Geometric Computing and Computer Vision

Feature extraction

Oleg Voynov
slides and images borrowed from a variety of sources, incl. slides by Nikolai Poliarnyi and others

COLMAP

DEMO

If you don't understand something, just ask

COLMAP Pipeline

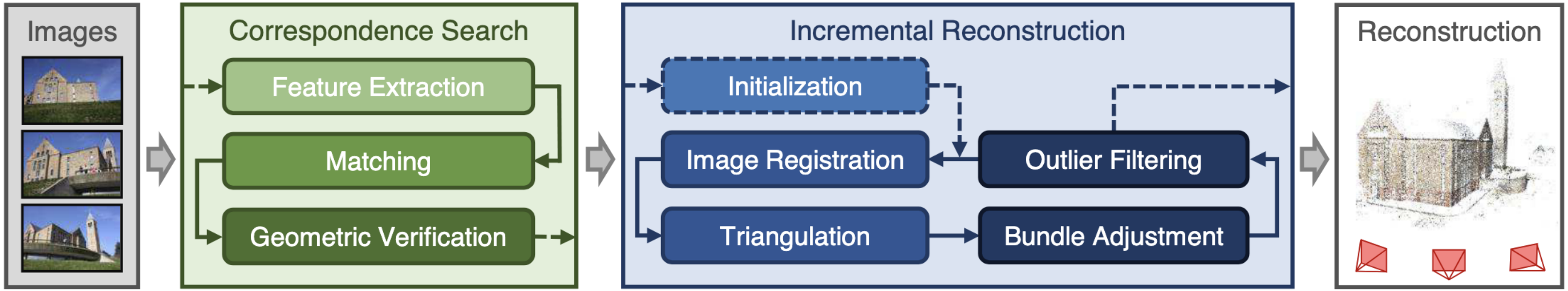


Figure 2. Incremental Structure-from-Motion pipeline.

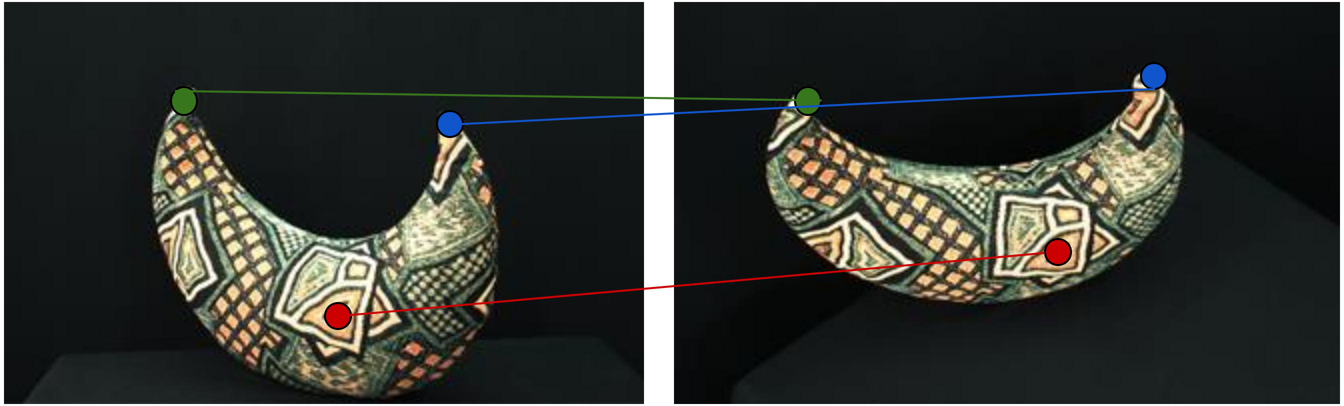
Class objectives: Feature extraction

- Learn what is a feature, what are the properties of a good feature
- Learn SIFT feature extraction method
- Try SIFT in OpenCV, learn some basic method from numpy, matplotlib, PIL

COLMAP

DEMO

KEY POINTS



First view of image

Second view of image

we want to find the correspondences between same points of the different views of the object

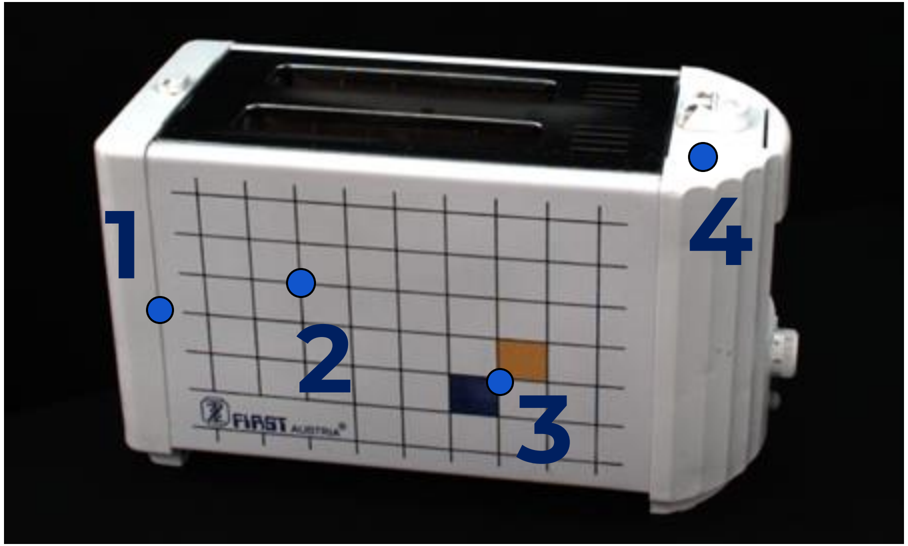
WHAT ARE THE GOOD POINTS?

KEY POINTS

TOASTER

FOX

KEY POINTS

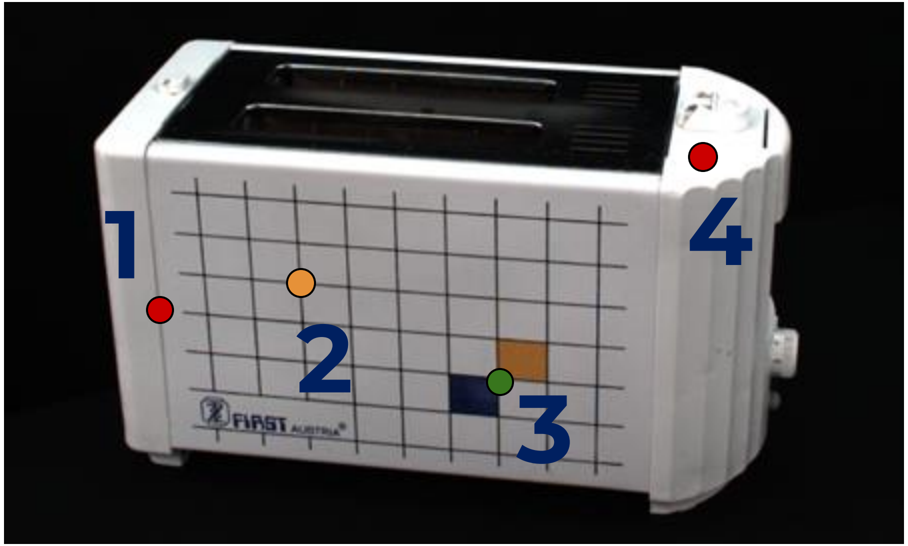


TOASTER

FOX

Which parts we can easily identify and differentiate from the rest of the object?

KEY POINTS

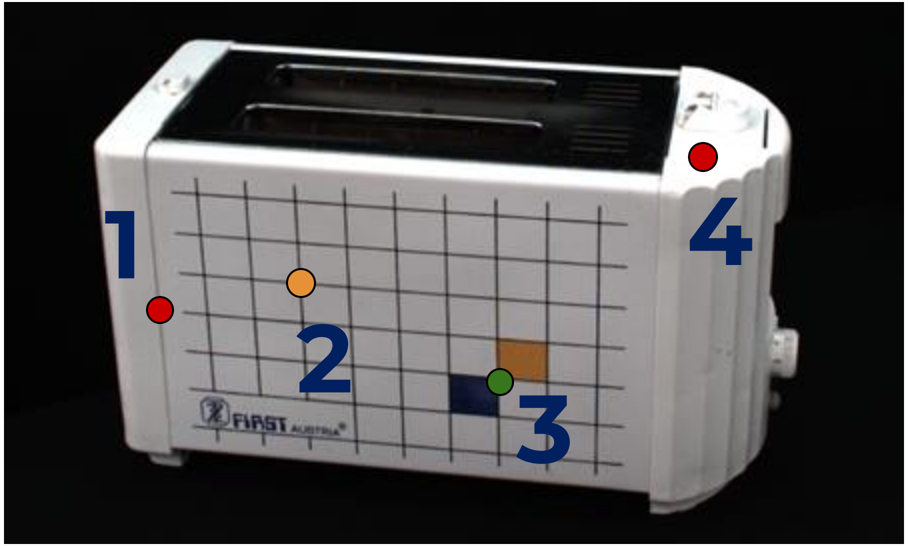


TOASTER

FOX

- BAD
- Conditionally BAD
- GOOD

KEY POINTS



TOASTER

FOX

- BAD
- Conditionally BAD
- GOOD

good key points: corners and blobs.

WHAT KIND OF DISTORTIONS DO WE HAVE?

CHALLENGES

We require a method for detecting blob-like keypoints and a descriptor that is invariant to such image distortions

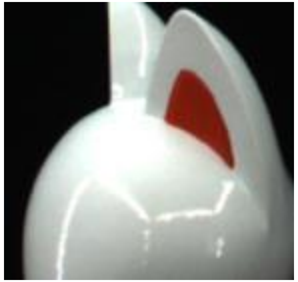
Q: What types of distortions can occur due to a change in viewpoint?

WHAT KIND OF DISTORTIONS DO WE HAVE ?

CHALLENGES

We require a method for detecting blob-like keypoints and a descriptor that is invariant to such image distortions

Original Image



Scaling

Rotation

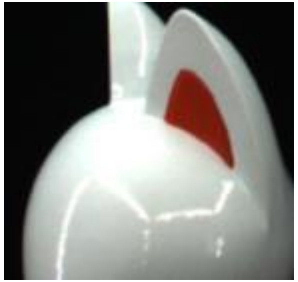
Lighting

WHAT KIND OF DISTORTIONS DO WE HAVE ?

CHALLENGES

We require a method for detecting blob-like keypoints and a descriptor that is invariant to such image distortions

Original Image



Scaling

Rotation

Lighting

The SIFT algorithm is robust to these changes

WHAT KIND OF DISTORTIONS DO WE HAVE ?

CHALLENGES

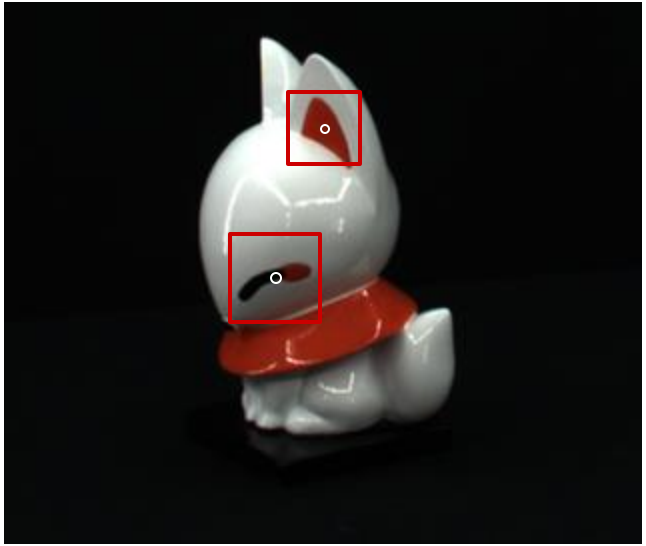
Extreme change of perspective

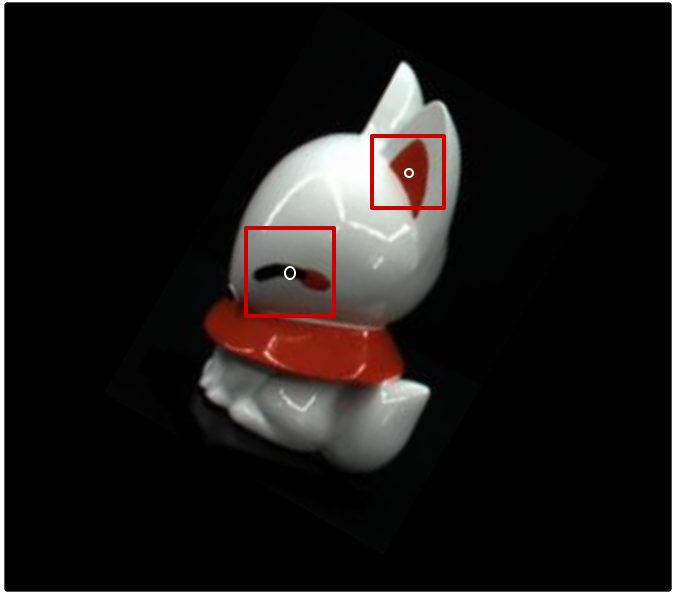
Extreme changes of lightning, weather

The SIFT algorithm is NOT robust to these changes

INVARIANT TO

ROTATIONS

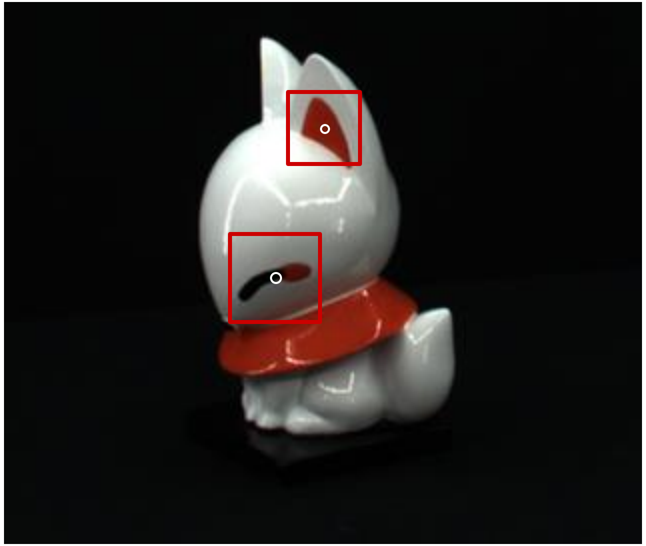


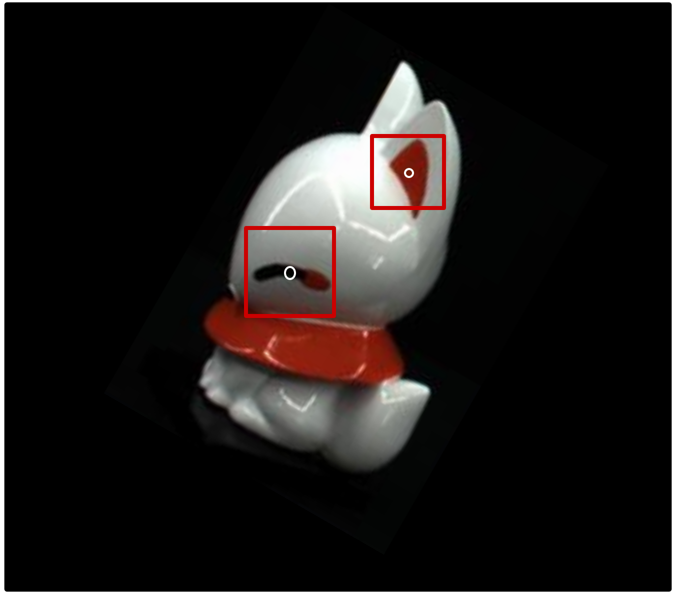


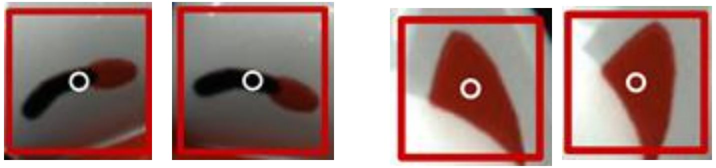
Q: If we take a patch and extract the color information from it in vector form, will it be invariant to rotation?

INVARIANT TO

ROTATIONS



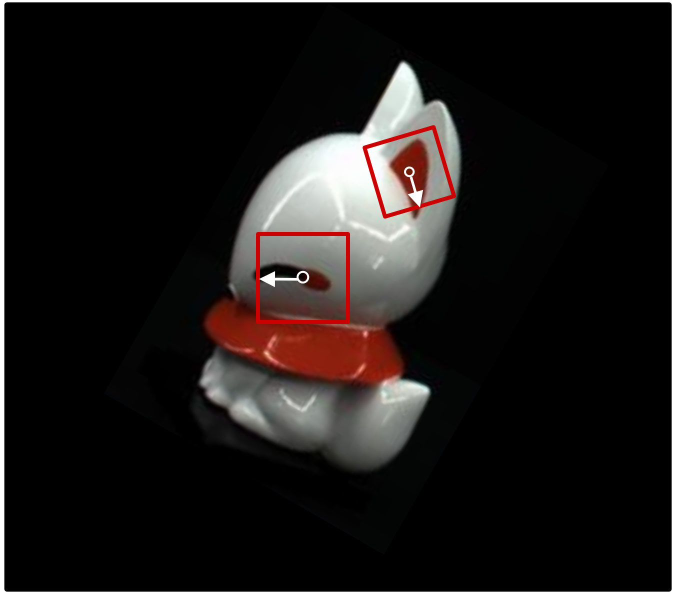




Q: If we take a patch and extract the color information from it in vector form, will it be invariant to rotation?

NO!

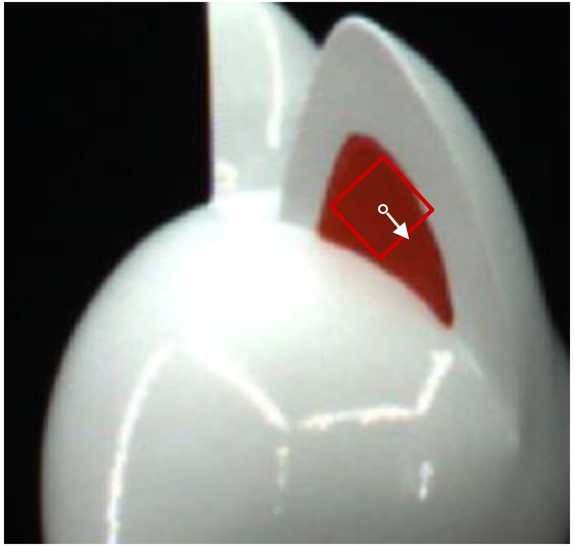
ROTATIONS



You need to be able to find the **“canonical” direction** and take patches along it

INVARIANT

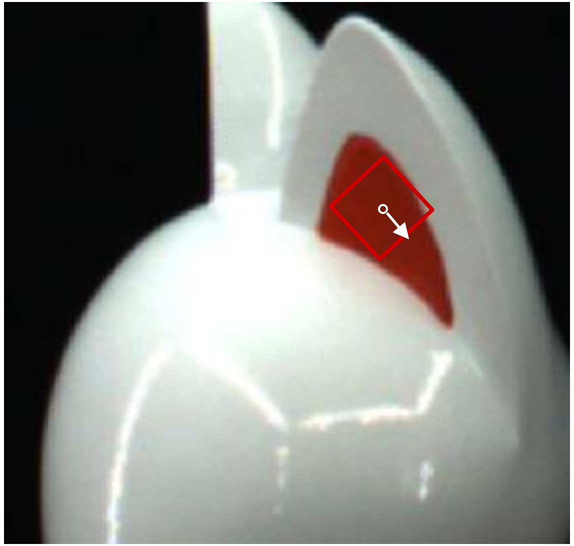
SCALE



Q: If we take a fixed size patch, is it invariant to scale change?

INVARIANT

SCALE

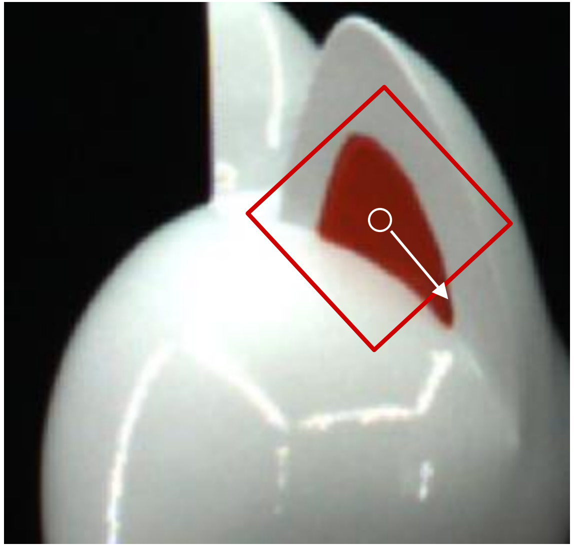


Q: If we take a fixed size patch, is it invariant to scale change?

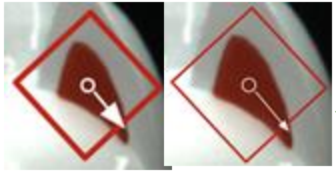
NO!

INVARIANT

SCALE



You need to adjust the patch size to the scale of the image



YES!

Light Conditions

Q: How we can achieve invariance to light conditions?

SCALE-INVARIANT FEATURE TRANSFORM

SIFT

Distinctive Image Features from Scale-Invariant Keypoints

David G. Lowe
Computer Science Department
University of British Columbia
Vancouver, B.C., Canada
lowe@cs.ubc.ca

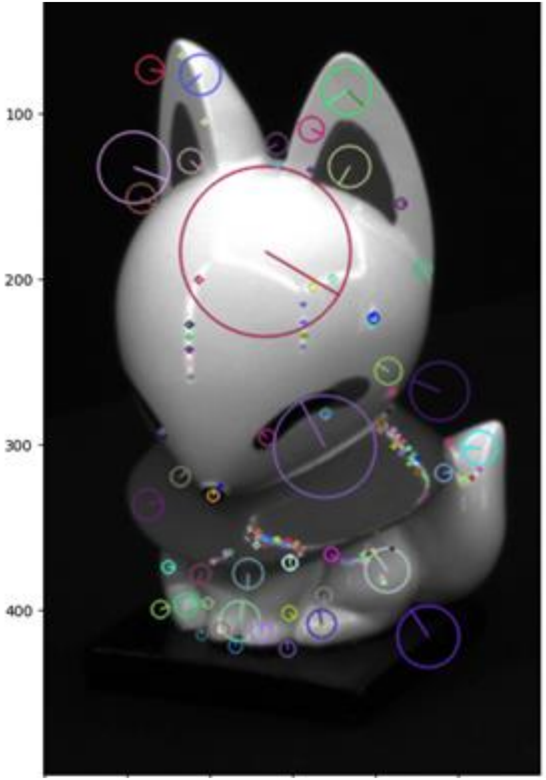
January 5, 2004

Abstract

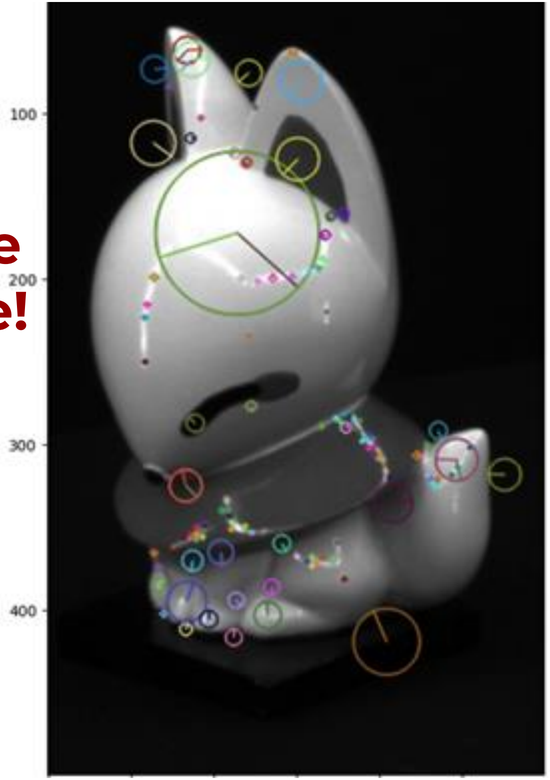
paper link

LET'S SEE SOME EXAMPLES!

SIFT



Code time!



Q:

Now, if we have these keypoints with 128-dimensional signatures on each image, how do we find points that correspond to the same locations of the object in different views?

Comparing SIFT DESCRIPTORS

Essentially comparing two arrays of data.

Let $H_1(k)$ and $H_2(k)$ be two arrays of data of length N.

L2 Distance:

$$d(H_1, H_2) = \sqrt{\sum_k (H_1(k) - H_2(k))^2}$$

Smaller the distance metric, better the match.

Perfect match when $d(H_1, H_2) = 0$.

Comparing SIFT DESCRIPTORS

Essentially comparing two arrays of data.

Let $H_1(k)$ and $H_2(k)$ be two arrays of data of length N.

Normalized Correlation:

$$d(H_1, H_2) = \frac{\sum_k [(H_1(k) - \overline{H_1})(H_2(k) - \overline{H_2})]}{\sqrt{\sum_k (H_1(k) - \overline{H_1})^2 \sum_k (H_2(k) - \overline{H_2})^2}}$$

Where:

$$\overline{H_i} = \frac{1}{N} \sum_{k=1}^{N} H_i(k)$$

Larger the distance metric, better the match.

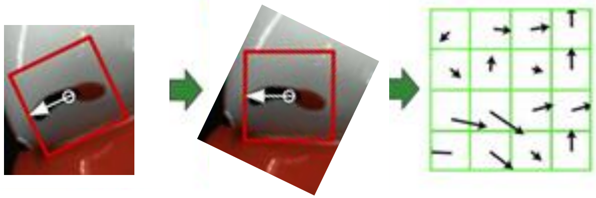
Perfect match when $d(H_1, H_2) = 1$.

INVARIANT

SCALE

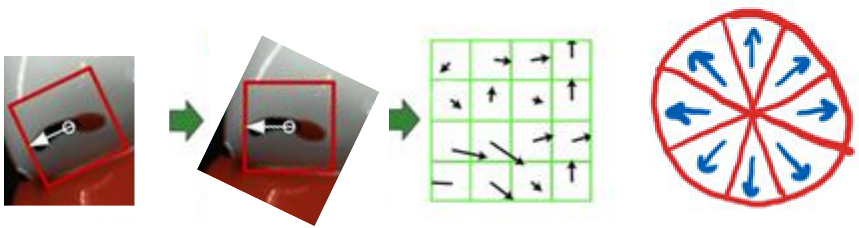
- Invariance to rotation
- Invariance to scale

HOG



- **Invariance to rotation**
- **Invariance to scale**

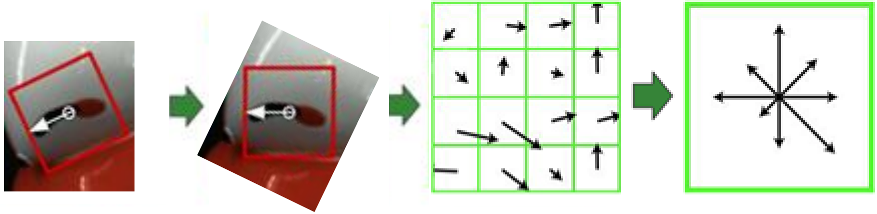
HOG



- Invariance to rotation
- Invariance to scale

That is, all gradients vote for 8 directions,
i.e., 8 orientation bins in the histogram.

HOG



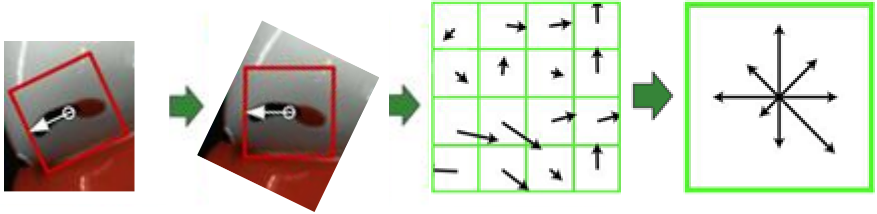
Histogram of Gradients (HoG)

What image changes do not affect HOG?

- **Invariance to rotation**
- **Invariance to scale**

also a hog

HOG



Histogram of Gradients (HoG)

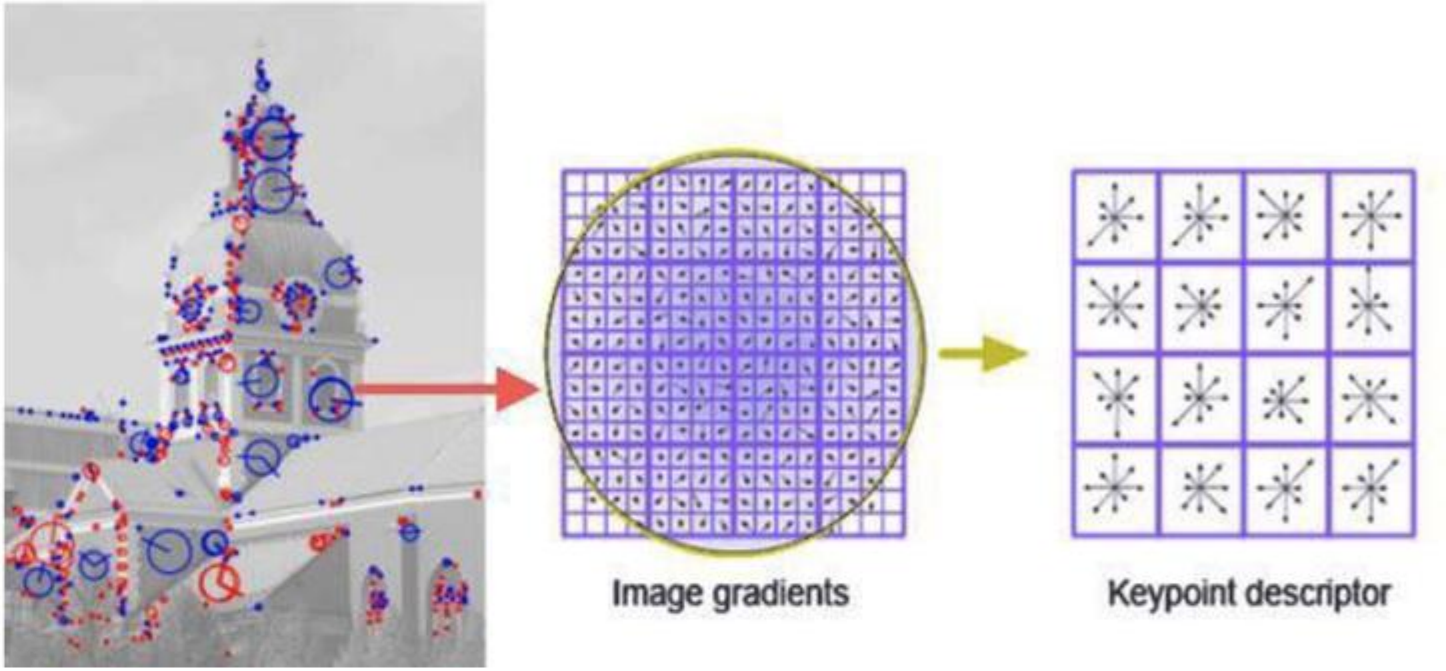
- **Invariance to rotation**
- **Invariance to scale**

What image changes do not affect HOG?

- Multiplicative changes in brightness
- Additive changes in brightness
- Small shifts or translations — do not change HOG significantly

also a hog

DESCRIPTORS



https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774

DETECTOR
&
DESCRIPTORS

How do we determine the canonical scale and orientation, and how do we detect keypoints for description?

1) **We select blobs using key points**
2) **For each point we invariantly define:**
   a) **Rotation**
   b) **Scale**

How to select blobs?

KEY POINTS

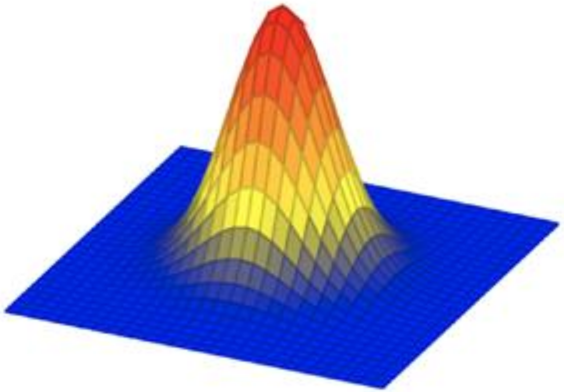
BLUR

KEY POINTS

GAUSSIAN FILTER

In the continuous domain:

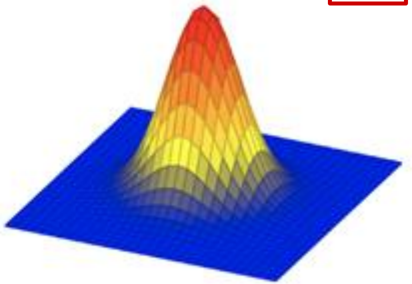
$$g(x,y;\sigma) = \frac{1}{2\pi\sigma^2} \exp -\frac{x^2+y^2}{2\sigma^2}$$

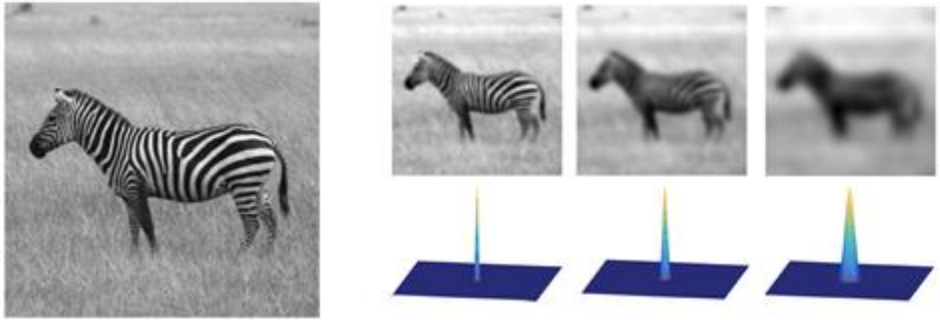


BLUR

In the continuous domain:

$$g(x,y;\sigma) = \frac{1}{2\pi\sigma^2} \exp -\frac{x^2+y^2}{\boxed{2\sigma^2}}$$





QUESTION

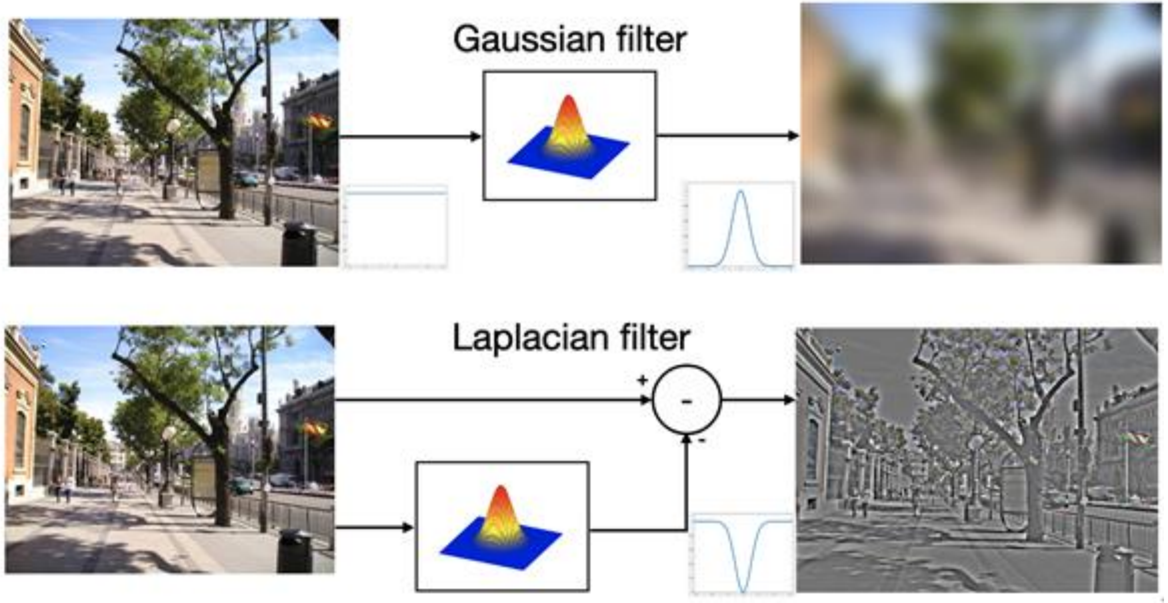
What happens
if we subtract one image
from another?

-

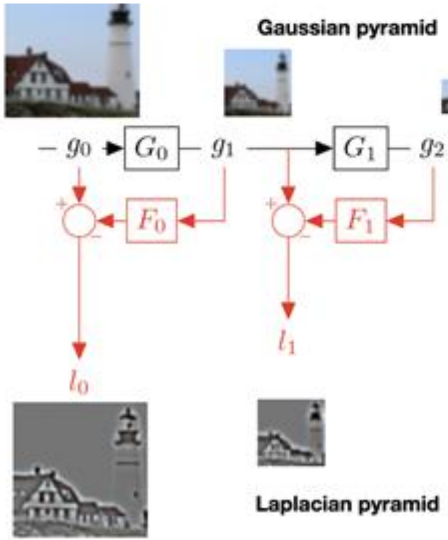
= ?

BLURRING

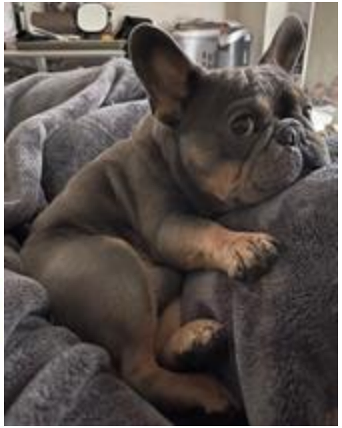
What about the opposite of blurring?



DOG

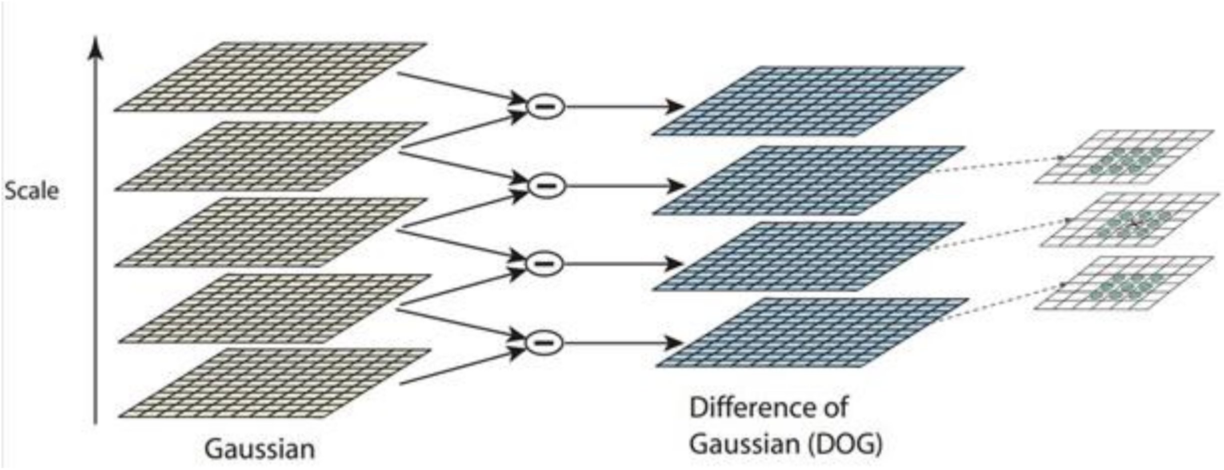


or
Differences of Gaussians (DoG)



also a dog

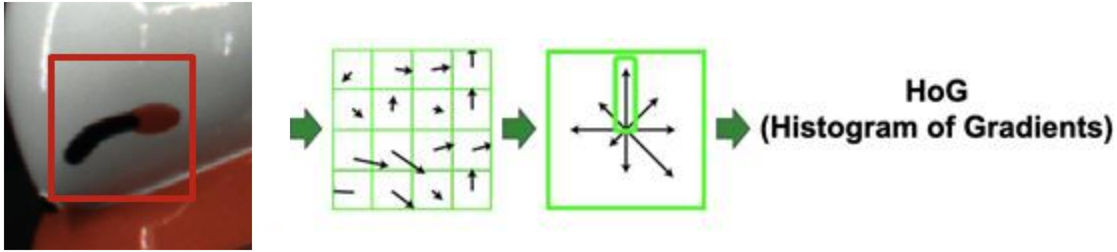
SIFT PROCESS



DETECTOR
&
DESCRIPTORS

- **Detect keypoints that correspond to distinct features**
- **Get their canonical scale using DoG**

- **How to find the canonical orientation**



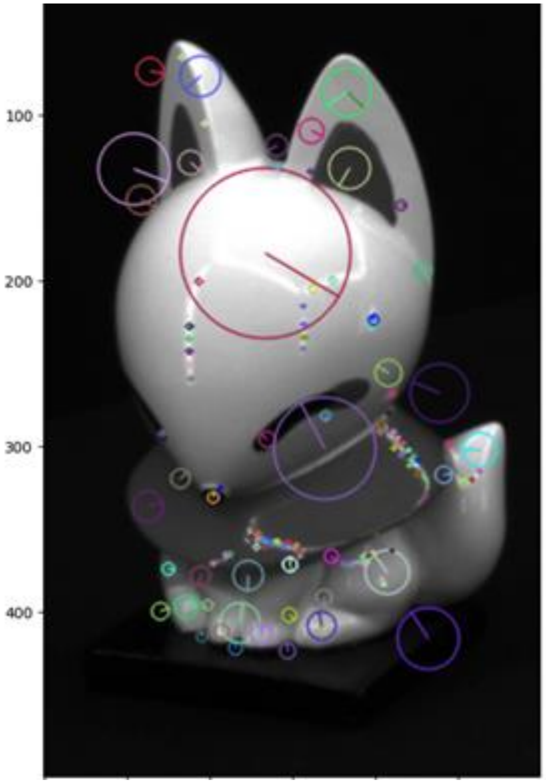
DETECTOR
&
DESCRIPTORS

SIFT algorithm:

- Build the DoG pyramid: a set of images, each containing details at a specific scale.
- Detect distinctive features as extrema in the DoG.
- Determine the canonical scale as the scale at which the extremum is found.
- To find the canonical orientation, take the non-canonical neighborhood, compute its HoG, and select the dominant orientation.
- To compute the descriptor, take the canonical, rotated, and scaled neighborhood, divide it into 16 regions, compute HoG in each one, and use the collection of HoGs as the descriptor.

LET'S SEE SOME EXAMPLES!

SIFT



Code time!