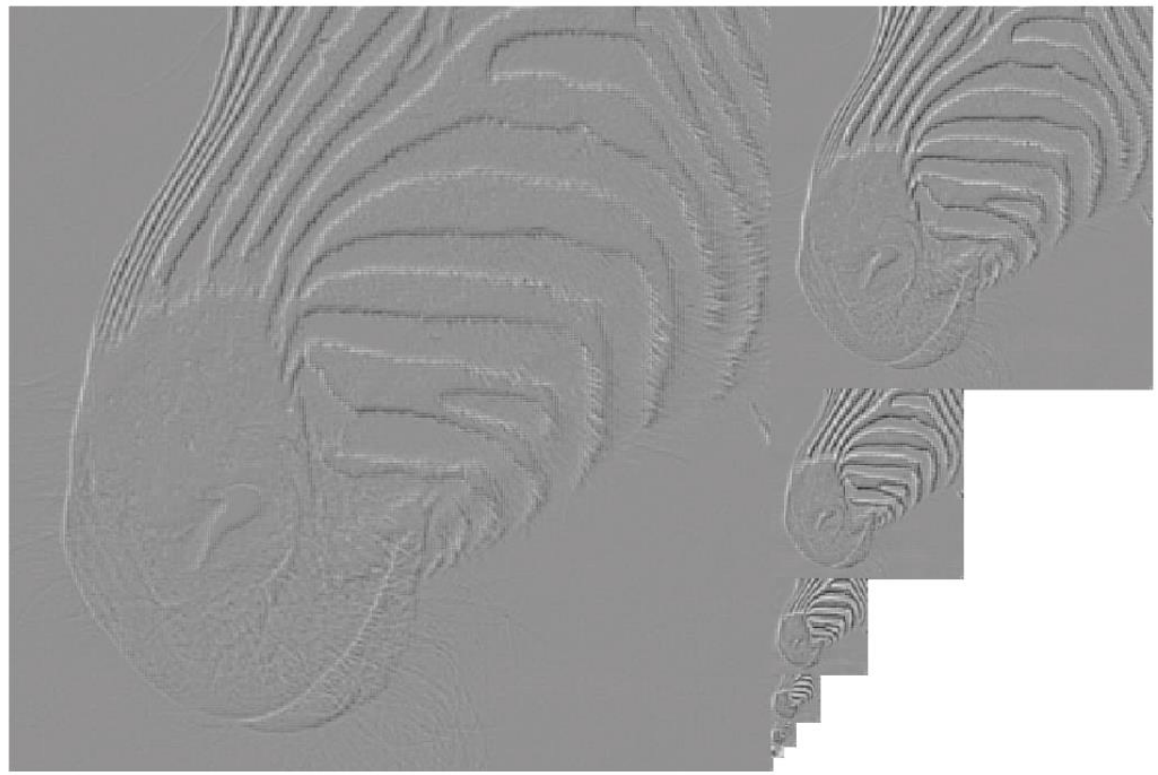


http://sepwww.stanford.edu/data/media/public/sep/morgan/texturematch/paper_html/node3.html#SECTION00012000000000000000

Laplacian Pyramid



512 256 128 64 32 16 8



From Forsyth

Two Popular Scale Invariant Detectors

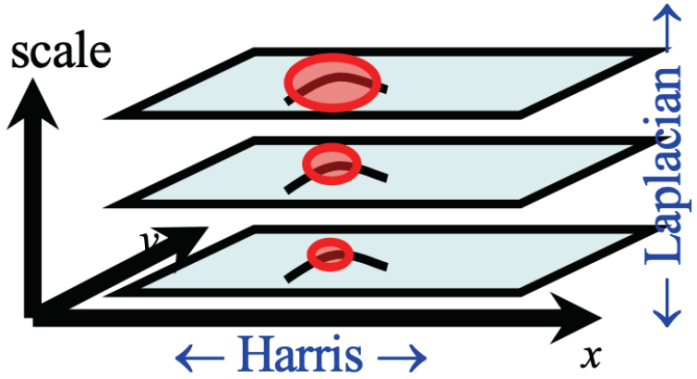
- Harris-Laplacian[1]

Find local maximum of:

  - Harris corner detector

in space (image coordinates)

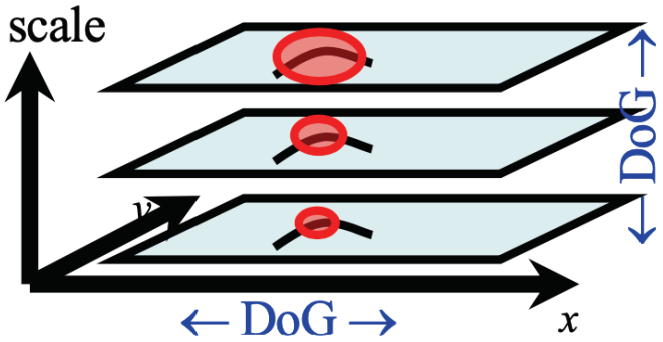
  - Laplacian in scale



- SIFT[2]

Find local maximum of:

  - Difference of Gaussians in space and scale



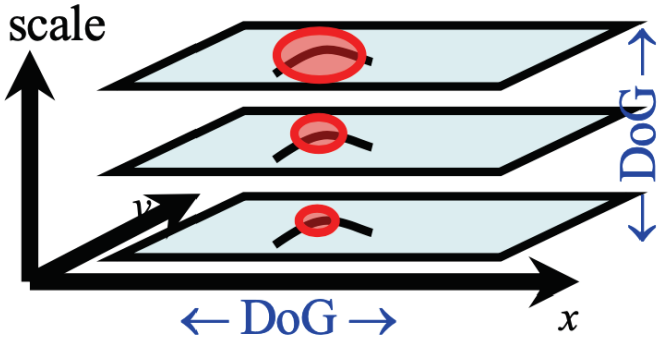
[1]K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2]D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

SIFT

- SIFT

Find local maximum of:

  - Difference of Gaussians in space and scale

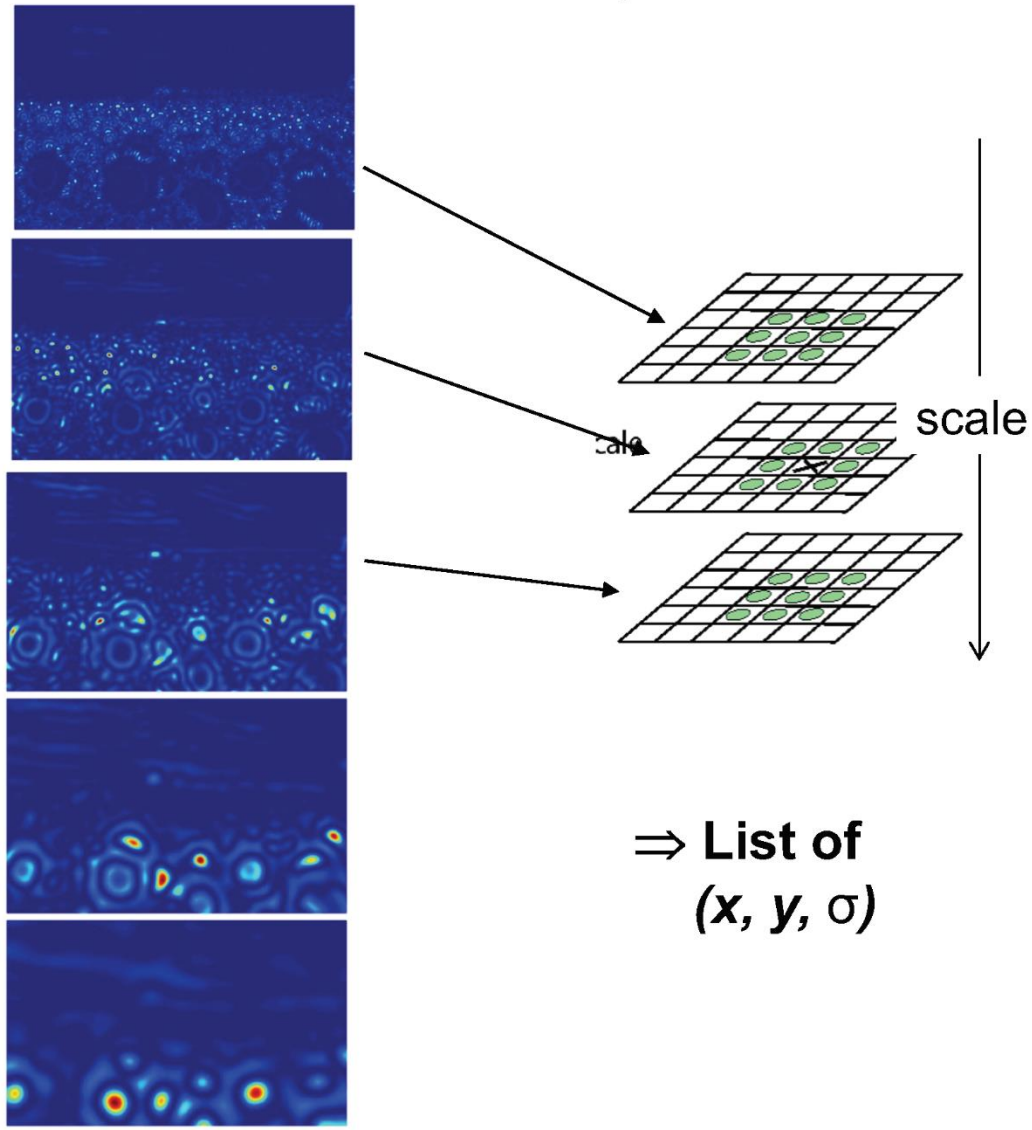


- Interest points are local maxima in both position and scale.

Scale invariant interest points

Interest points are local maxima in both position and scale.

$L_{xx}(\sigma) + L_{yy}(\sigma)$



⇒ **List of** ***(x, y,*** σ***)***

Squared filter response maps

Slide credit: Kristen Grauman

Keypoint Spatial Localization

- There are still a lot of points, some of them are not good enough.
- The locations of keypoints may not be accurate.
- Eliminating edge points.

Accurate Localization of Key Points

- Discard points with low contrast
  - Step one: Second order Taylor expansion of the LOG:

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^\mathbf{T}\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

  - Step two: Taking the derivative w.r.t x and set it to zero to refine the optimum.

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}.$$

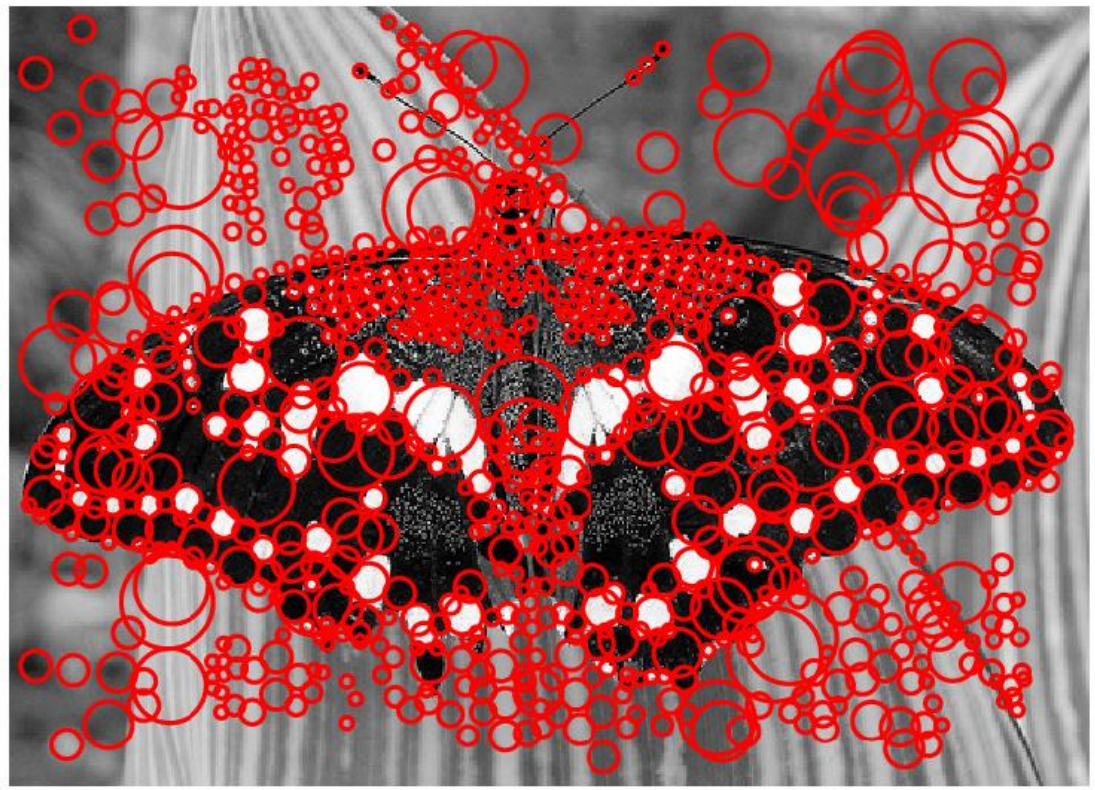
  - Step three: Evaluate D at $\hat{x}$

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}}.$$

  - Reject points with D less than a threshold (e.g 0.3 in the paper)

Eliminating edge responses

- Laplacian has strong response along edges



Removing Edge Points

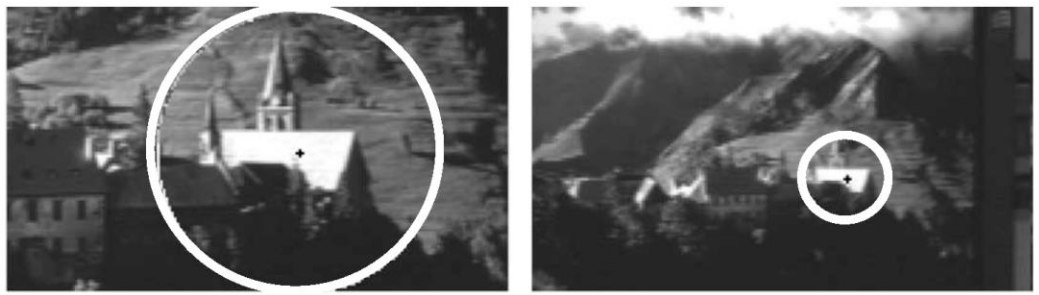
- Such a point has large principal curvature acrossthe edge but a small one in the perpendicular direction
- The principal curvatures can be calculated from a Hessian function $H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$
- The eigenvalues of H are proportional to the principal curvatures, so two eigenvalues shouldn't have much difference

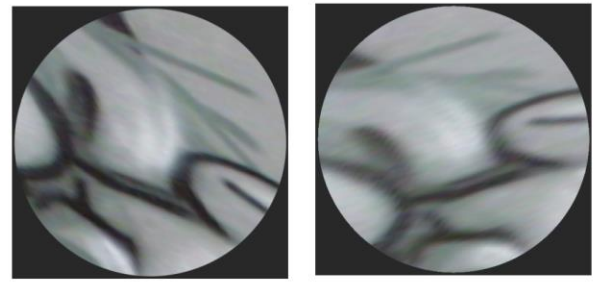
Overview

- Motivation for SIFT
- SIFT Feature Detector
- SIFT Descriptor
- Application

From feature detection to feature description

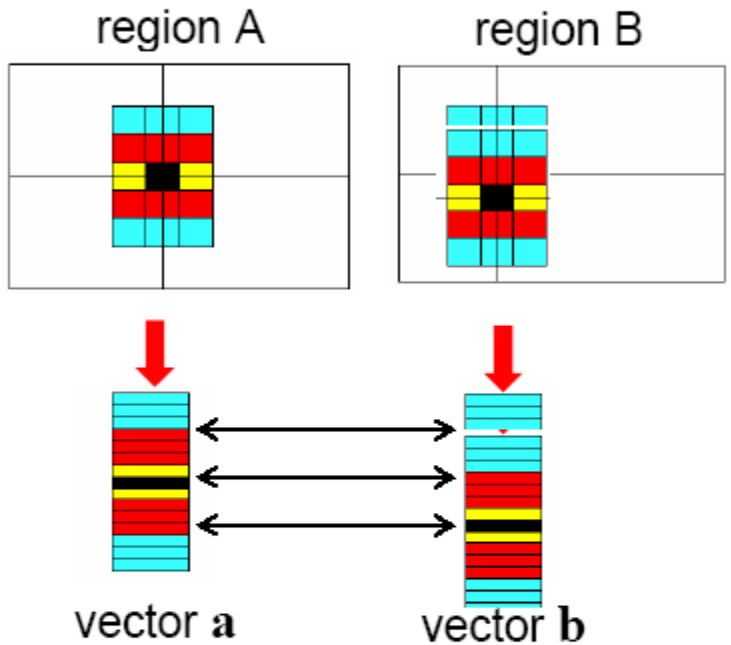
- To recognize the same pattern in multiple images, we need to match appearance "signatures" in the neighborhoods of extracted keypoints
  - But corresponding neighborhoods can be related by a scale change or rotation
  - We want to *normalize* neighborhoods to make signatures invariant to these transformations





Slide from Svetlana Lazebnik

Raw patches as local descriptors



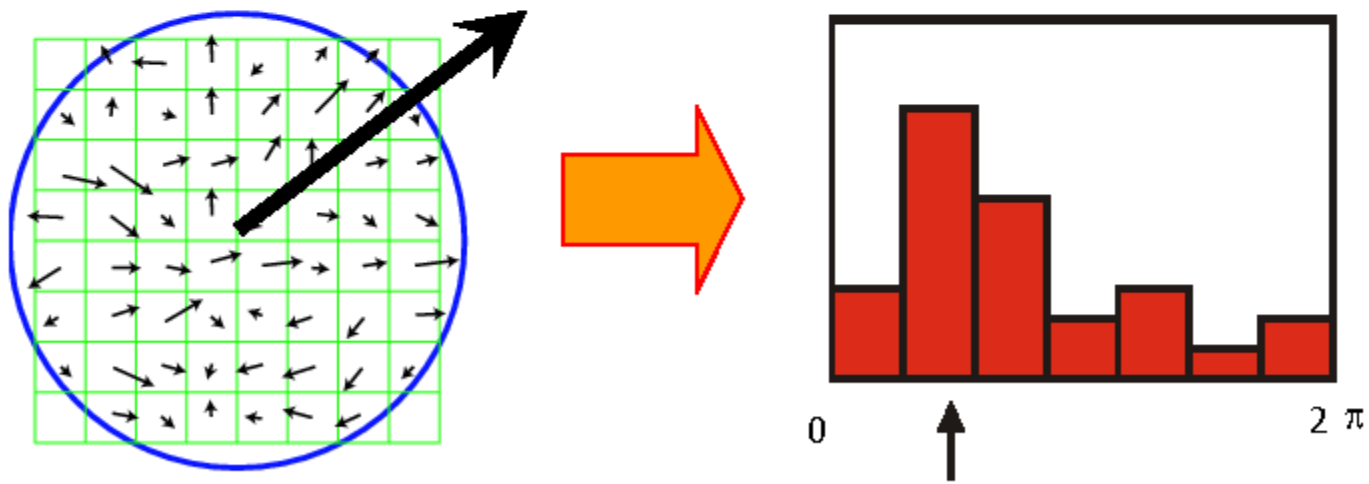
The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

But this is very sensitive to even small shifts, rotations.

Figure: Andrew Zisserman

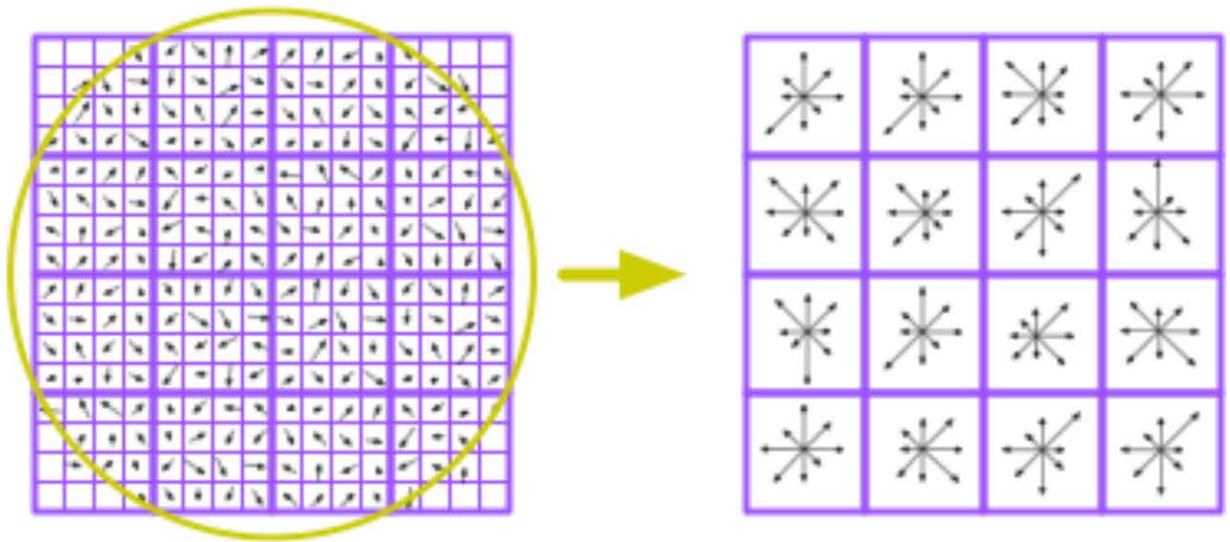
Finding a reference orientation

- Create histogram of local gradient directions in the patch
- Assign reference orientation at peak of smoothed histogram



SIFT descriptors

- Inspiration: complex neurons in the primary visual cortex



D. Lowe, Distinctive image features from scale-invariant keypoints, *IJCV* 60 (2), pp. 91-110, 2004

SIFT Descriptor

- Based on 16x16 image patch
- 4x4 subregions
- 8 bins in each subregion
- 4x4x8=128 dimensions in total

128-D Descriptor

- 16x16 Gradient window is taken. Partitioned into 4x4 subwindows.
- Histogram of 4x4 samples in 8 directions
- Gaussian weighting around center(σ is 0.5 times that of the scale of a keypoint)
- 4x4x8 = 128 dimensional feature vector

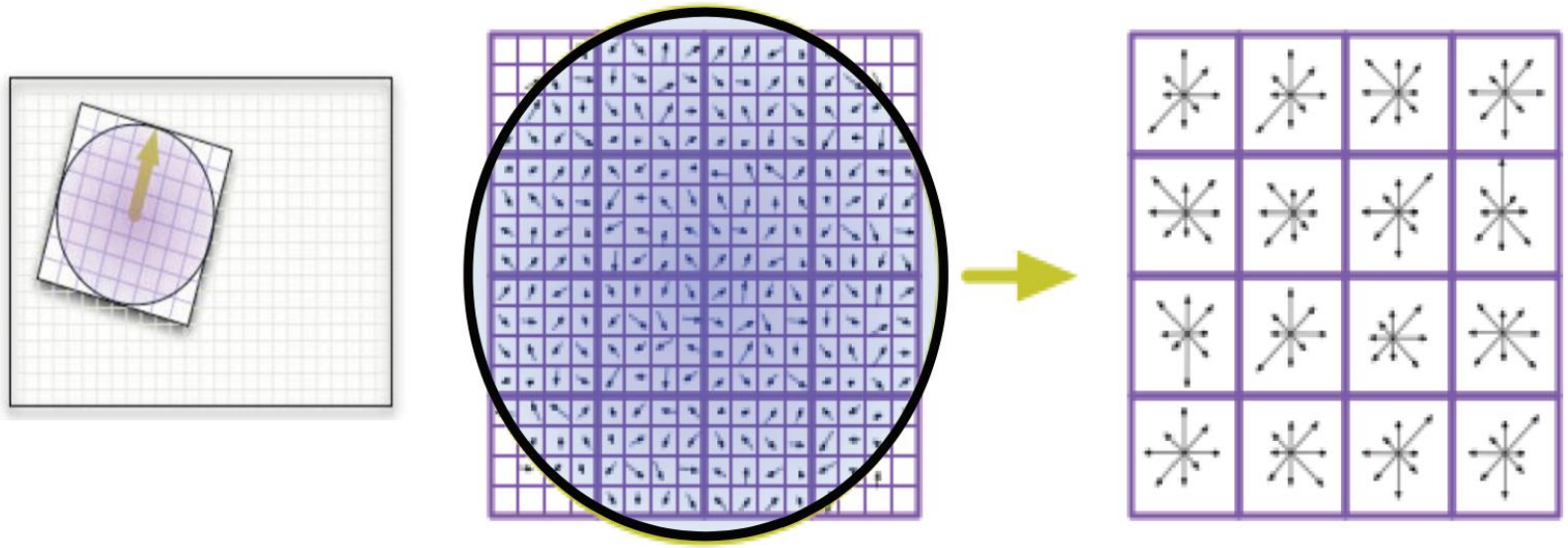
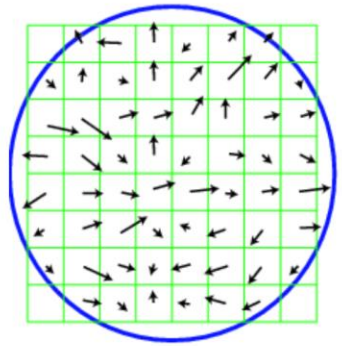


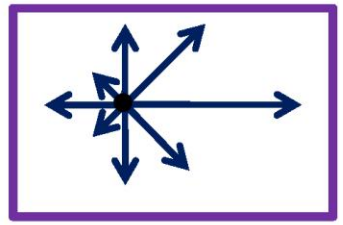
Image from: Jonas Hurrelmann

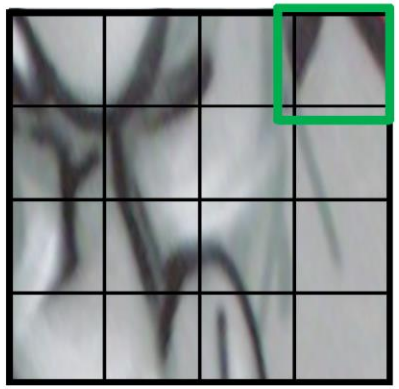
Scale Invariant Feature Transform (SIFT) descriptor [Lowe 2004]

- Use histograms to bin pixels within sub-patches according to their orientation.

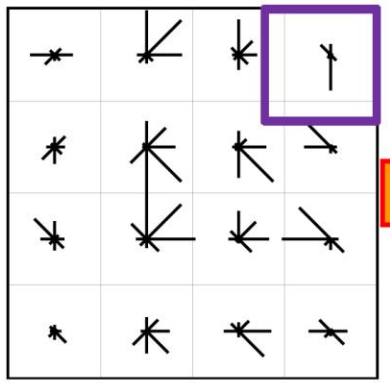


gradients





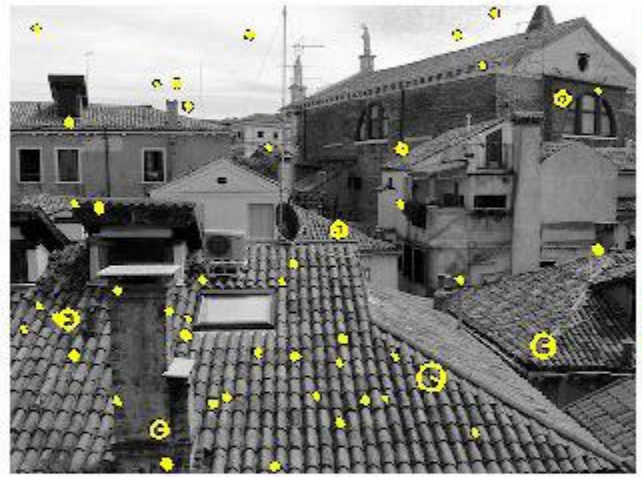
subdivided local patch



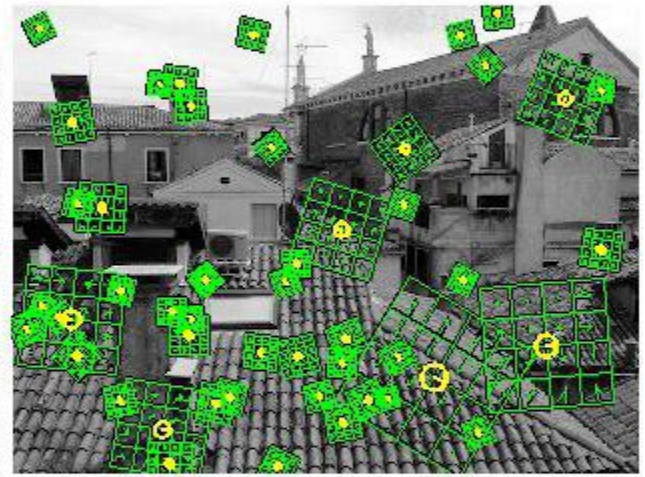
histogram per grid cell

Final descriptor = concatenation of all histograms, normalize

Scale Invariant Feature Transform (SIFT) descriptor [Lowe 2004]



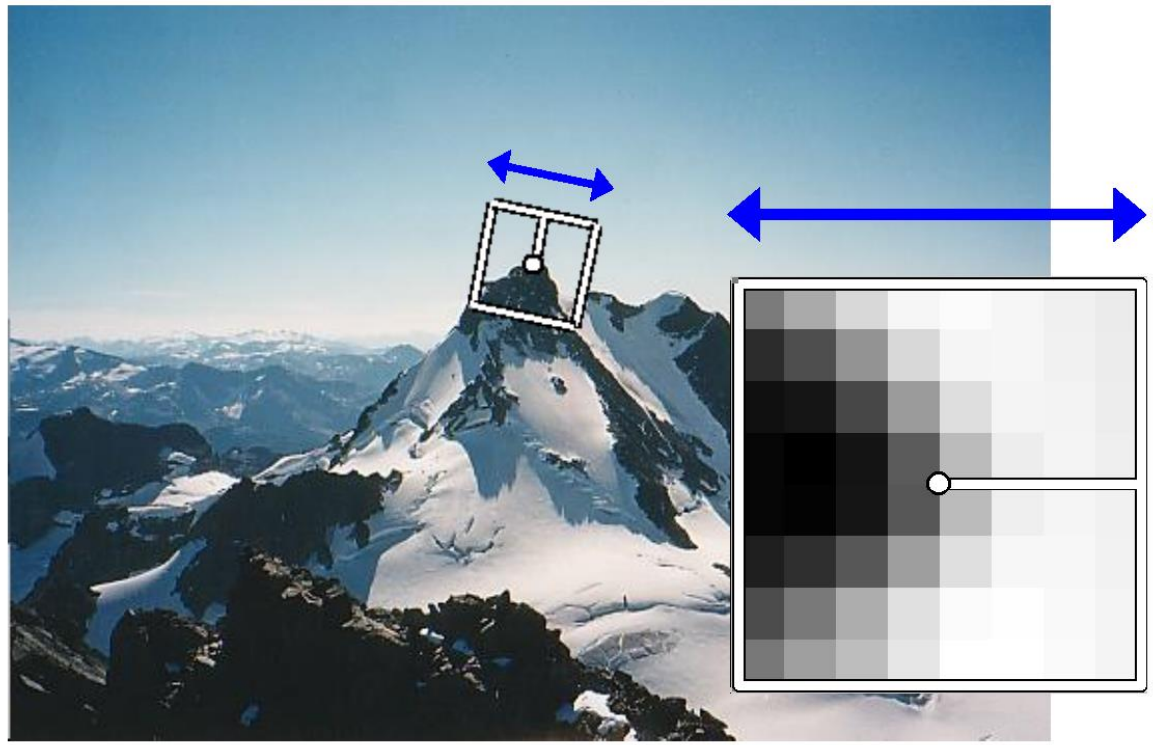
Interest points and their scales and orientations (random subset of 50)



SIFT descriptors

http://www.vlfeat.org/overview/sift.html

Making descriptor rotation invariant

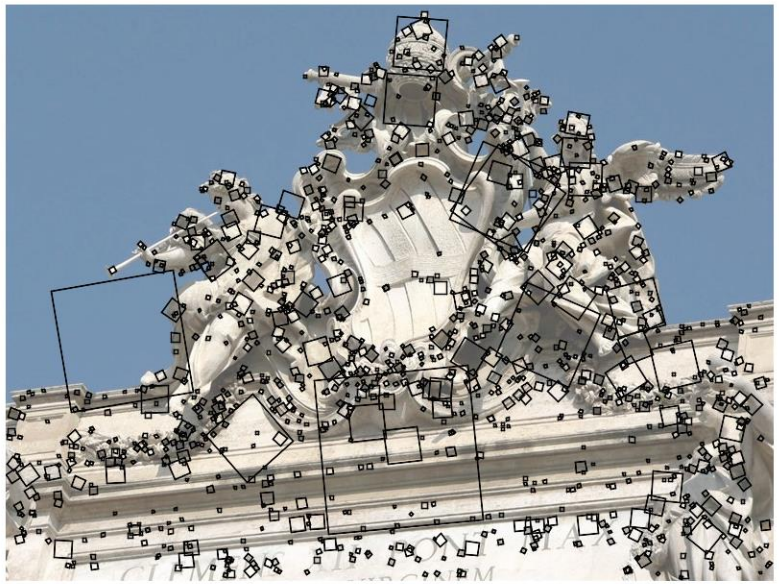


- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.

Image from Matthew Brown

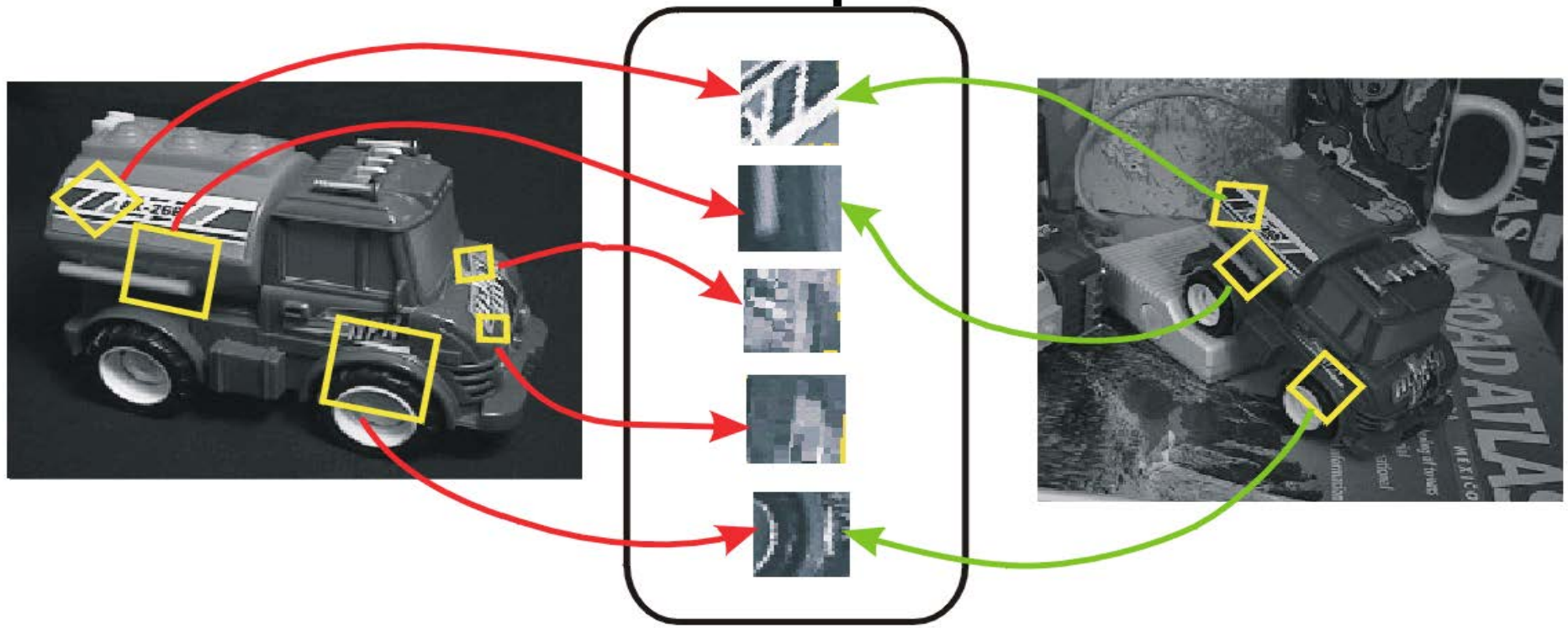
SIFT features

- Detected features with characteristic scales and orientations:



David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

From keypoint detection to feature description



- Detection is *covariant*:

  features(transform(image)) = transform(features(image))

- Description is *invariant*:

  features(transform(image)) = features(image)

SIFT descriptor performance

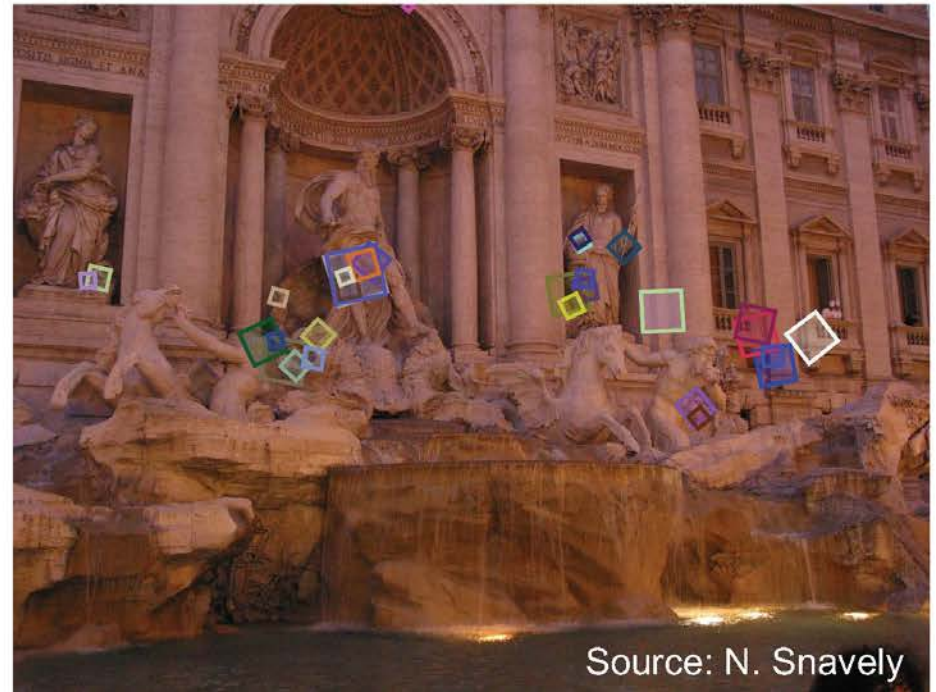
- Very robust to view angle chages.
  - 80% repeatability at:
    - 10% image noise
    - 45° viewing angle
    - 1k-100k keypoints in database

- Best *descriptor* according to [Mikolajczyk & Schmid 2005]'s extensive survey

Properties of SIFT

Extraordinarily robust detection and description technique

- Can handle changes in viewpoint
  - Up to about 60 degree out-of-plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night
- Fast and efficient—can run in real time
- Lots of code available



Source: N. Snavely

Other Scale/View-angle Invariant Features

- SURF
- Harris-Affine
- Ahrris-Laplacian
- MSER
- Daisy
- BRIEF
- ORB
- BRISK
- ……

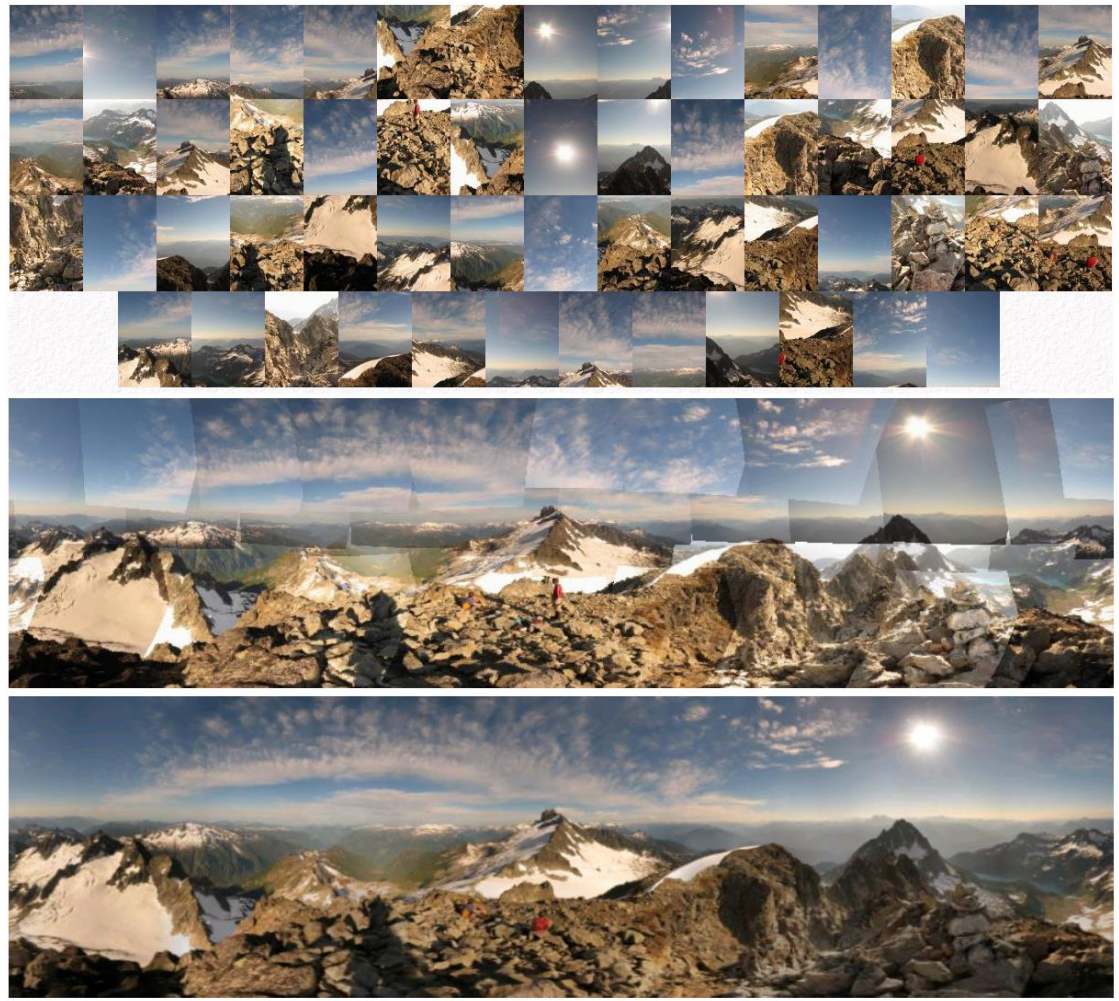
- For most local invariant feature detectors, executables are available online:

  - http://robots.ox.ac.uk/~vgg/research/affine

Applications of local invariant features

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition
- …

Automatic mosaicing



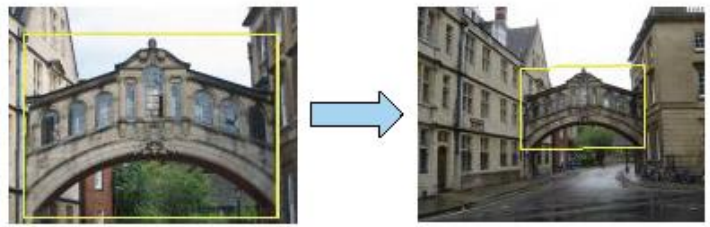
Matthew Brown

http://matthewalunbrown.com/autostitch/autostitch.html

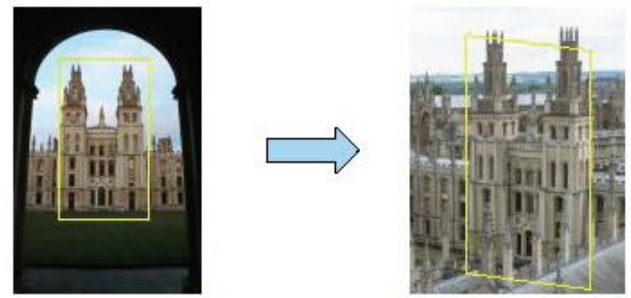
Wide baseline stereo

[Image from T. Tuytelaars ECCV 2006 tutorial]

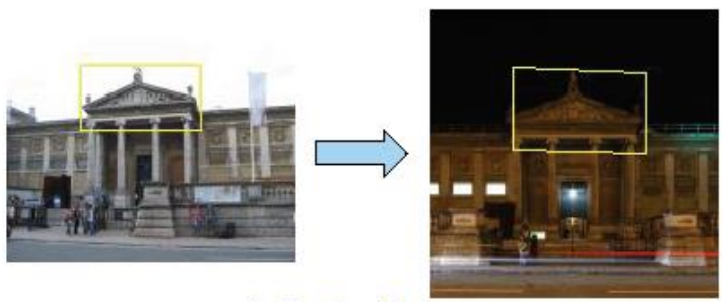
Recognition of specific objects, scenes



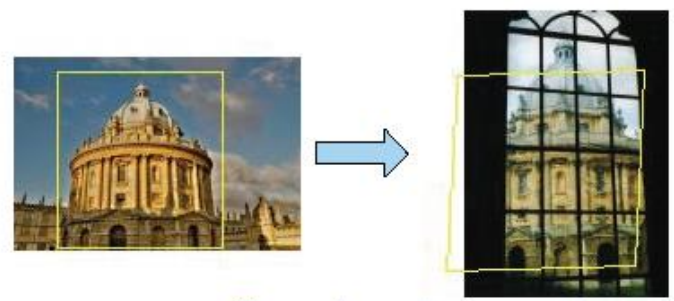
Scale



Viewpoint



Lighting



Occlusion

Slide credit: J. Sivic

Reading

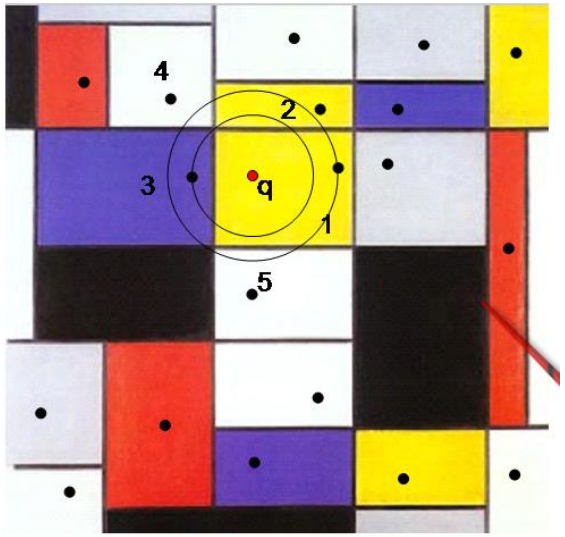
- SIFT on Wikipedia.
- D. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60 (2), pp. 91-110, 2004. This paper contains details about efficient implementation of a Difference-of-Gaussians scale space.
- T. Lindeberg, "Feature detection with automatic scale selection," International Journal of Computer Vision 30 (2), pp. 77-116, 1998. This is advanced reading for those of you who are *really* interested in the mathematical details.

More useful slides on SIFT

The SIFT (Scale Invariant Feature Transform) Detector and ...
courses.cs.washington.edu › notes › SIFT_white2011
PDF
Apr 15, 2011 - The **SIFT** (Scale Invariant **Feature**. Transform) **Detector** and Descriptor developed by David Lowe. University of British Columbia. Initial paper ...

Matching

Closest-point indexing – KD Trees



Piet Mondrian, early use of KD trees

2 Answers

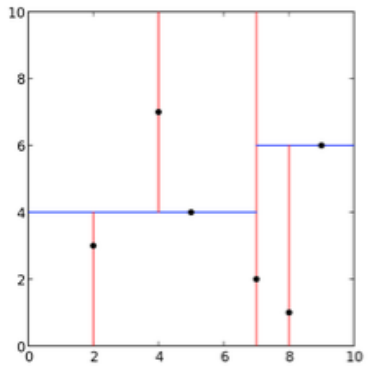
Steven Schmatz, BSE/MSE Computer Science, University of Michigan College of Engineering (2019)

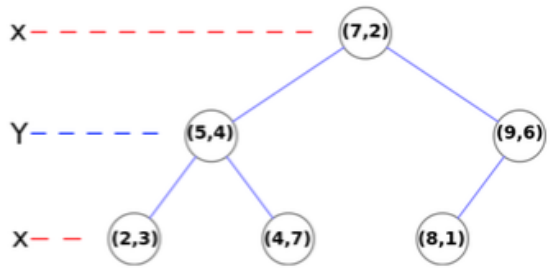
Answered May 29, 2018 · Upvoted by Kaan Yılmaz, M.S. Machine Learning & Complex Adaptive Systems, Chalmers University of Technology (2020) and Alon Amit, CS degree and many years of coding.

Originally Answered: How does a k-d tree work to find the K nearest neighbors?

K-d trees are a wonderful invention that enable $O(k \log n)$ **(expected) lookup times** for the k nearest points to some point x. This is extremely useful, especially in cases where an $O(n)$ lookup time is intractable already.

First, you need to understand what a k-d tree is. A k-d tree is a data structure that partitions space by repeatedly choosing a point in the data set to split the current partition in half. It accomplishes this by alternating the dimension which performs the split.





So here, we first start by splitting at the line $x = 7$, then we split by $y = 4$ on the left and $y = 6$ on the right. ... (more)

154 1 2

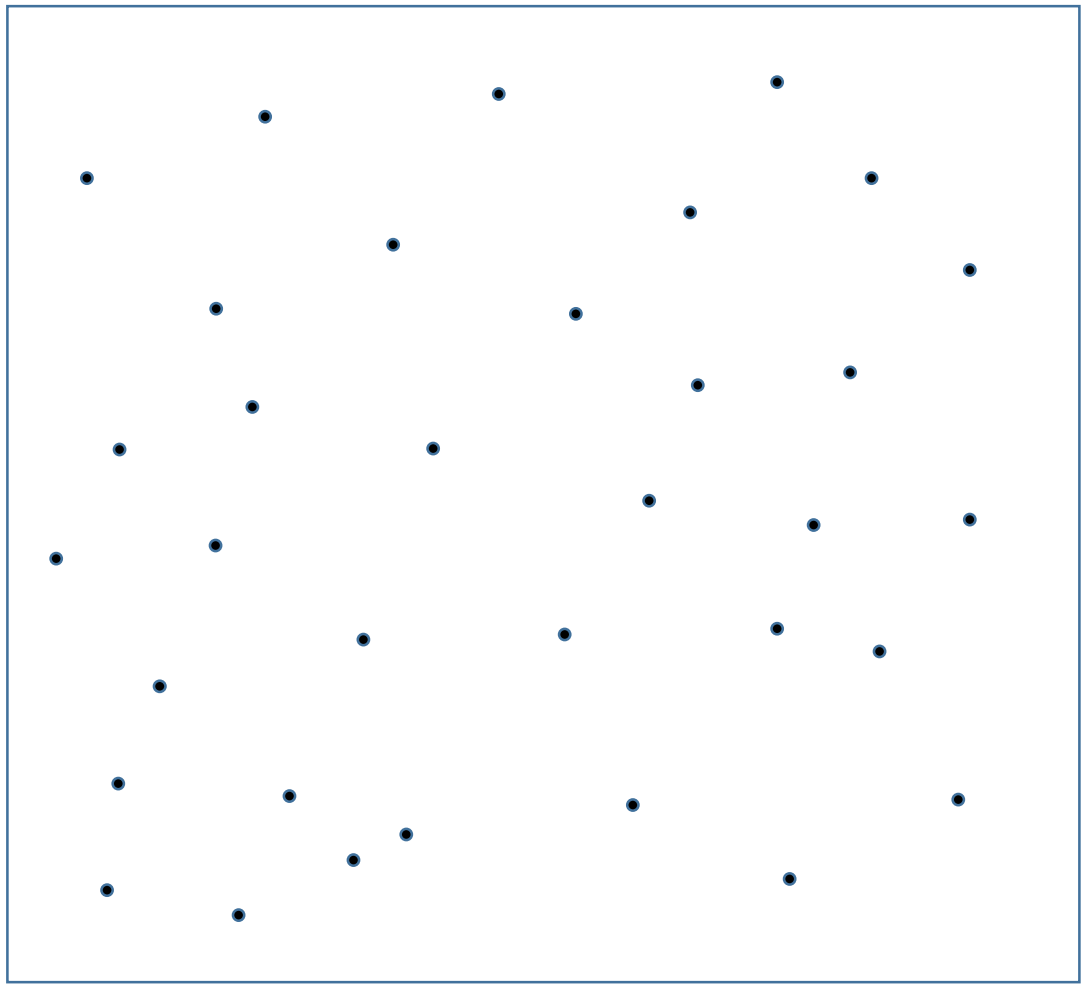
2 comments from Konstantin Burlachenko and more

KD-trees – two phases

- Construction of tree
- Query for closest point.

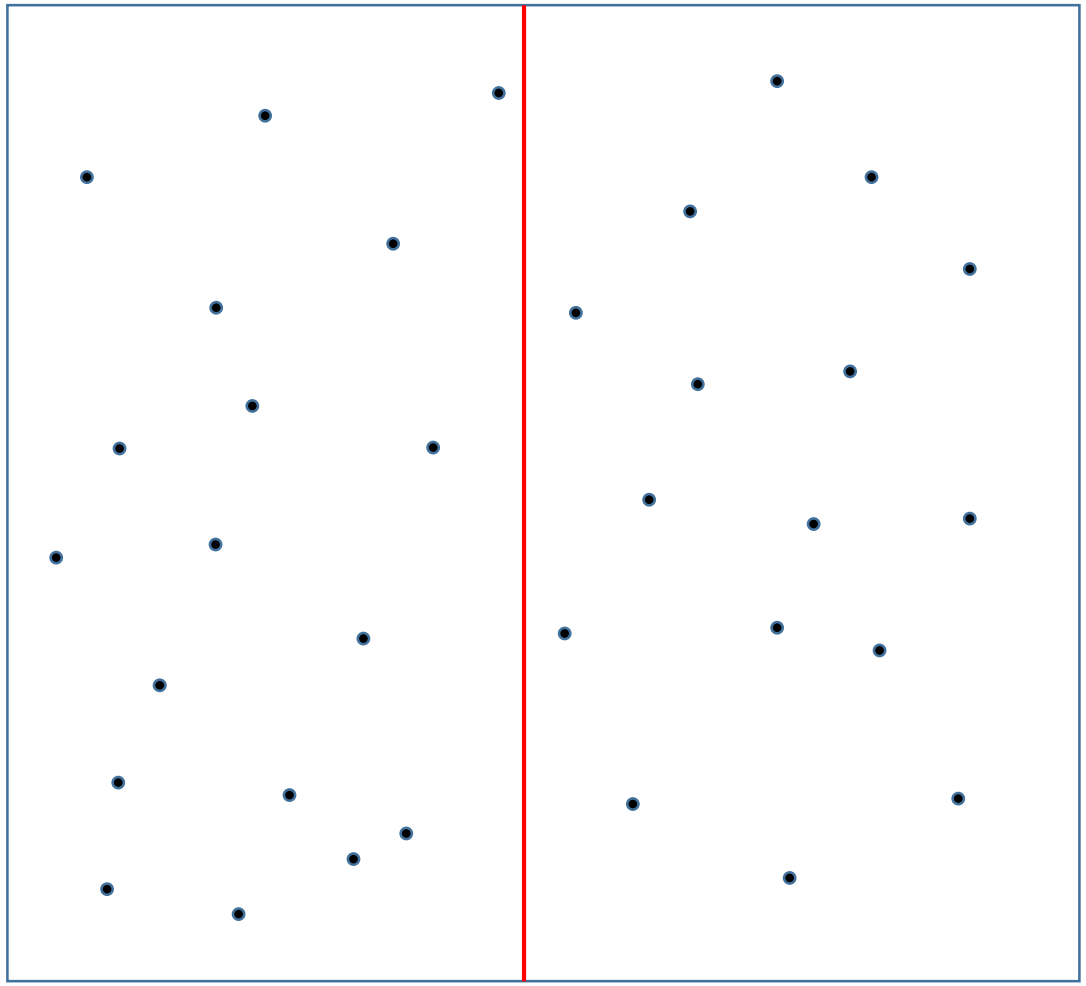
KD-trees – two phases

- Construction of tree



KD-trees – two phases

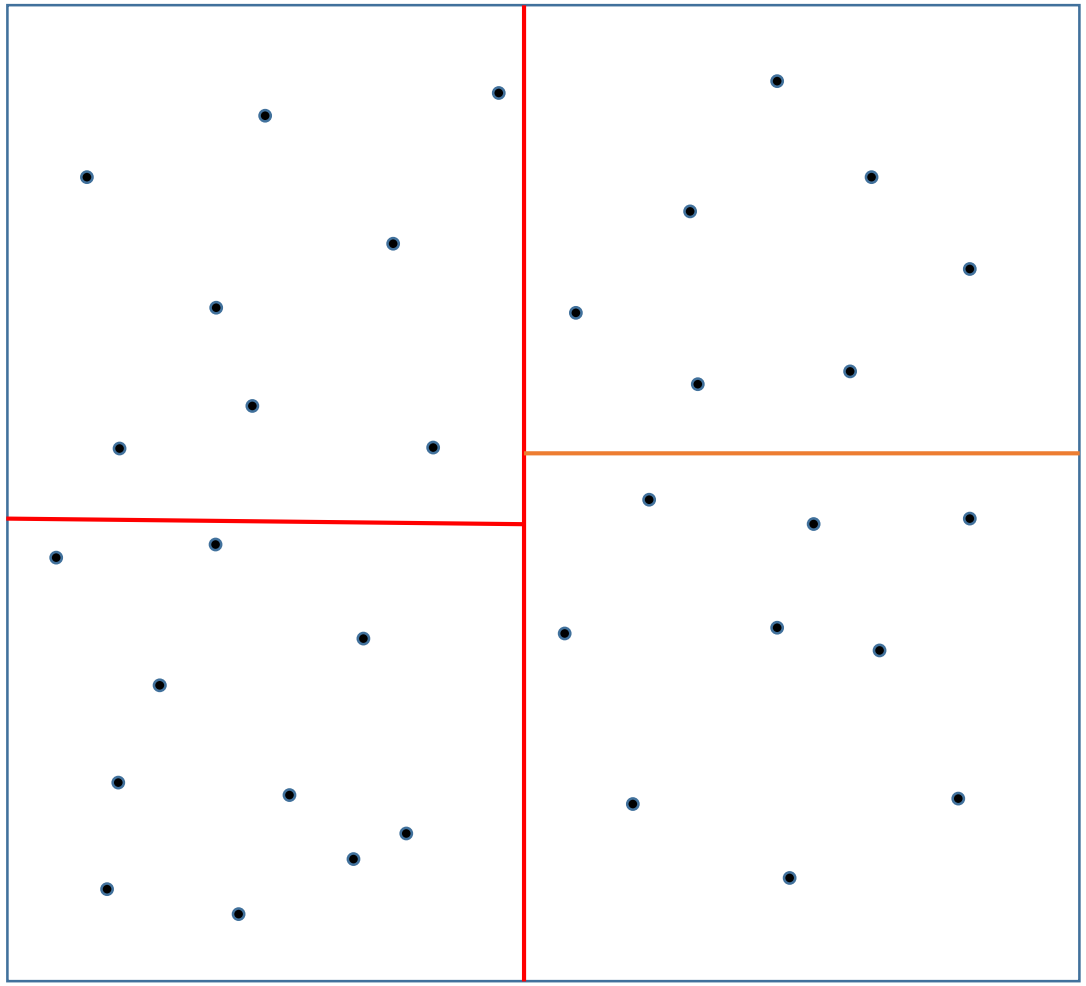
- Construction of tree



Split data in half along one dimension

KD-trees – two phases

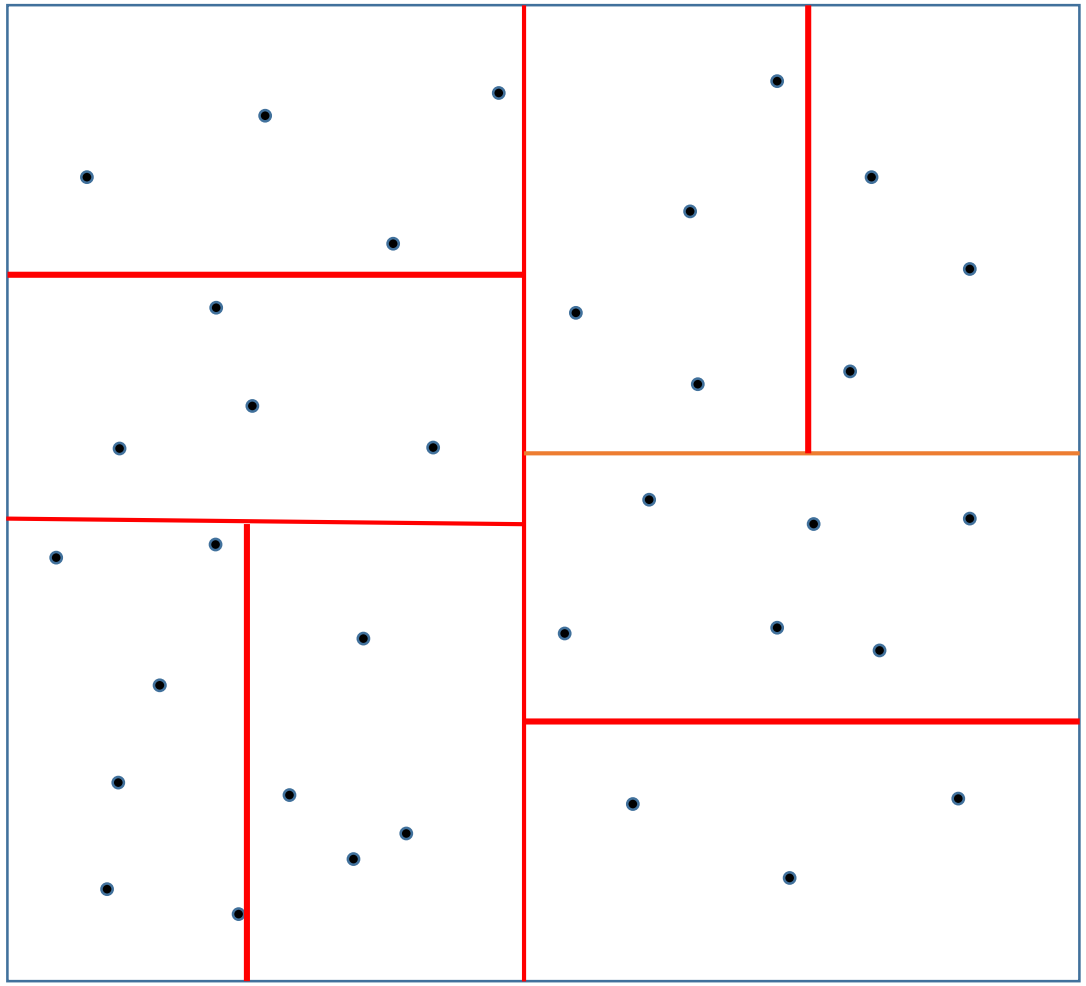
- Construction of tree



Split each of the resulting cells along some dimension

KD-trees – two phases

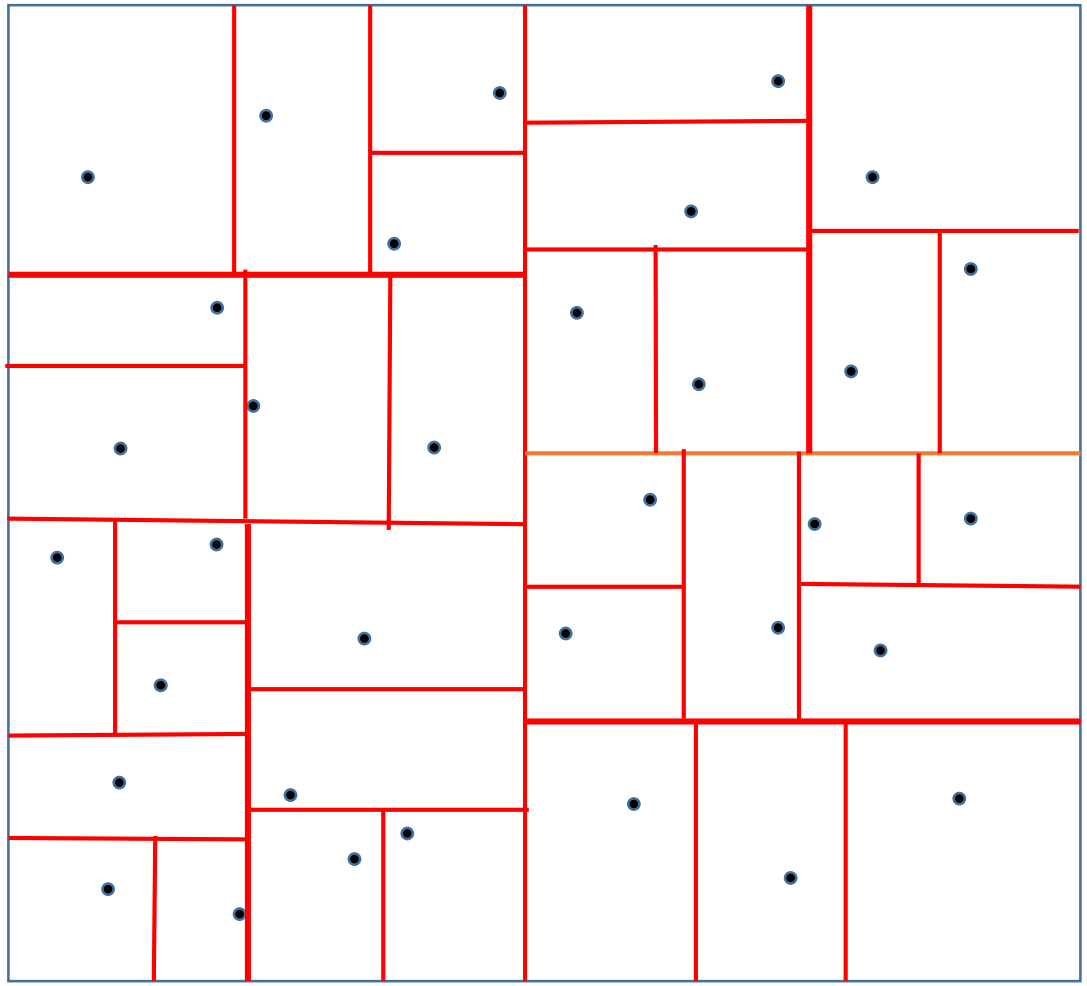
- Construction of tree



Keep going

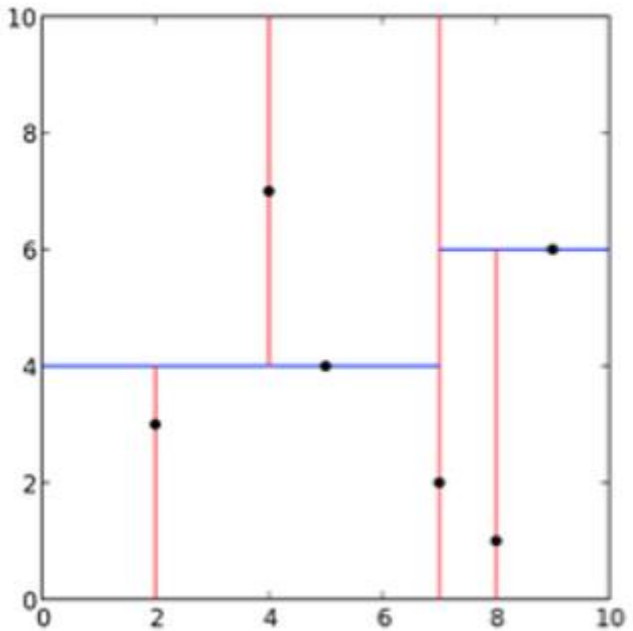
KD-trees – two phases

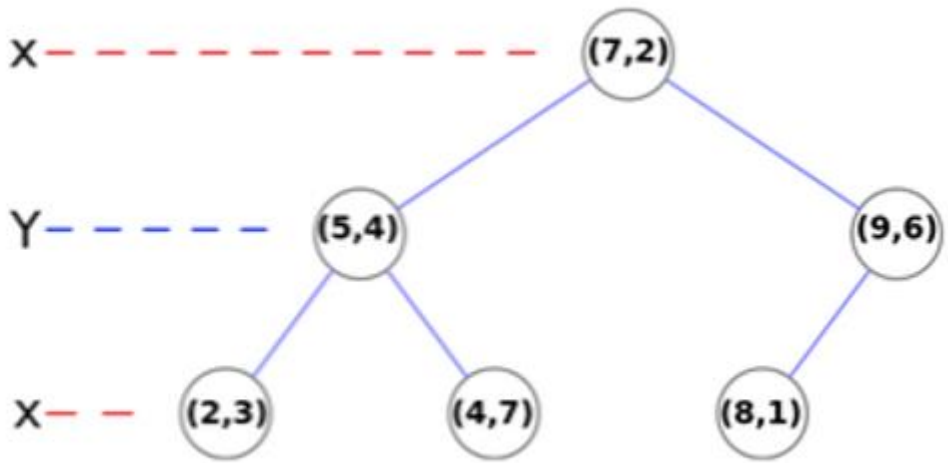
- Construction of tree



Until cells contain only one point

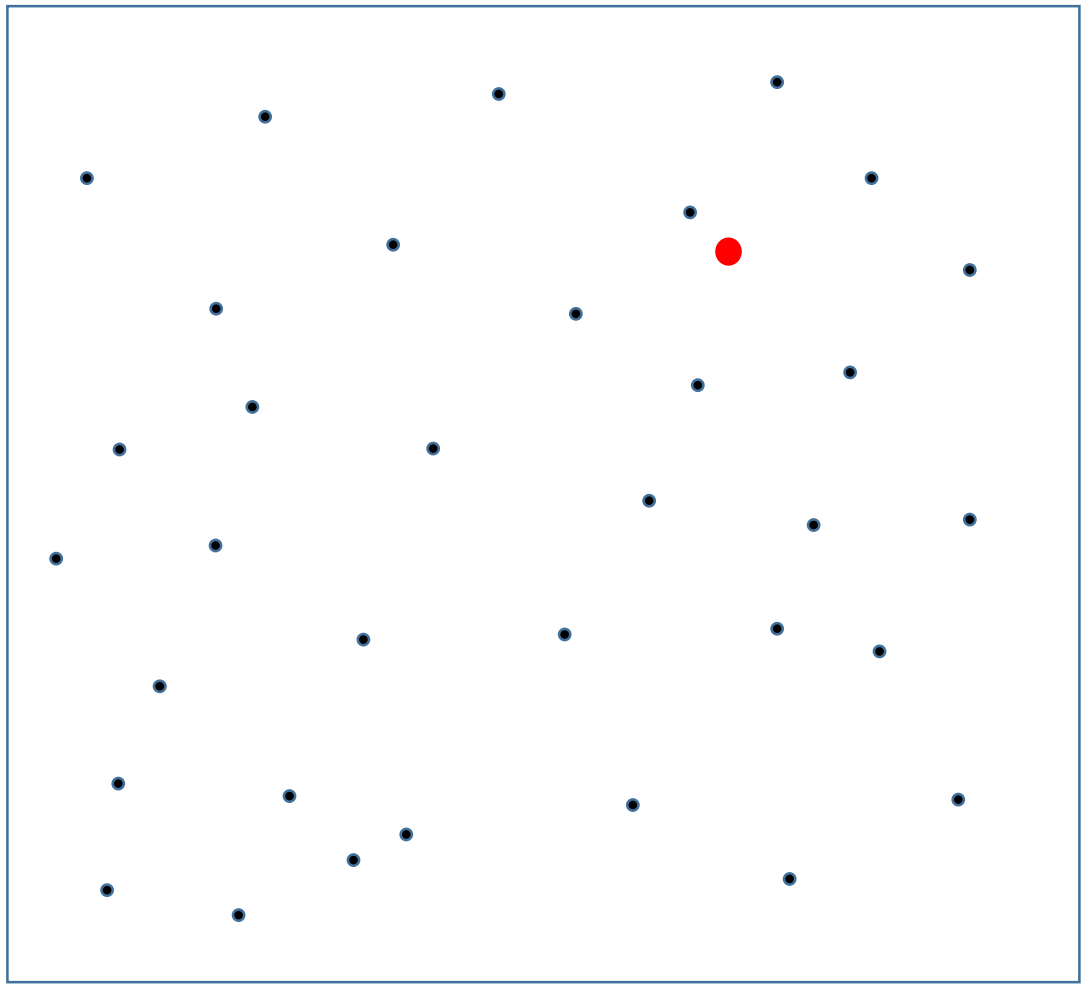
Build a search tree
- Each node stores the dimension number and partition value where split was made





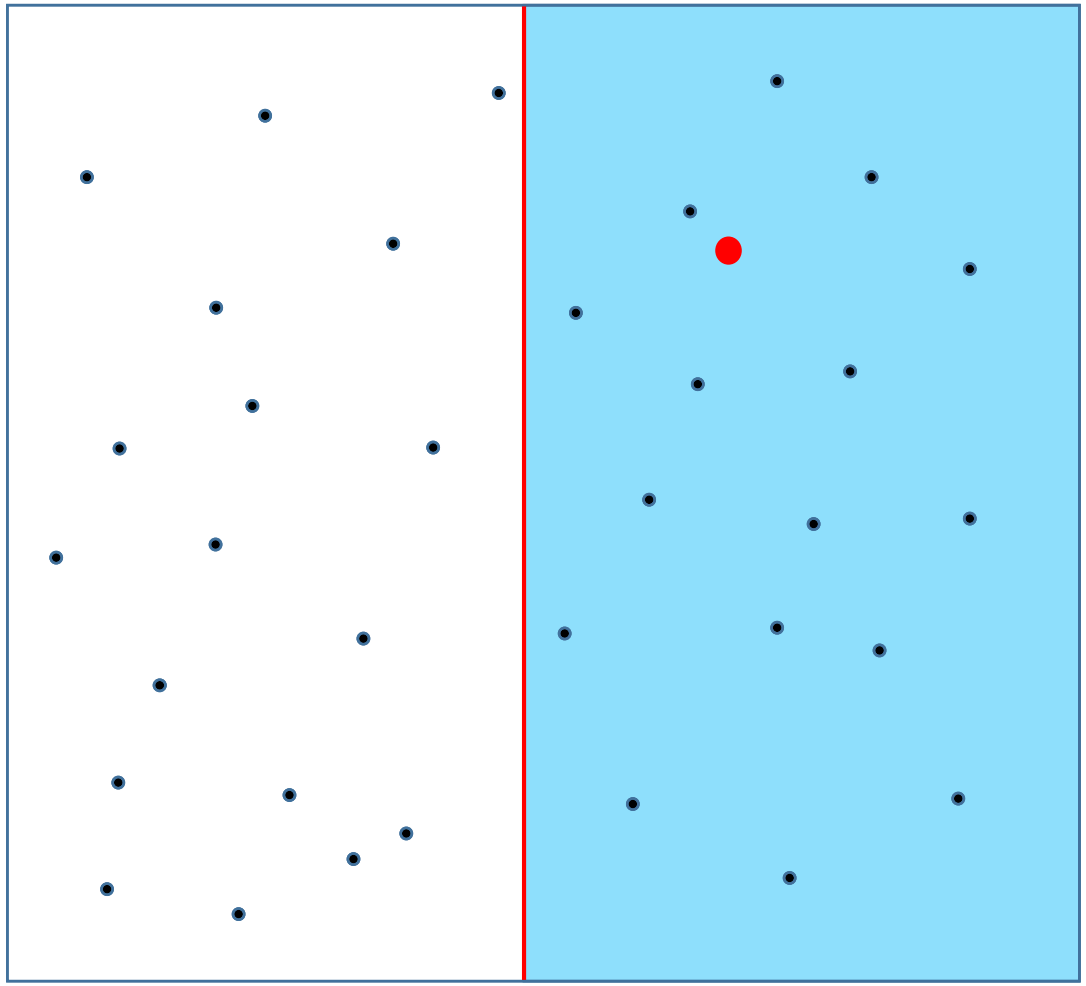
KD-trees – two phases

- Search



KD-trees – two phases

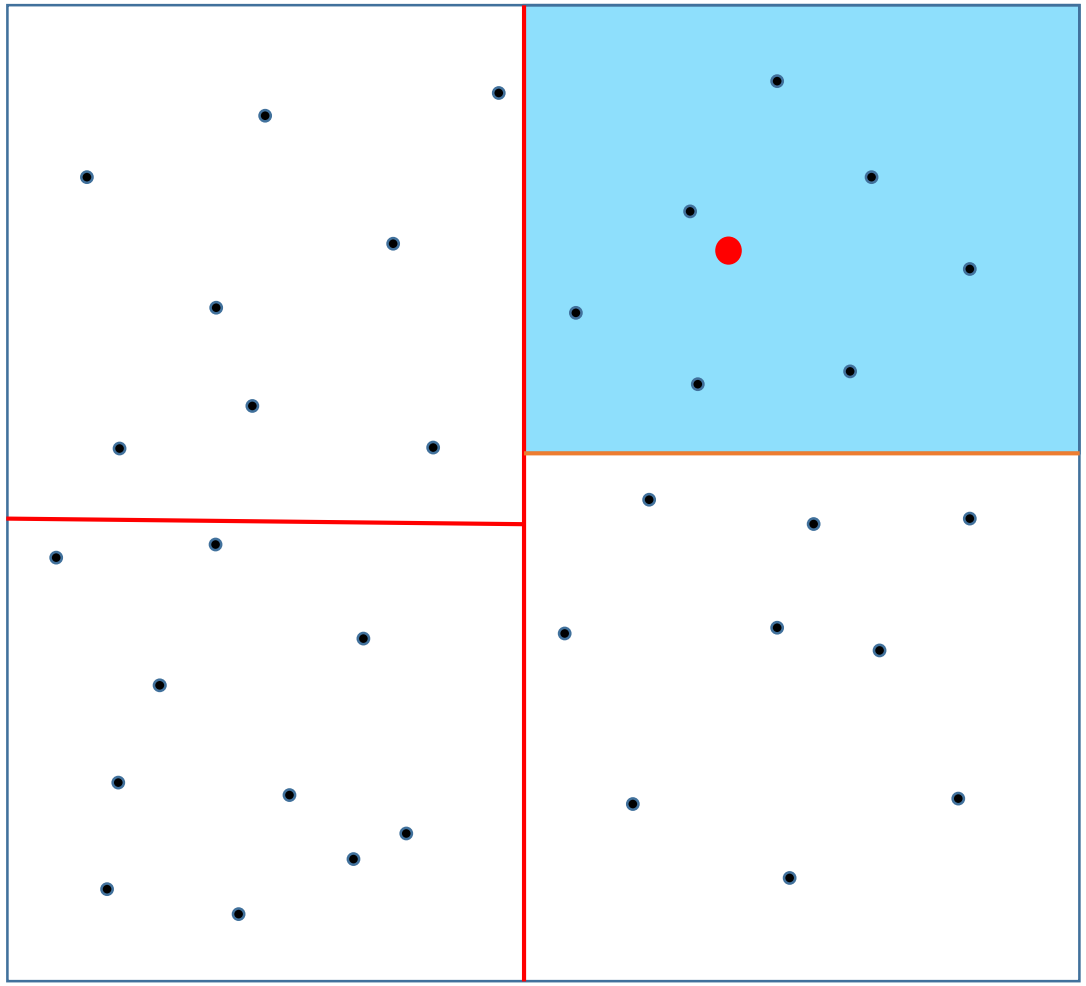
- Construction of tree



Find which partition query lies in

KD-trees – two phases

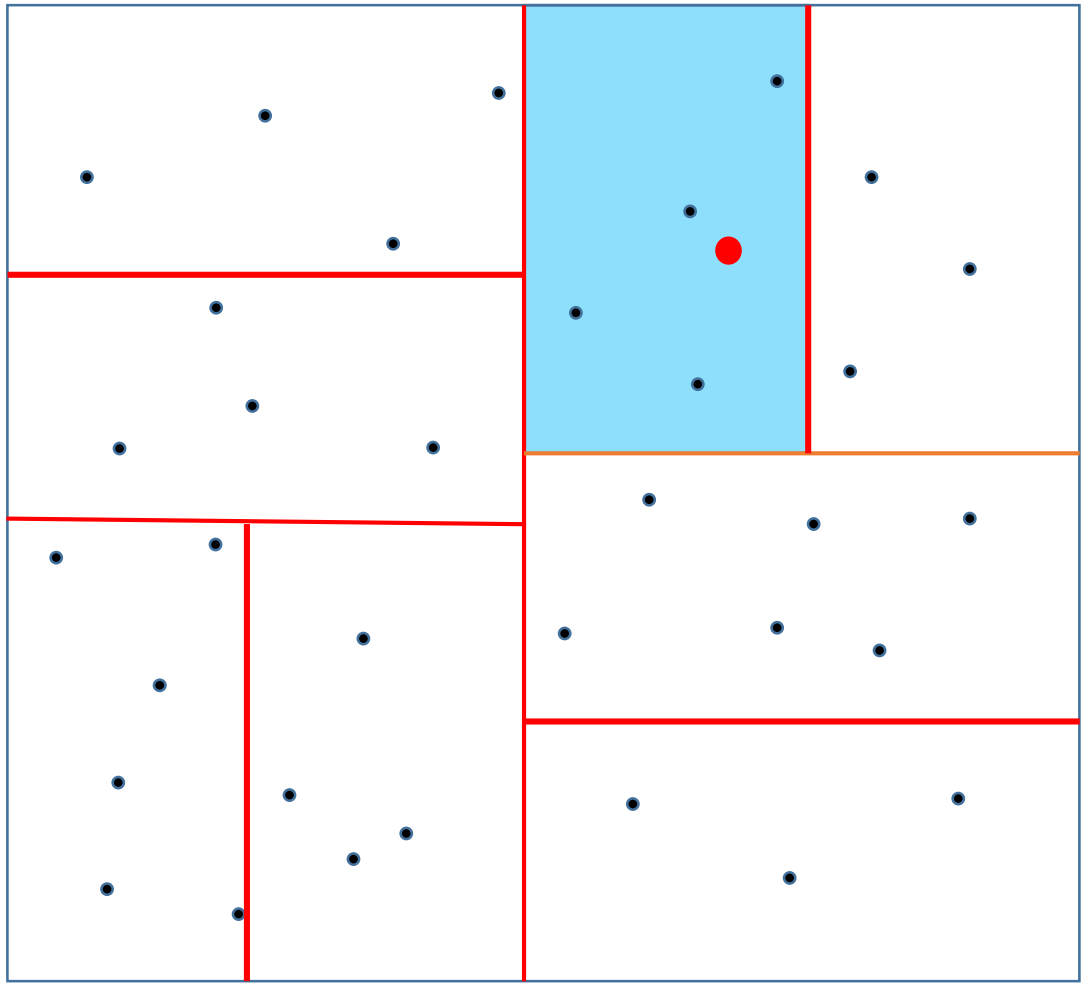
- Construction of tree



Keep searching down tree

KD-trees – two phases

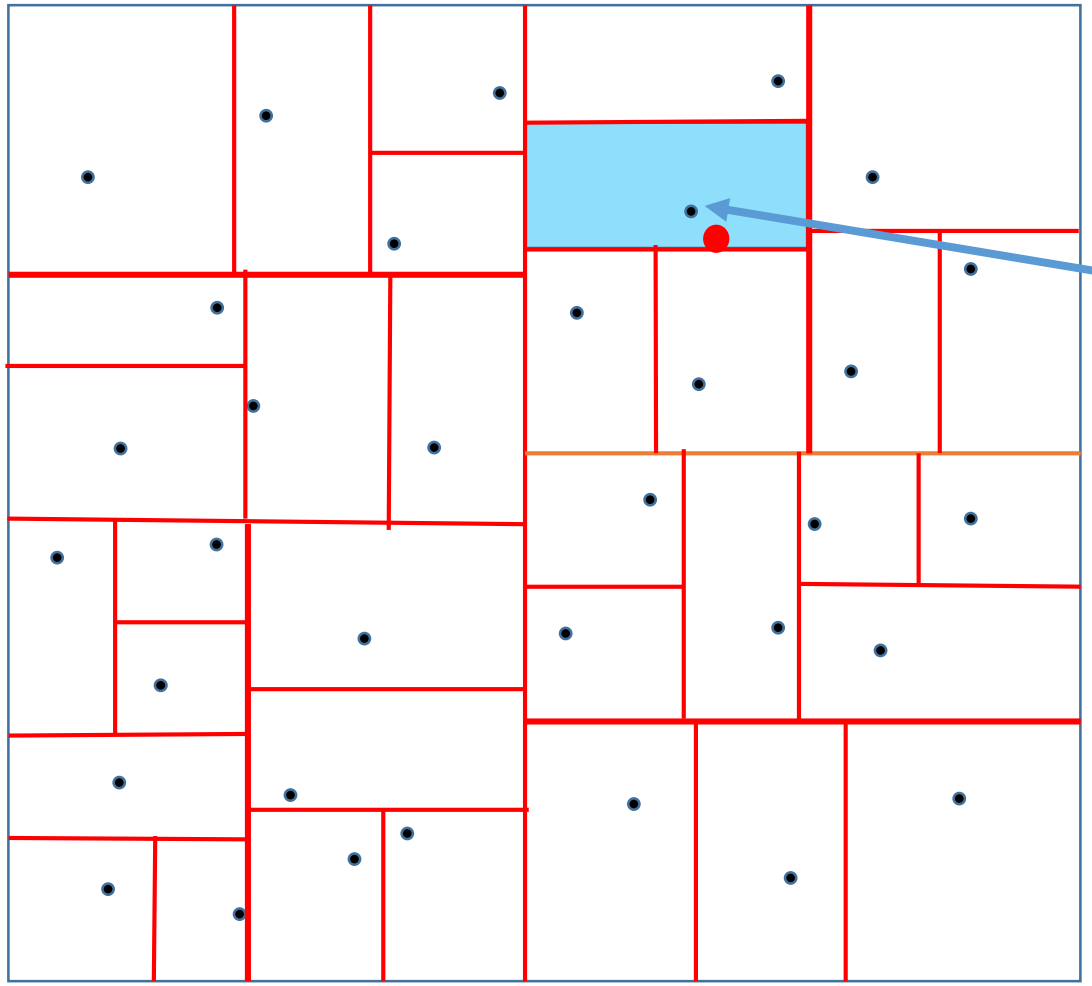
- Construction of tree



Keep going

KD-trees – two phases

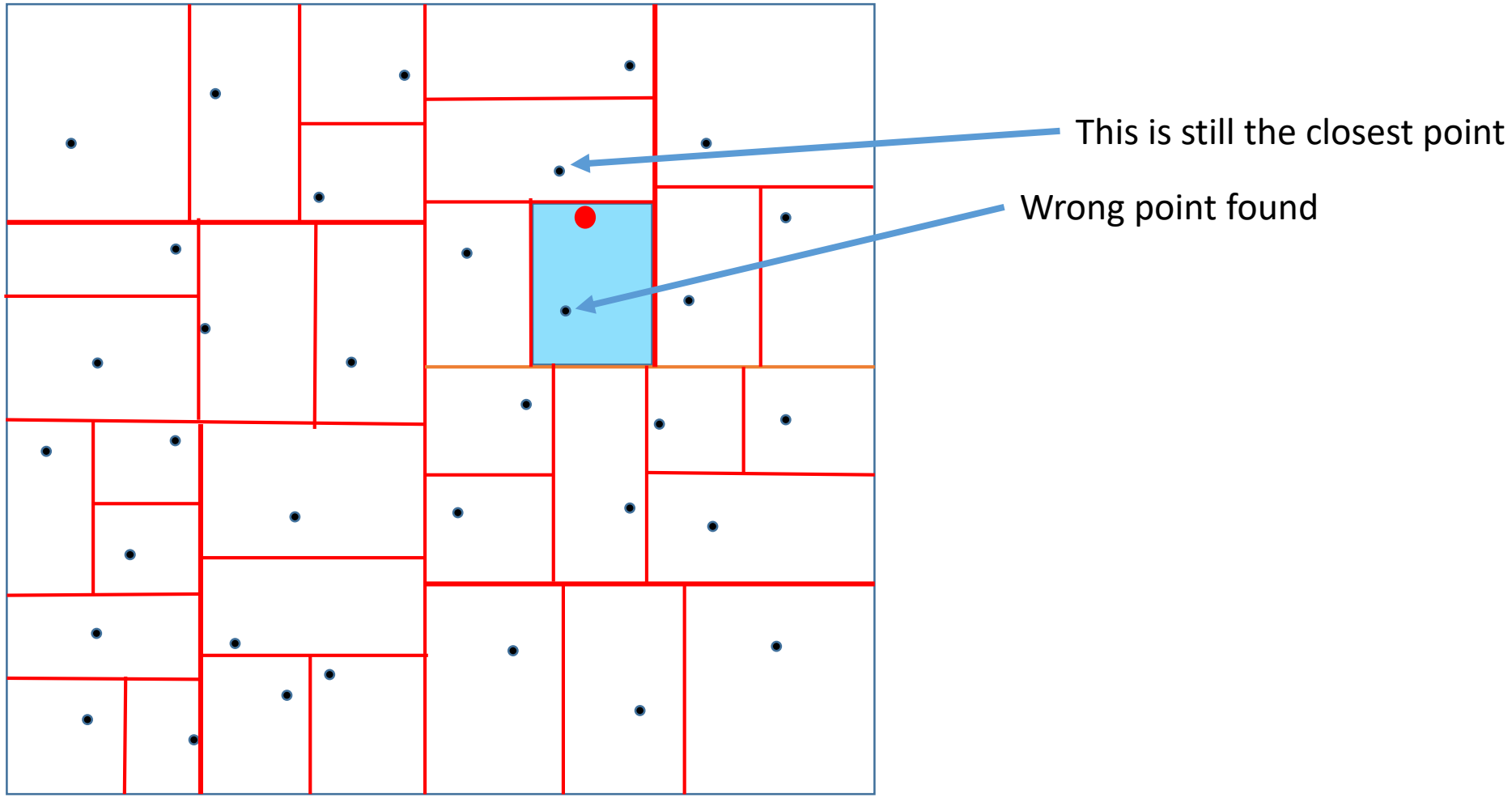
- Construction of tree



Closest point is point in cell

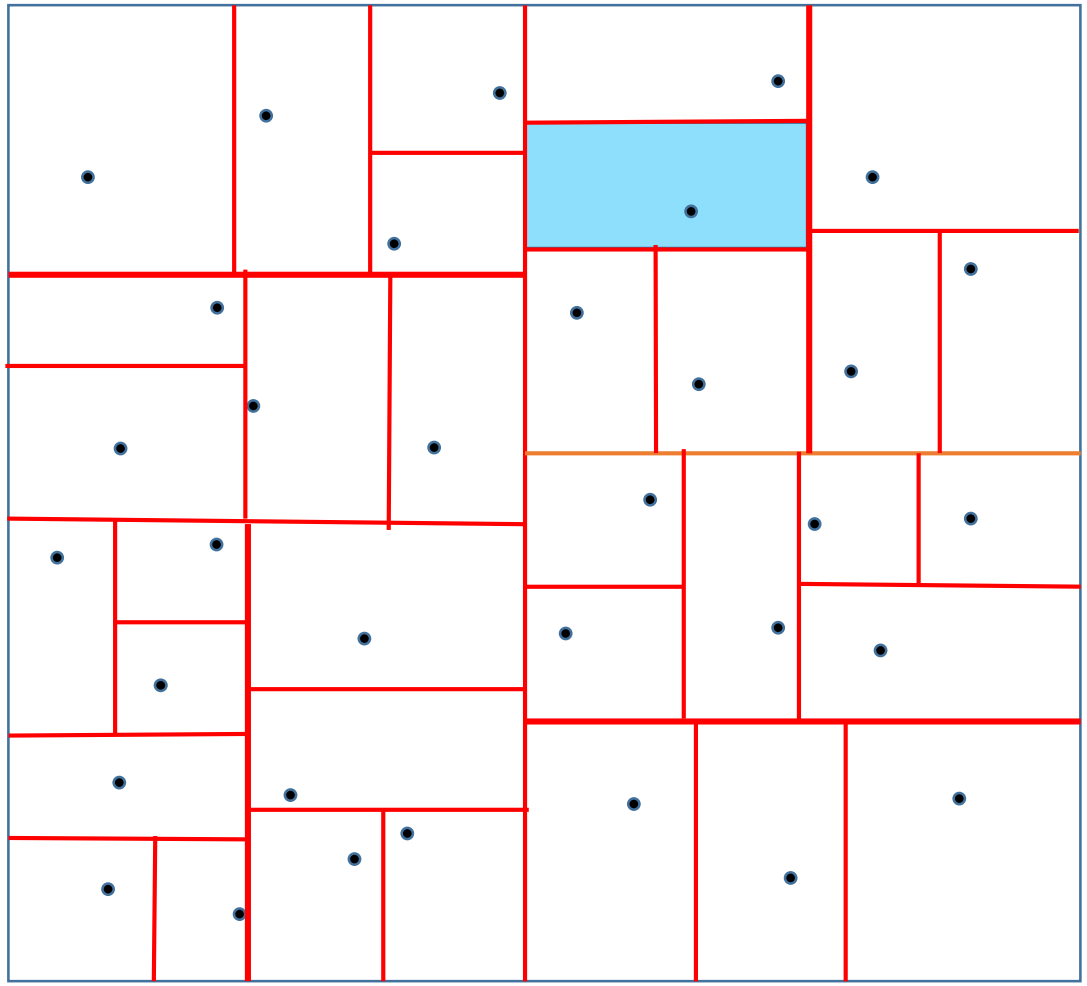
KD-trees – two phases

- It can be worse



KD-trees – two phases

- Trace back



Solution:
Trace back up the tree, seeing where you went wrong

KD tree

- Works very well in low dimensions
- Less and less reliable in higher dimensions
- The curse of dimensionality.

SIFT descriptor is length 128. Need closest search in 128-dimensions.

- Suppose there are 10^6 = 2^20 features
- The tree will be of depth 20.
- Only 20 dimensions are tested – values of descriptor in other 108 dimensions are ignored!

Solutions proposed for closest search.

Scalable Recognition with a Vocabulary Tree

David Nistér and Henrik Stewénius
Center for Visualization and Virtual Environments
Department of Computer Science, University of Kentucky
http://www.vis.uky.edu/~dnister/ http://www.vis.uky.edu/~stewe/

Abstract

A recognition scheme that scales efficiently to a large number of objects is presented. The efficiency and quality is exhibited in a live demonstration that recognizes CD-covers from a database of 40000 images of popular music CD's.

The scheme builds upon popular techniques of indexing descriptors extracted from local regions, and is robust to background clutter and occlusion. The local region descriptors are hierarchically quantized in a vocabulary tree. The vocabulary tree allows a larger and more discriminatory vocabulary to be used efficiently, which we show experimentally leads to a dramatic improvement in retrieval quality. The most significant property of the scheme is that the tree directly defines the quantization. The quantization and the indexing are therefore fully integrated, essentially being one and the same.

The recognition quality is evaluated through retrieval on a database with ground truth, showing the power of the vocabulary tree approach, going as high as 1 million images.

1. Introduction

Object recognition is one of the core problems in computer vision, and it is a very extensively investigated topic. Due to appearance variabilities caused for example by non-rigidity, background clutter, differences in viewpoint, orientation, scale or lighting conditions, it is a hard problem.

One of the important challenges is to construct methods that scale well with the size of the database, and can select one out of a large number of objects in acceptable time. In this paper, a method handling a large number of objects is presented. The approach belongs to a currently very popular class of algorithms that work with local image regions and

This work was supported in part by the National Science Foundation under award number IIS-0545920, Faculty Early Career Development (CAREER) Program.

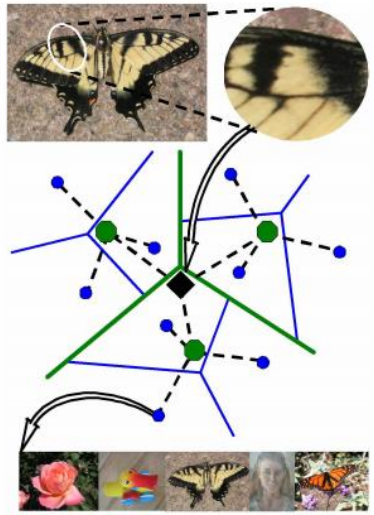
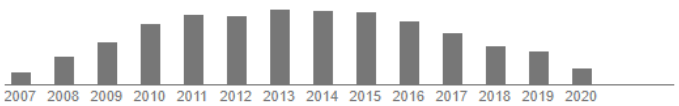


Figure 1. A vocabulary tree with branch factor three and only two levels for illustration purposes. A large number of elliptical regions are extracted from the image and warped to canonical positions. A descriptor vector is computed for each region. The descriptor vector is then hierarchically quantized by the vocabulary tree. In the first quantization layer, the descriptor is assigned to the closest of the three green centers. In the second layer, it is assigned to the closest of the three blue descendants to the green center. With each node in the vocabulary tree there is an associated inverted file with references to the images containing an instance of that node. The images in the database are scored hierarchically using the inverted files at multiple levels of the vocabulary tree.

Scalable recognition with a vocabulary tree

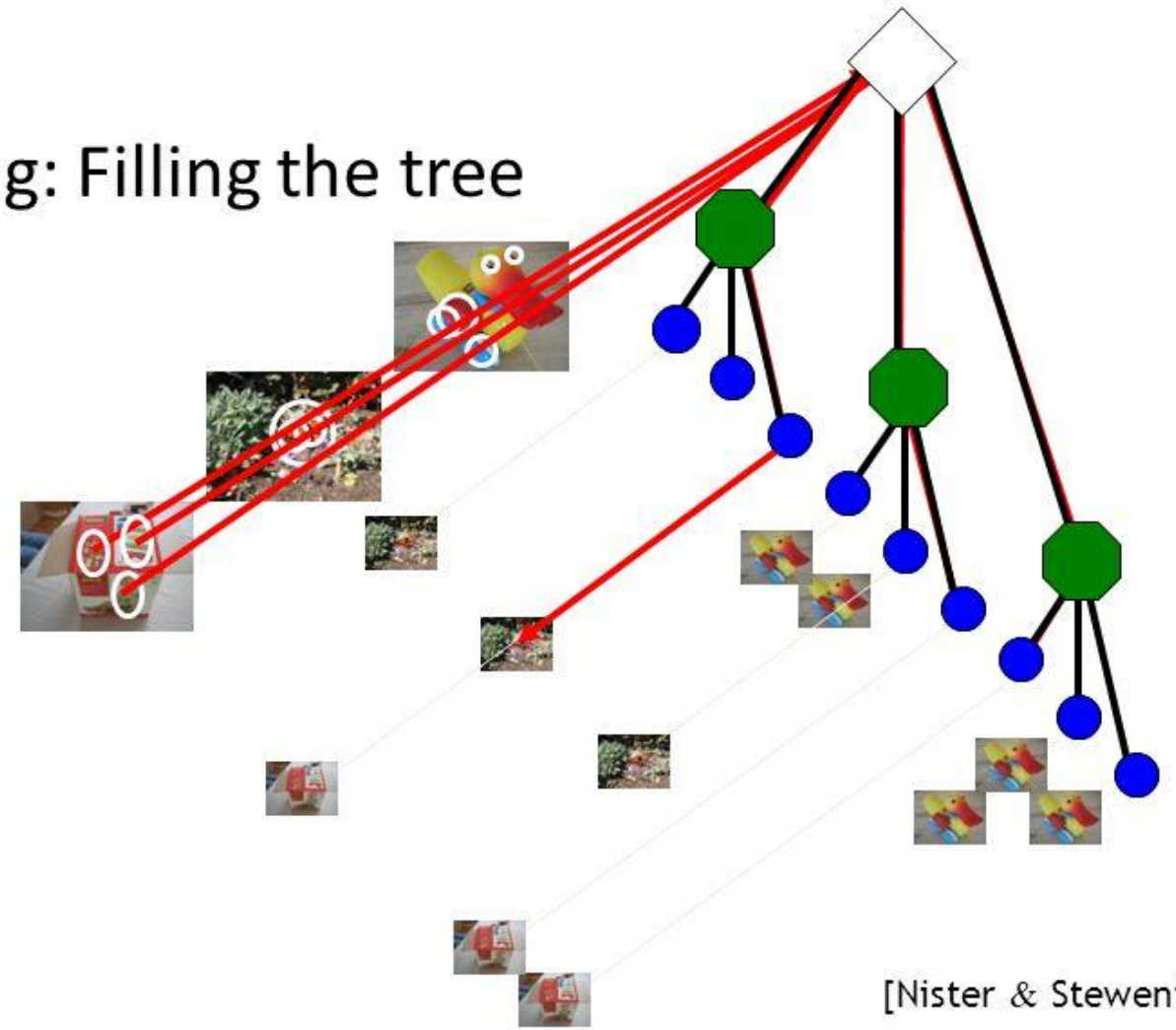
Authors	David Nister, Henrik Stewenius
Publication date	2006/6/17
Conference	2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
Volume	2
Pages	2161-2168
Publisher	Ieee
Description	A recognition scheme that scales efficiently to a large number of objects is presented. The efficiency and quality is exhibited in a live demonstration that recognizes CD-covers from a database of 40000 images of popular music CD's. The scheme builds upon popular techniques of indexing descriptors extracted from local regions, and is robust to background clutter and occlusion. The local region descriptors are hierarchically quantized in a vocabulary tree. The vocabulary tree allows a larger and more discriminatory vocabulary to be used efficiently, which we show experimentally leads to a dramatic improvement in retrieval quality. The most significant property of the scheme is that the tree directly defines the quantization. The quantization and the indexing are therefore fully integrated, essentially being one and the same. The recognition quality is evaluated through retrieval on a database with ground truth, showing ...
Total citations	Cited by 4360



Scholar articles
Scalable recognition with a vocabulary tree
D Nister, H Stewenius - 2006 IEEE Computer Society Conference on Computer ..., 2006
Cited by 4360 Related articles All 34 versions

Vocabulary Tree

Training: Filling the tree



[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Optimised KD-trees for fast image descriptor matching

Chanop Silpa-Anan
Seeing Machines, Canberra

Richard Hartley
Australian National University and NICTA.

Abstract

In this paper, we look at improving the KD-tree for a specific usage: indexing a large number of SIFT and other types of image descriptors. We have extended priority search, to priority search among multiple trees. By creating multiple KD-trees from the same data set and simultaneously searching among these trees, we have improved the KD-tree's search performance significantly. We have also exploited the structure in SIFT descriptors (or structure in any data set) to reduce the time spent in backtracking. By using Principal Component Analysis to align the principal axes of the data with the coordinate axes, we have further increased the KD-tree's search performance.

queries for image descriptors is the KD-tree [4], which is a form of balanced binary search tree.

Outline of KD-tree search. The general idea behind KD-trees is described now. The elements stored in the KD-tree are high-dimensional vectors in R^d. At the first level (root) of the tree, the data is split into two halves by a hyperplane orthogonal to a chosen dimension at a threshold value. Generally, this split is made at median in the dimension with the greatest variance in the data set. By comparing the query vector with the partitioning value, it is easy to determine to which half of the data the query vector belongs. Each of the two halves of the data is then recursively split in the same way to create a fully balanced binary tree. At the bottom of the tree, each tree node corresponds to a single point in the

Optimised KD-trees for fast image descriptor matching

Authors	Chanop Silpa-Anan, Richard Hartley
Publication date	2008/6/23
Conference	2008 IEEE Conference on Computer Vision and Pattern Recognition
Pages	1-8
Publisher	IEEE
Description	In this paper, we look at improving the KD-tree for a specific usage: indexing a large number of SIFT and other types of image descriptors. We have extended priority search, to priority search among multiple trees. By creating multiple KD-trees from the same data set and simultaneously searching among these trees, we have improved the KD-treepsilas search performance significantly.We have also exploited the structure in SIFT descriptors (or structure in any data set) to reduce the time spent in backtracking. By using Principal Component Analysis to align the principal axes of the data with the coordinate axes, we have further increased the KD-treepsilas search performance.