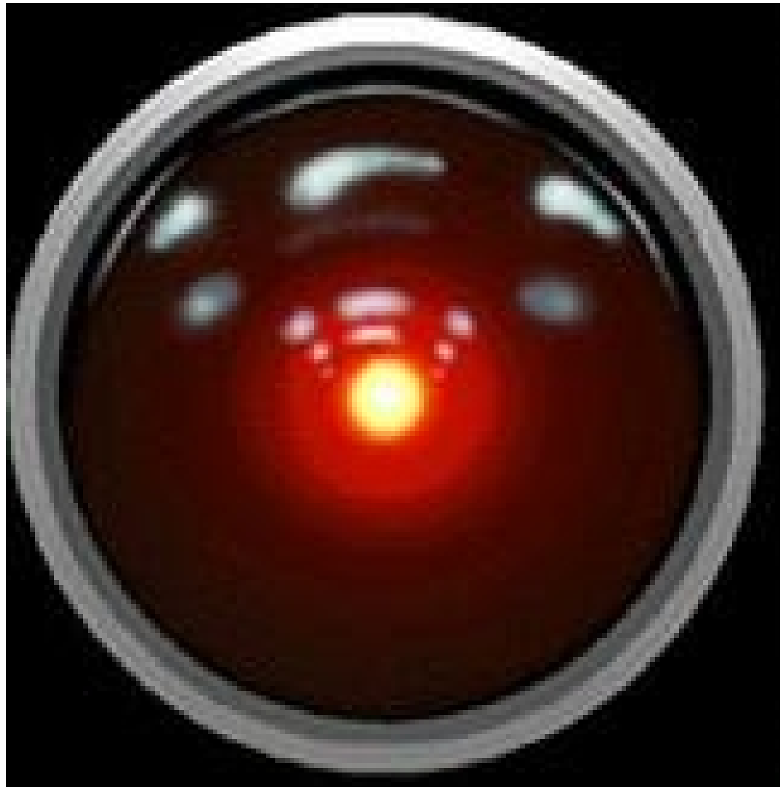
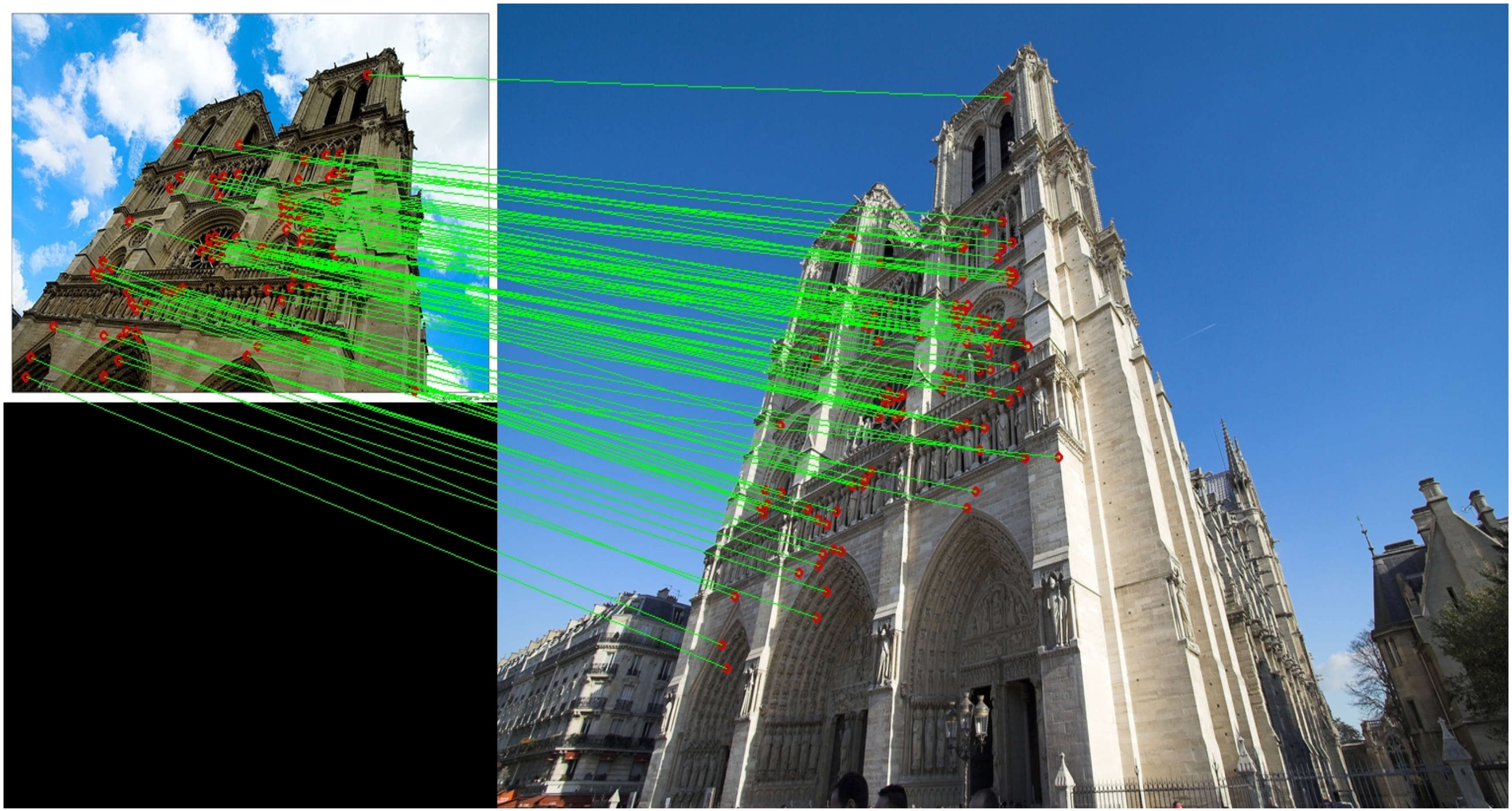


CS 378: Autonomous Intelligent Robotics

Instructor: Jivko Sinapov

http://www.cs.utexas.edu/~jsinapov/teaching/cs378/

Visual Registration and Recognition



Announcements

Homework 6 is out, due ~~4/5~~ 4/7

Installing our code base

- Github page:
  - https://github.com/utexas-bwi/bwi

Getting your project code up on github

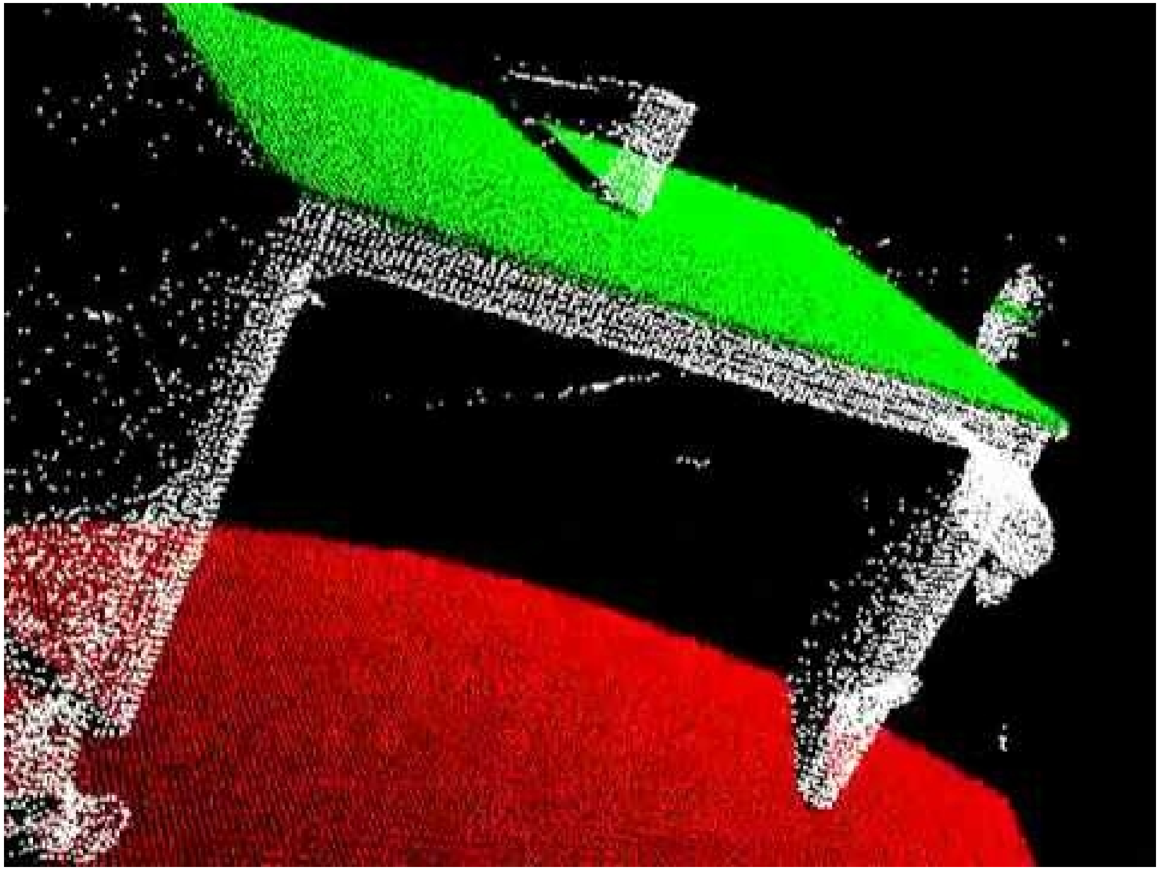
GitHub guide:

https://guides.github.com/activities/hello-world/

Reading Discussion

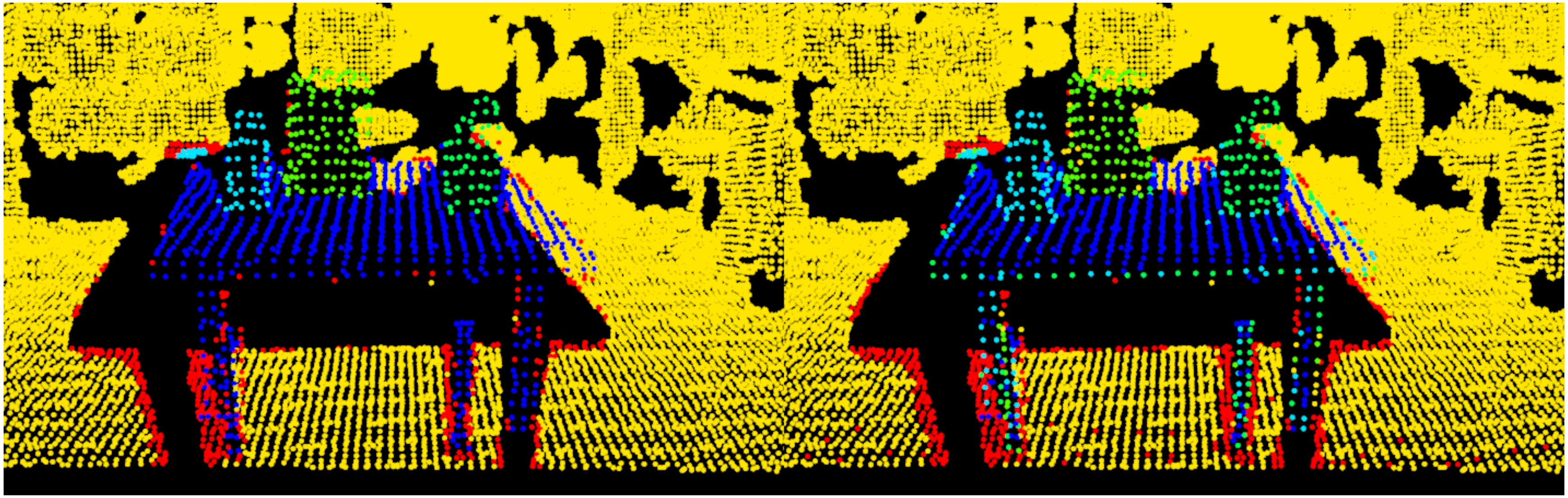
"I would like to see more examples of how to use PCL. I thought the two examples given in the paper were interesting and helped me to understand why PCL is needed. However, I would like to know more about how to implement those processes in a project. I also want to see what else the PCL is capable of doing."

Plane Detection in PCL



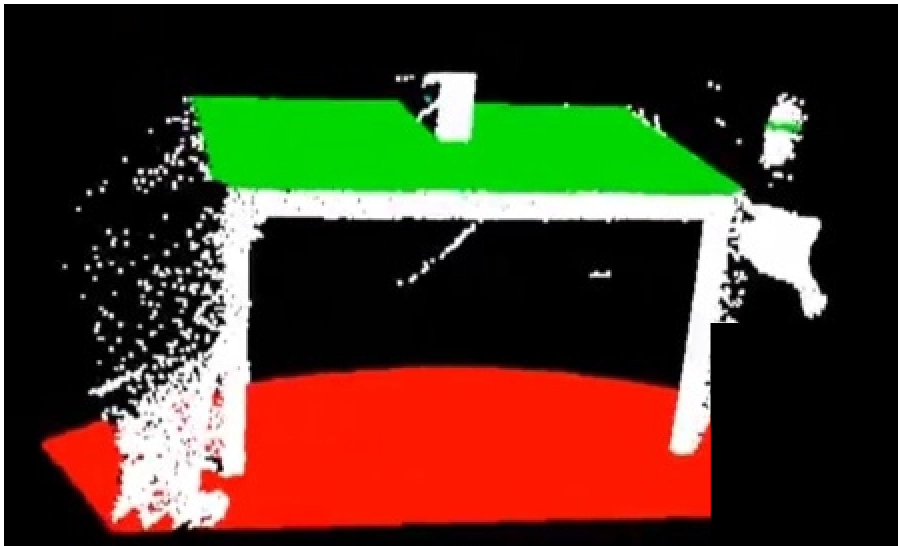
http://www.pointclouds.org/documentation/tutorials/planar_segmentation.php

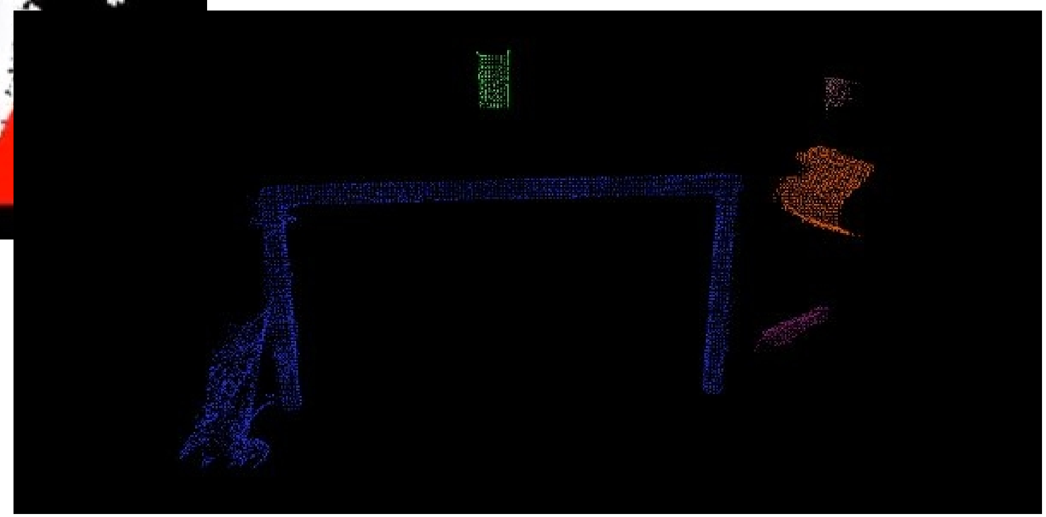
Voxel Grid Filter in PCL



http://pointclouds.org/documentation/tutorials/voxel_grid.php

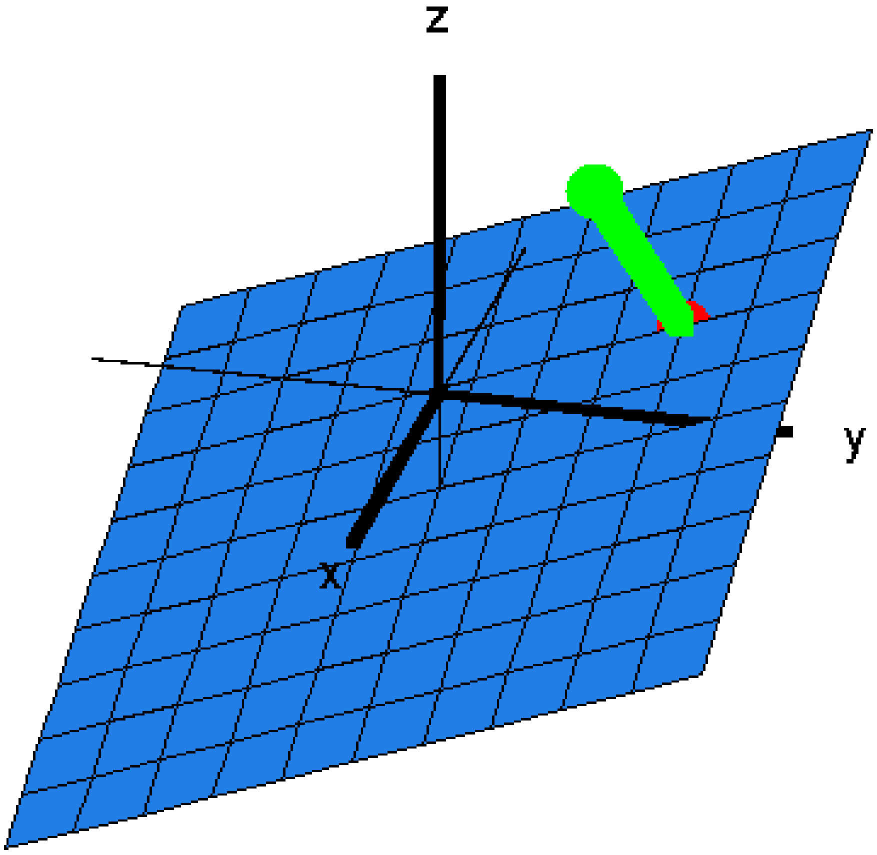
Euclidean Cluster Extraction





http://www.pointclouds.org/documentation/tutorials/cluster_extraction.php

Computing Point Normals



Computing Point Normals

- Tutorial http://pointclouds.org/documentation/tutorials/normal_estimation.php

- Videos:

http://www.youtube.com/watch?v=x1FSssJrfik

http://www.youtube.com/watch?v=d3z35hpwPZQ

Reading Discussion

- What is the maintenance of such a system like? How was it even created in the first place?
- How do people see? Do robots ever currently see better than people, as they were occasionally able to better identify objects by sound?
- Could the PCL be used to enhance human vision? Are there applications for it in virtual reality?

Reading Discussion

- What is the maintenance of such a system like? How was it even created in the first place?
- How do people see? Do robots ever currently see better than people, as they were occasionally able to better identify objects by sound?
- Could the PCL be used to enhance human vision? Are there applications for it in virtual reality?

Reading Discussion

"What exactly is SSE optimization? How does this affect performance with regards to point clouds a robot would use?"

Answer:

http://neilkemp.us/src/sse_tutorial/sse_tutorial.html

Reading Discussion

Readings for this week

1) Alex Smola and S.V.N. Vishwanathan, Introduction to Machine Learning, Chapter 1, Cambridge University Press, 2008

Readings for this week

In addition, this week, you get to pick a published, peer-reviewed conference or journal article.

Your reading response should be about your pick.

Robotics and AI Conferences

- IEEE International Conference on Robotics and Automation (ICRA)
- IEEE International Conference on Intelligent Robots (IROS)
- IEEE International Conference on Development and Learning (ICDL)
- Robotics Science and Systems (RSS)

Robotics and AI Conferences (con't)

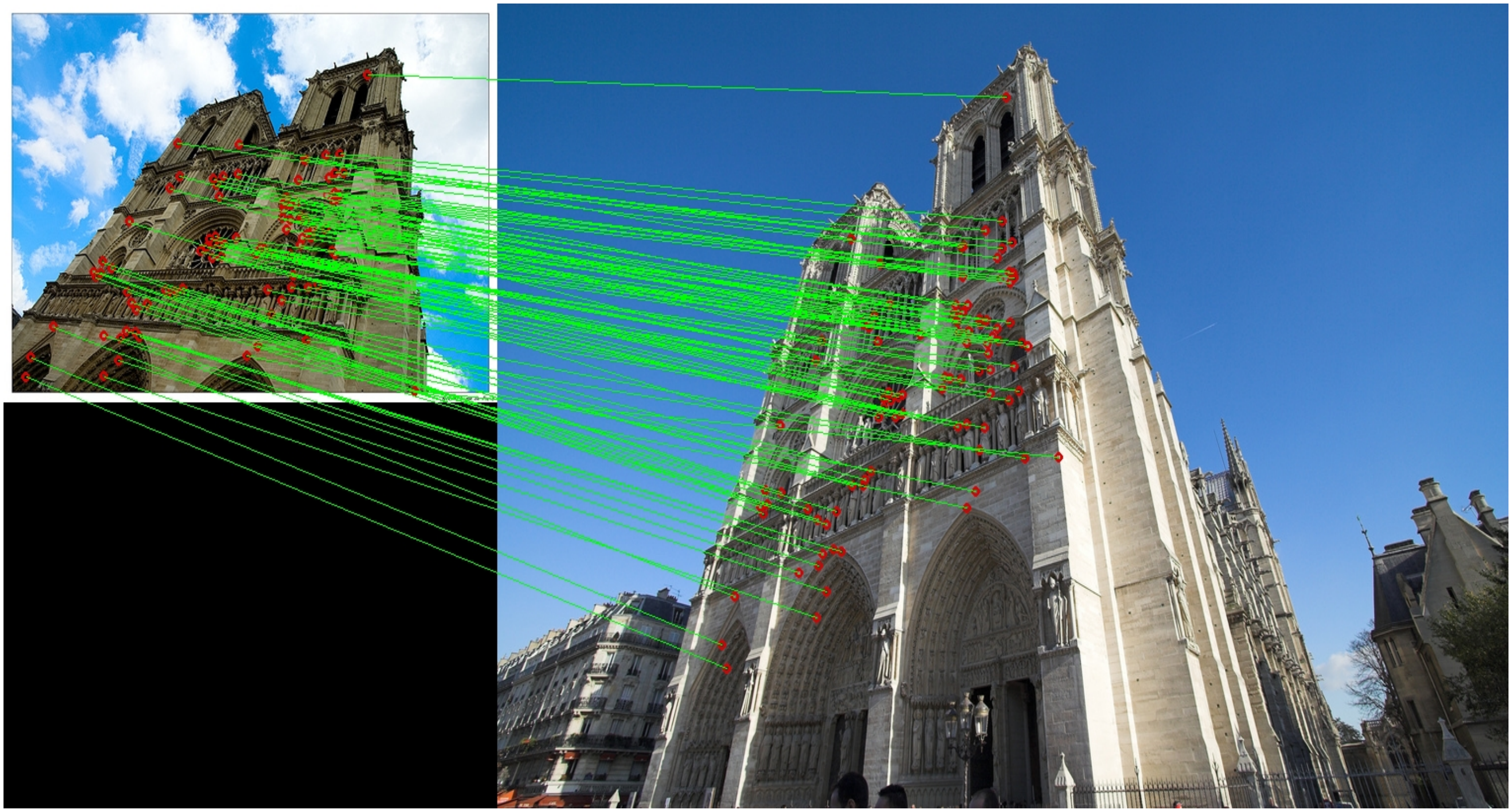
- ACM / IEEE International Conference on Human-Robot Interaction (HRI)
- International Conference on Social Robotics (ICSR)
- AAAI Conference on Artificial Intelligence (AAAI)
- International Joint Conference on Artificial Intelligence (IJCAI)

Robotics Journals

- IEEE Transactions on Robotics (TRO)
- IEEE Transactions on Autonomous Mental Development (TAMD)
- International Journal of Robotics Research (IJRR)
- Robotics and Autonomous System (RAS)

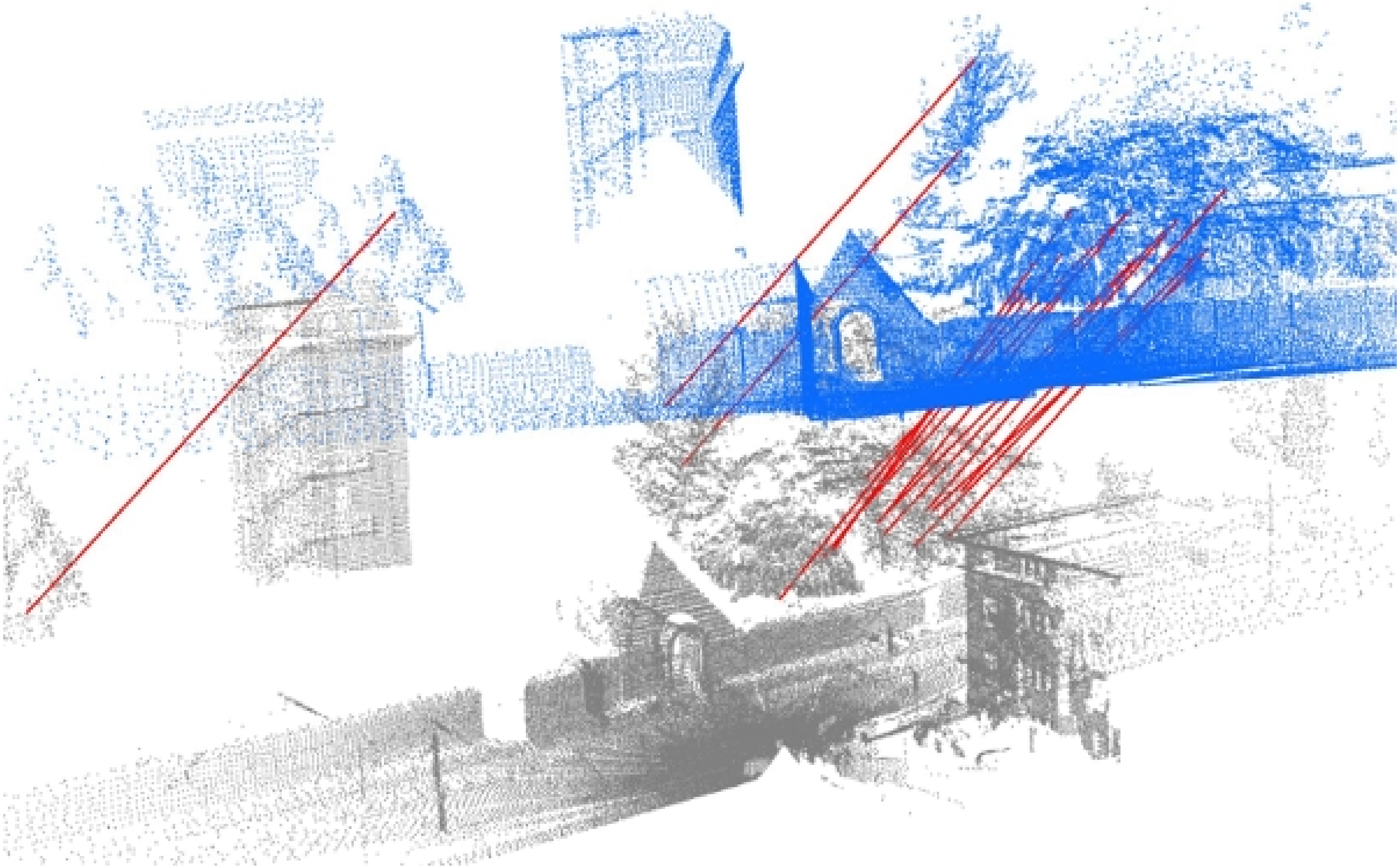
A brief tour of google scholar

Visual Registration and Recognition

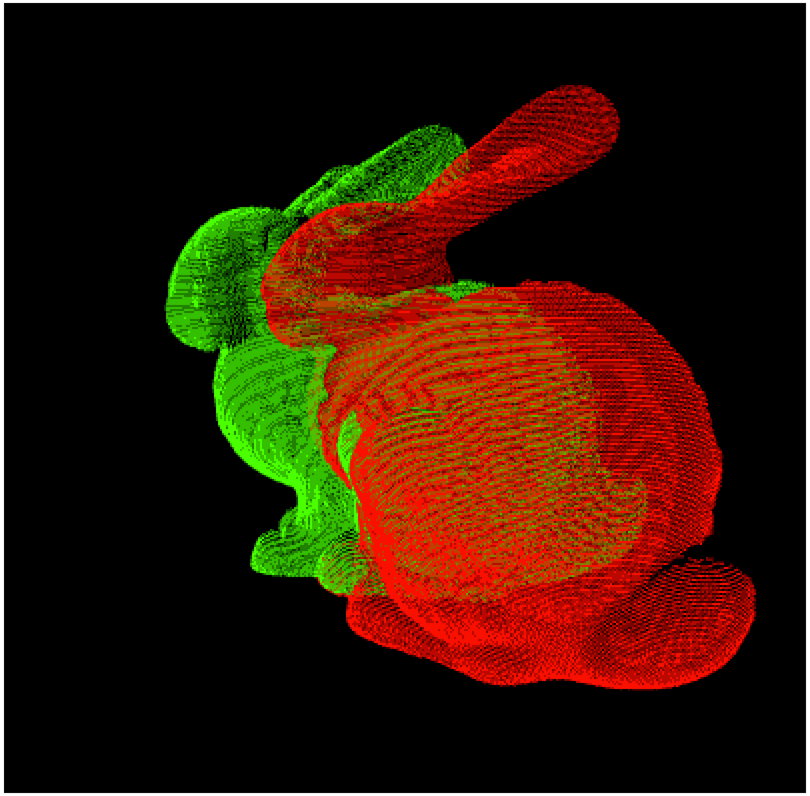


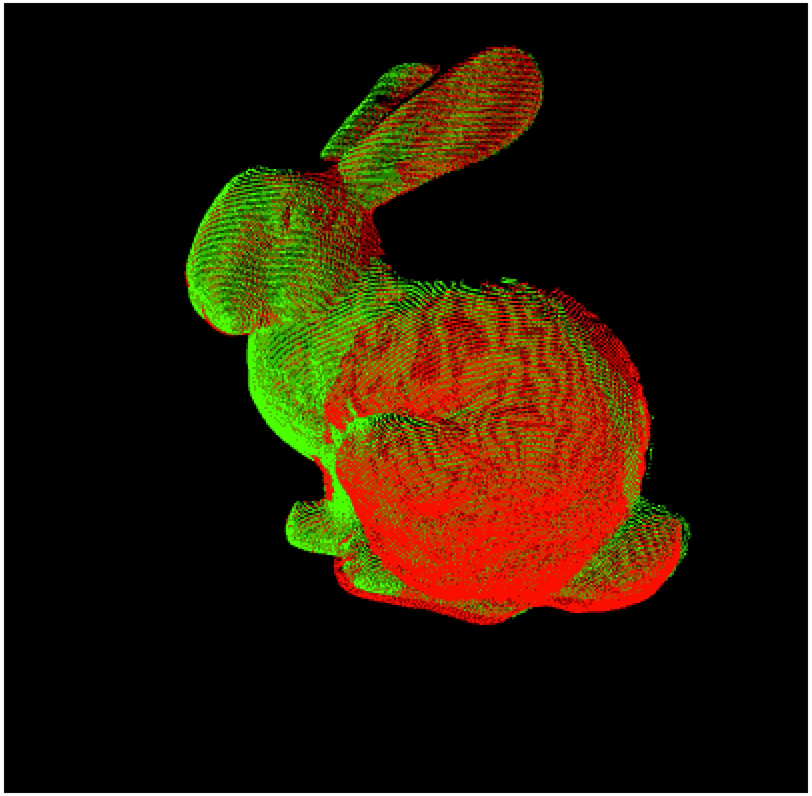
Visual Registration and Recognition

Registration



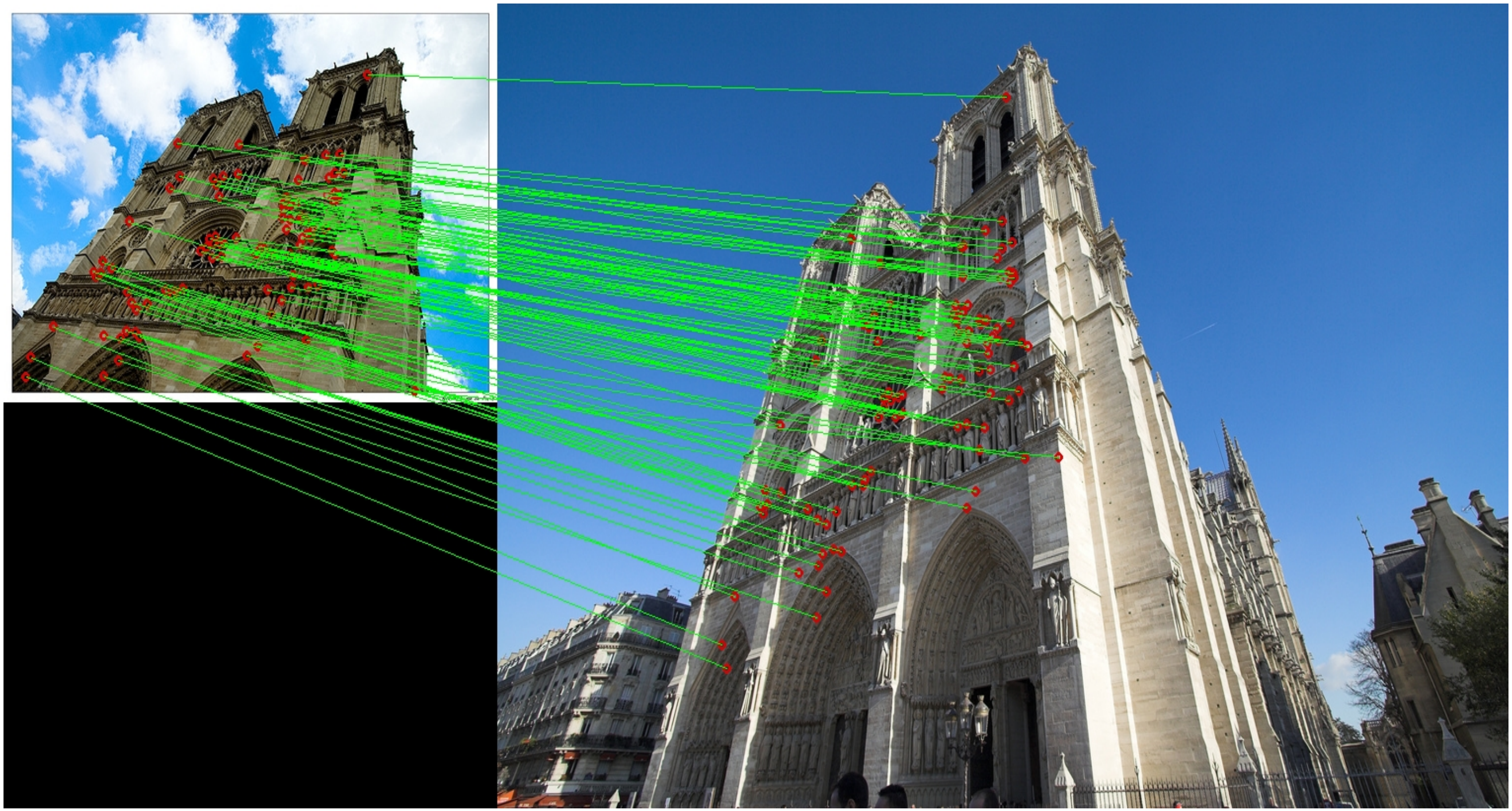
Registration



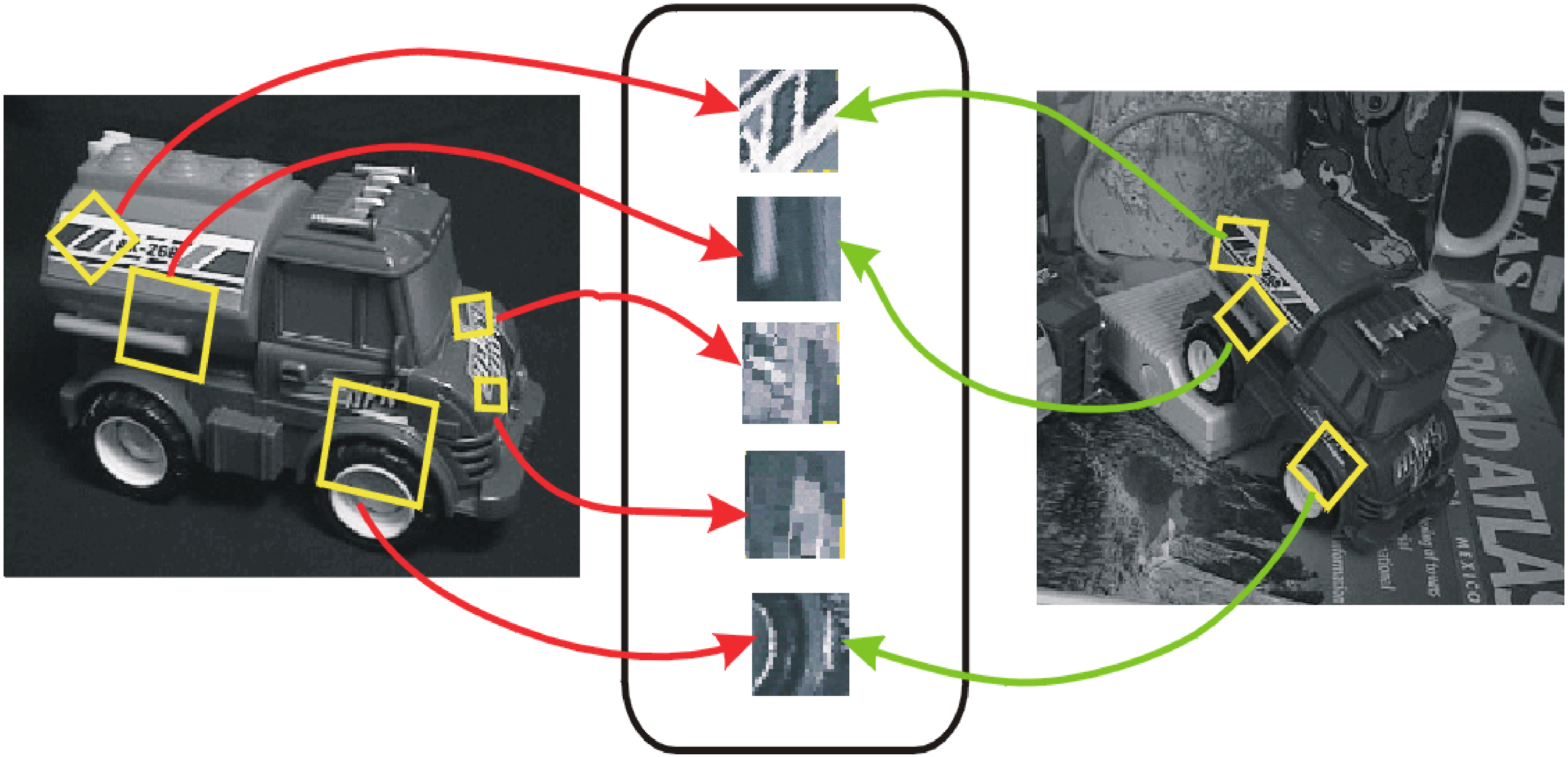


[http://vihari.github.io/personal_website/images/3dregistration.png]

Registration



Interest Point Registration

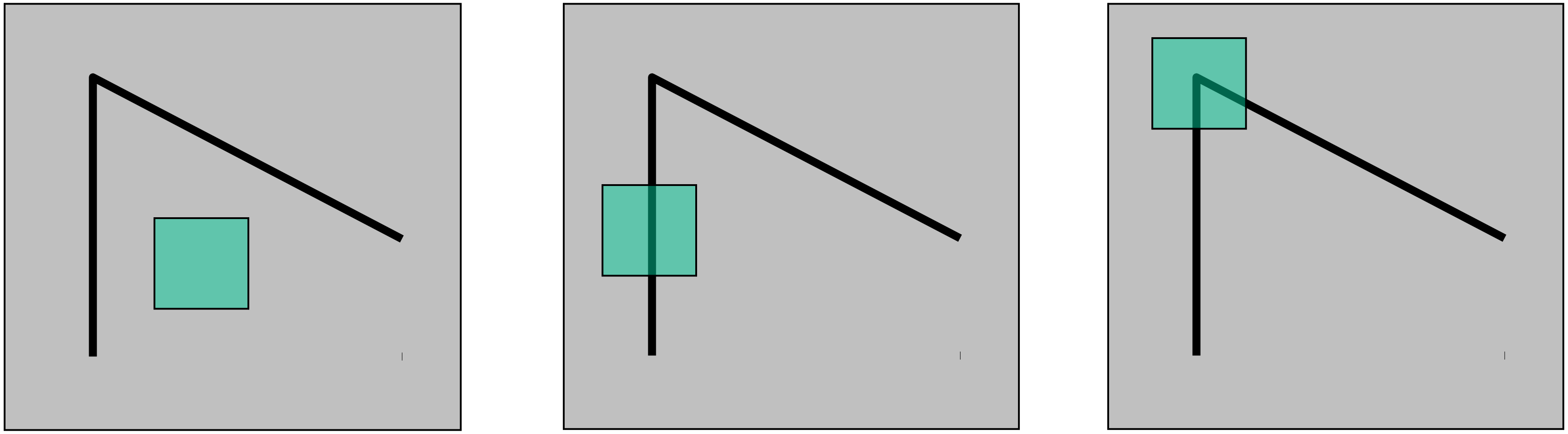


Interest Point Detection

- Look for image regions that are unusual
  - Leads to unambiguous matches in other images
  - How do we define unusual?

Suppose we only consider a small window of pixels

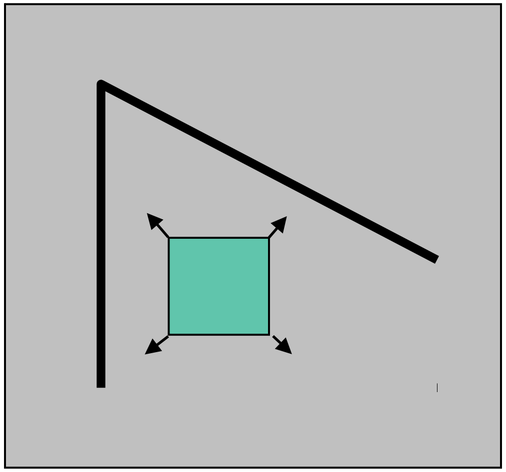
- What defines whether a feature is a good or bad candidate?

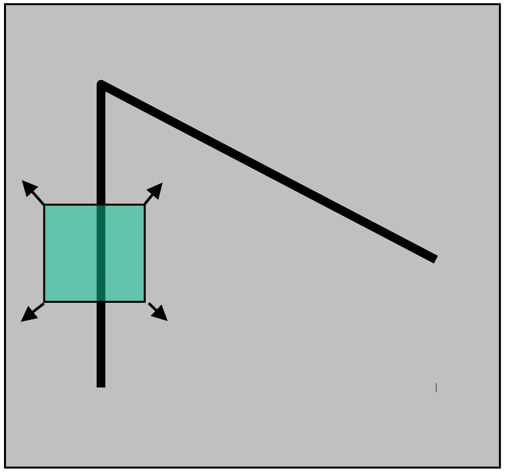


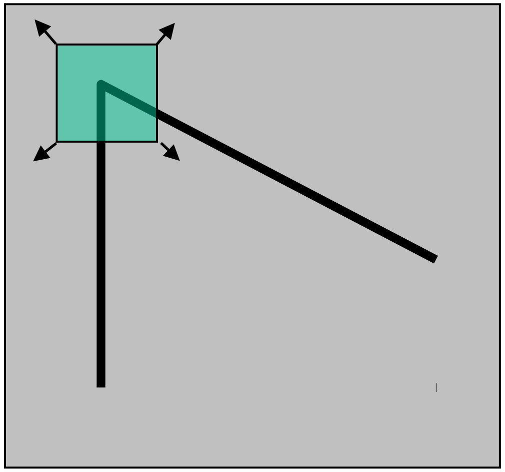
Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

Local measure of feature uniqueness

- How does the window change when you shift it?
- Shifting the window in *any direction* causes a *big change*







"flat" region:
no change in all directions

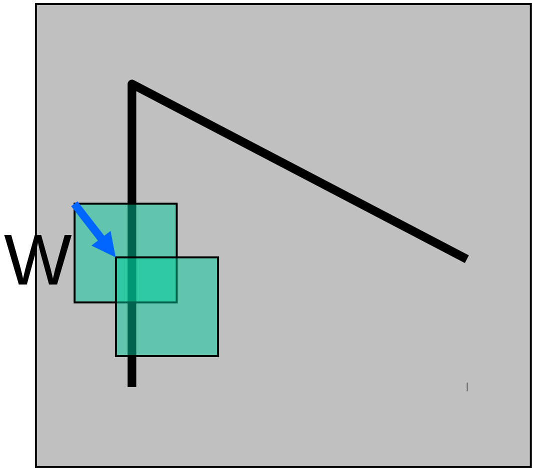
"edge":
no change along the edge direction

"corner":
significant change in all directions

Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

Consider shifting the window W by (u,v)

- how do the pixels in **W** change?
- compare each pixel before and after by summing up the squared differences (SSD)
- this defines an SSD "error" of $E(u,v)$:



Harris Detector: Mathematics

Change of intensity for the shift [*u*,*v*]:

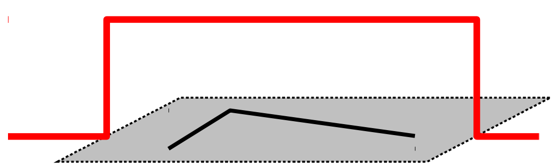
$$E(u,v)=\sum_{x,y} w(x,y)\left[\,I(x+u,y+v)-I(x,y)\,\right]^2$$

Window function

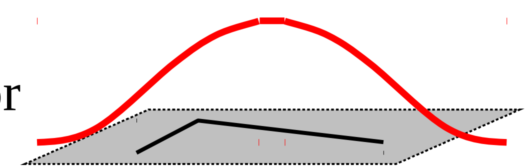
Shifted intensity

Intensity

Window function $w(x,y)$ =

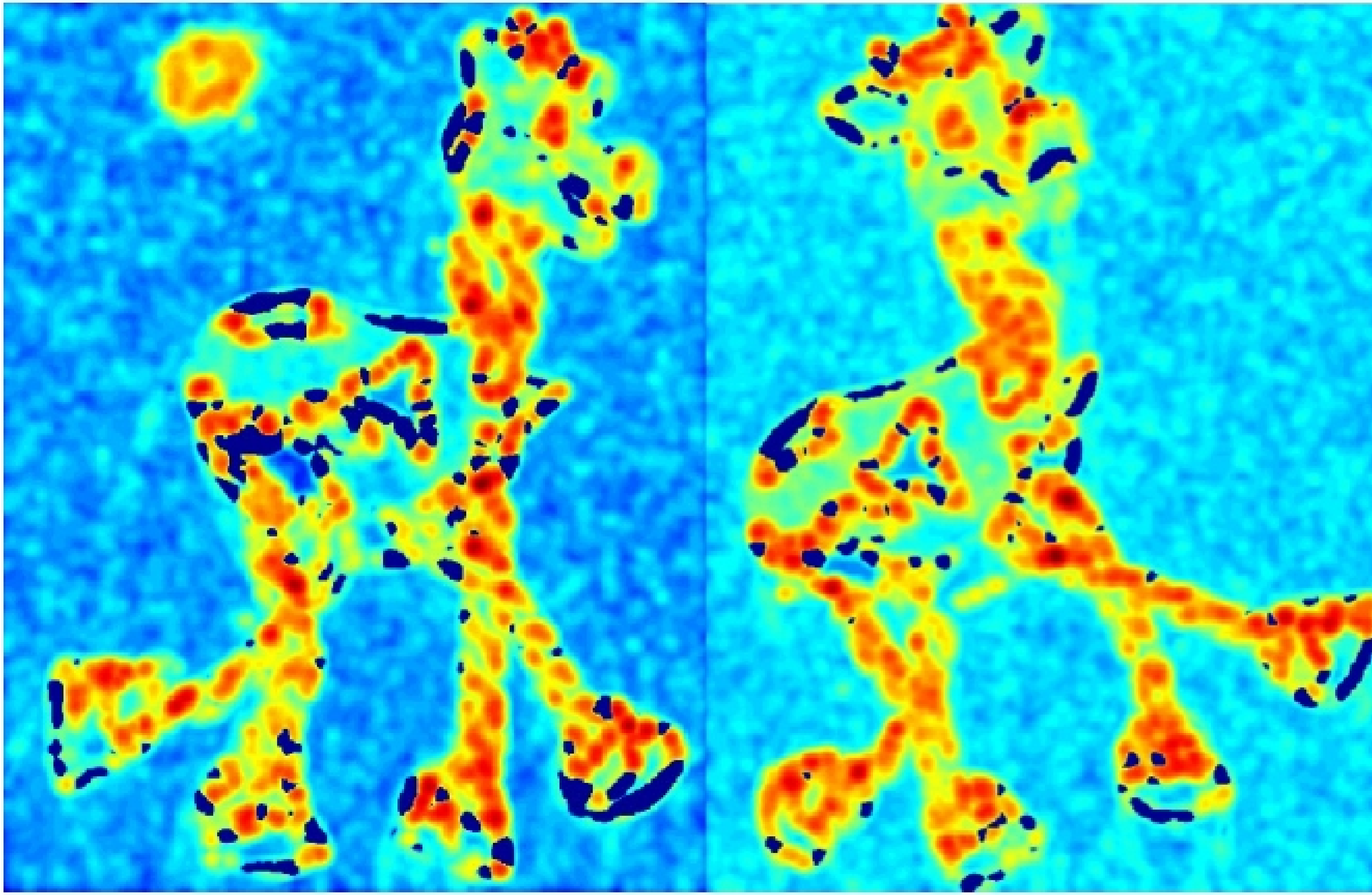


or

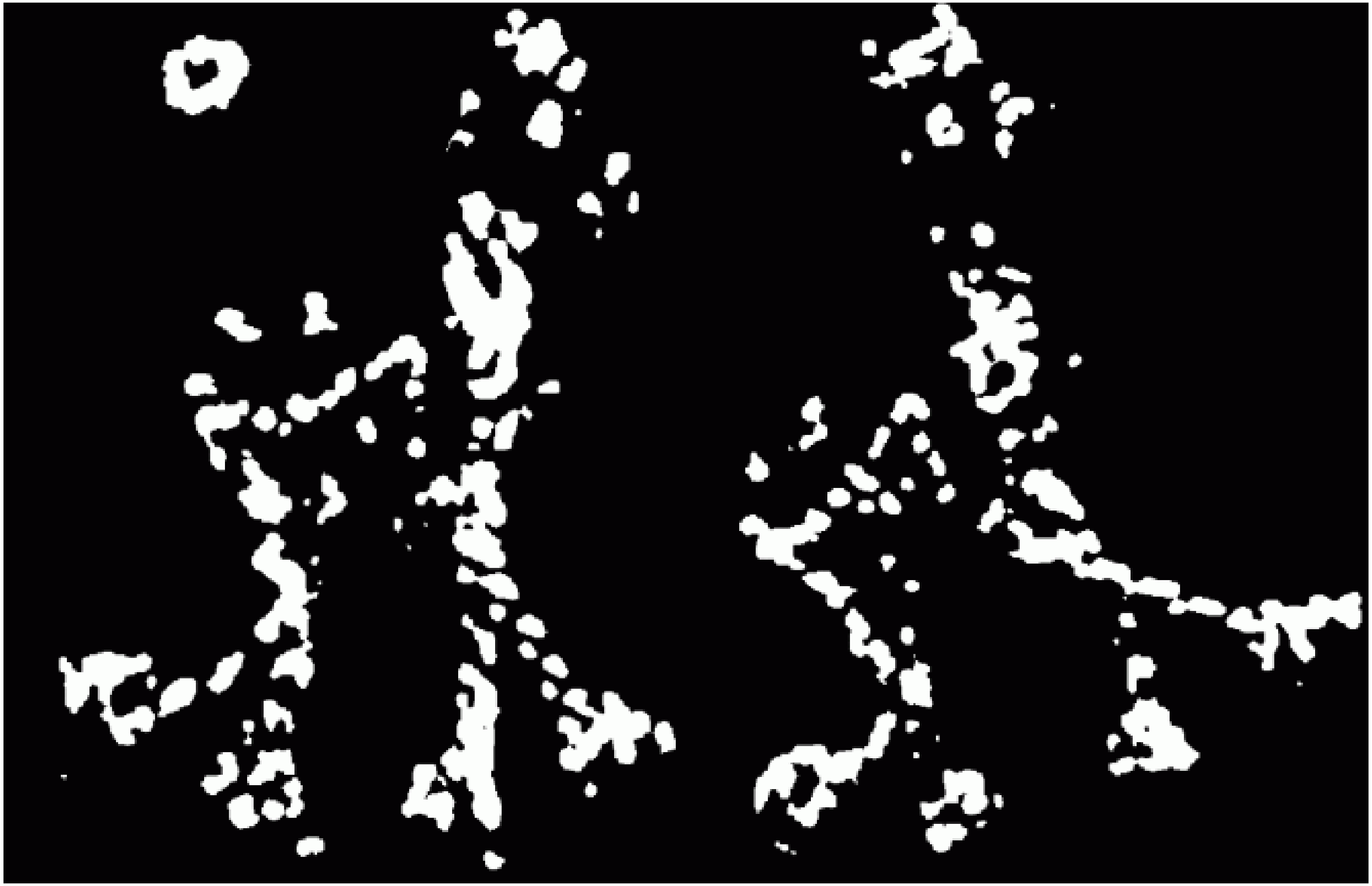


1 in window, 0 outside

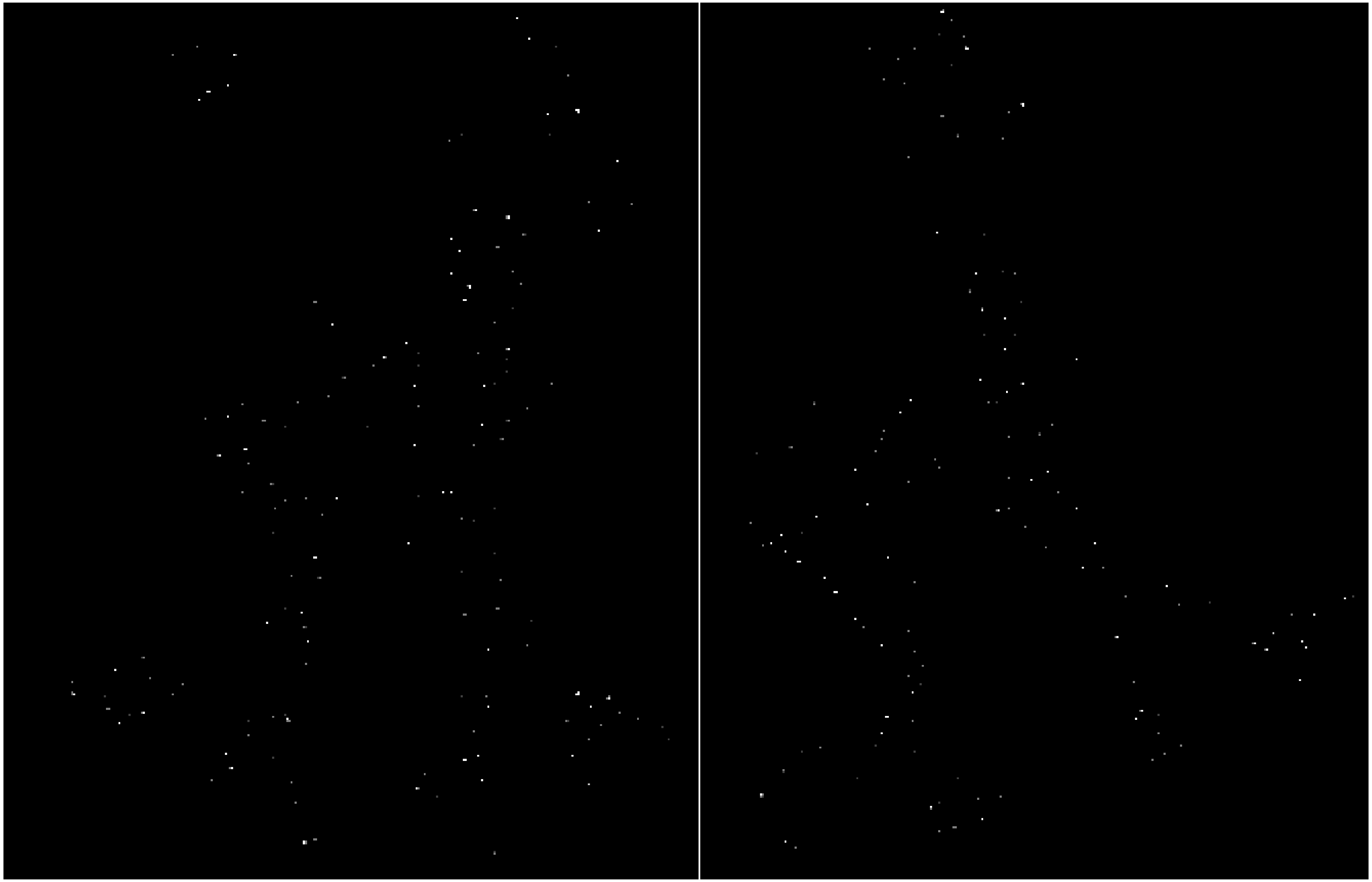
Gaussian

Eliminate small responses.



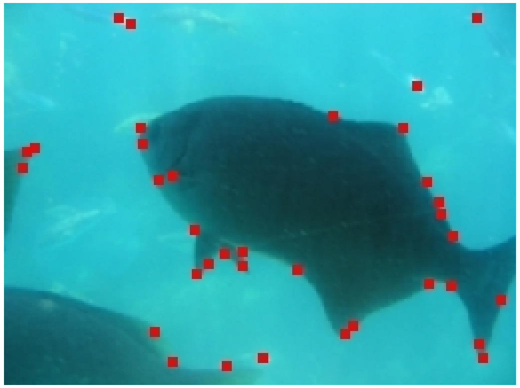
Find local maxima of the remaining.



OpenCV: finding features

cv::cornerHarris(...)

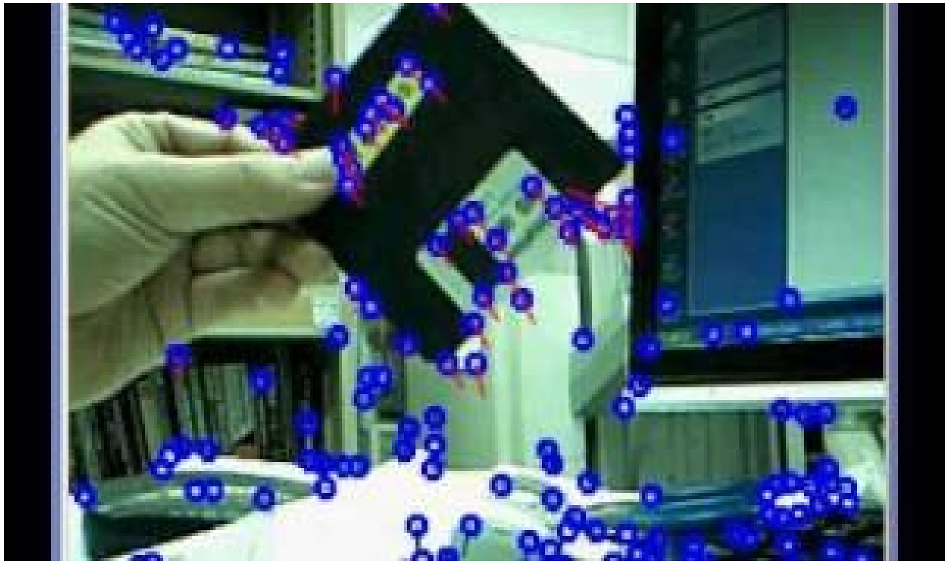
http://docs.opencv.org/2.4/doc/tutorials/features2d/trackingmotion/harris_detector/harris_detector.html



OpenCV: finding features

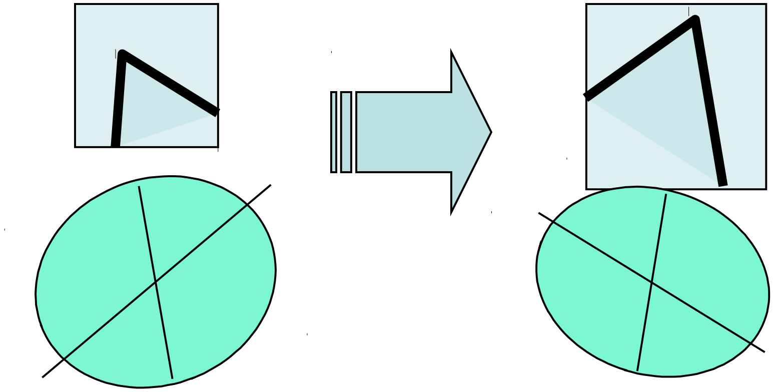
Shi and Tomasi '94: cv::goodFeaturesToTrack(...)

http://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html



Harris Detector: Some Properties

- Rotation invariance

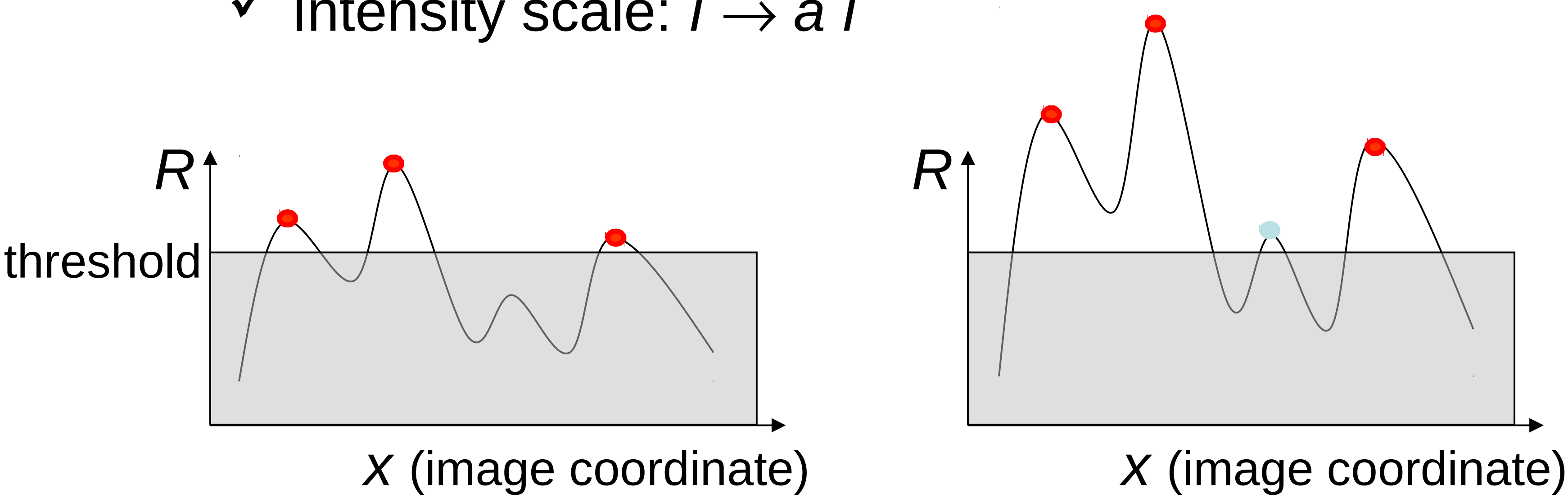


Ellipse rotates but its shape remains the same

Corner response R is invariant to image rotation

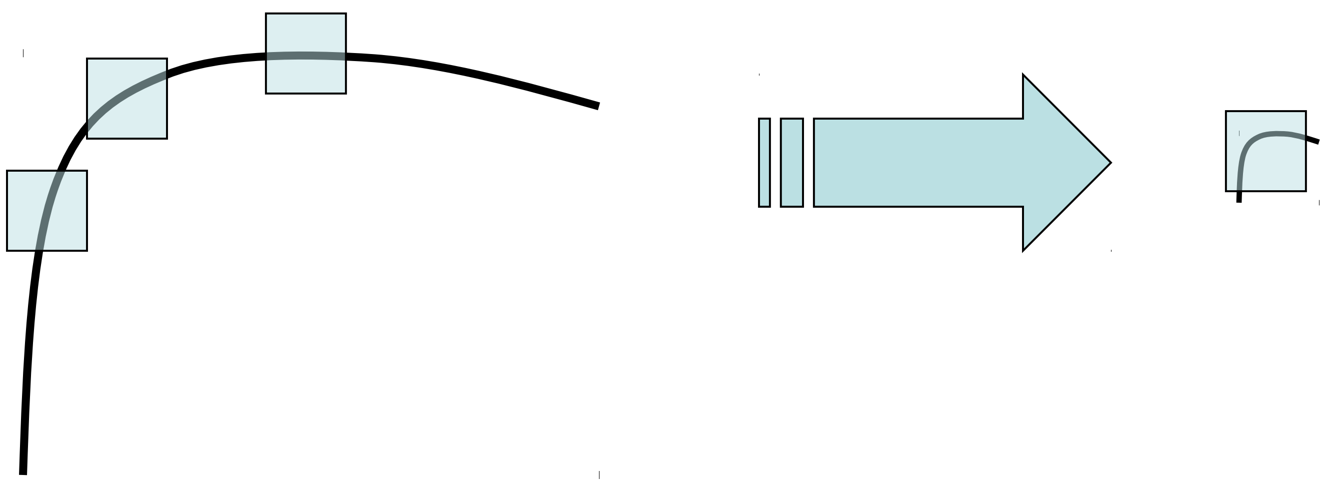
Harris Detector: Some Properties

- Partial invariance to *affine intensity* change
  - ✓ Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$
  - ✓ Intensity scale: $I \rightarrow a\, I$



Harris Detector: Some Properties

- But: non-invariant to *image scale*!

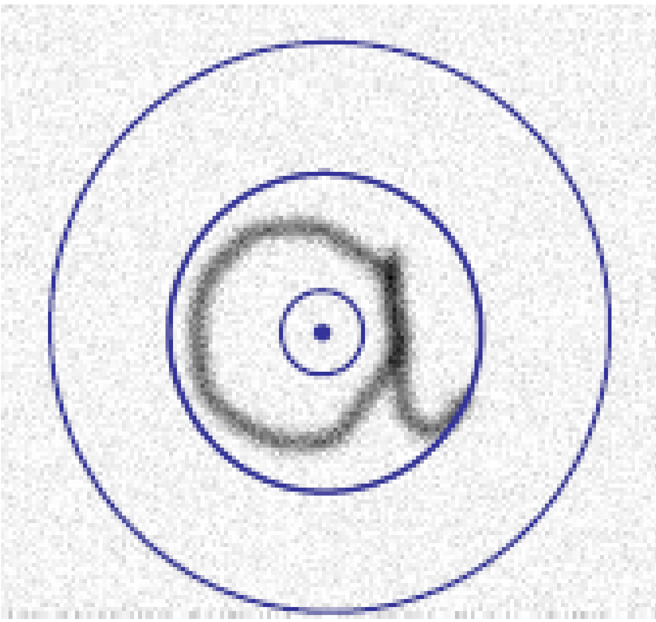


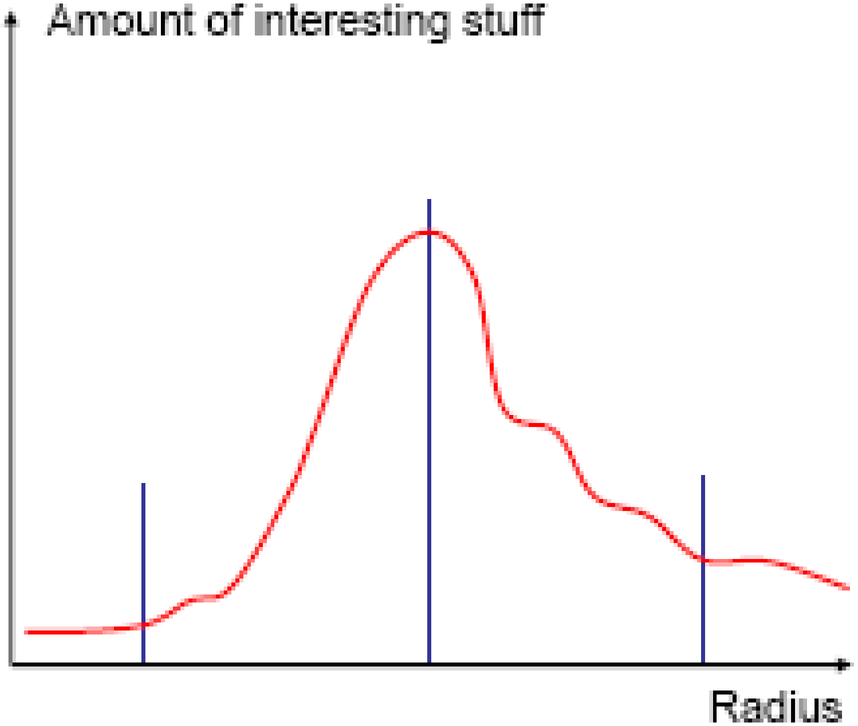
All points will be classified as edges

Corner !

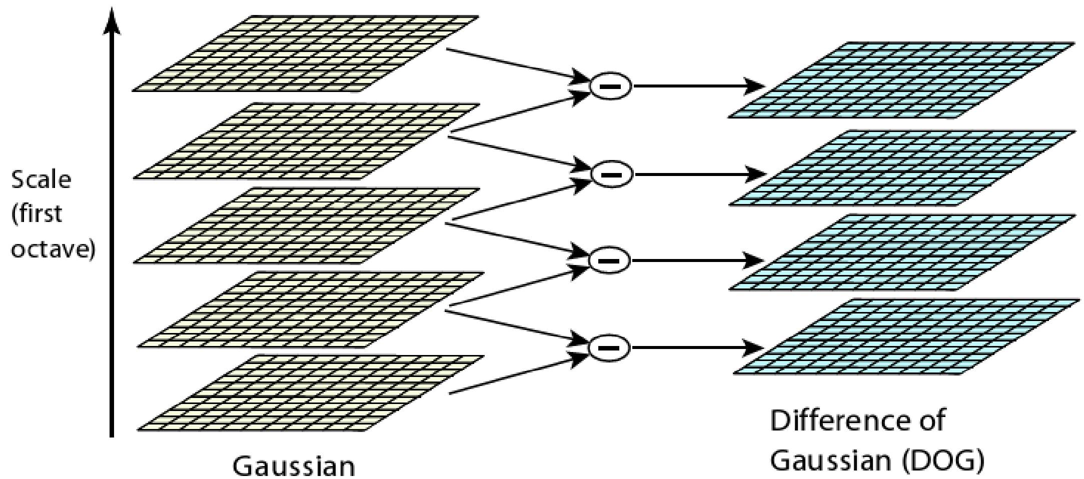
Achieving Scale Invariance

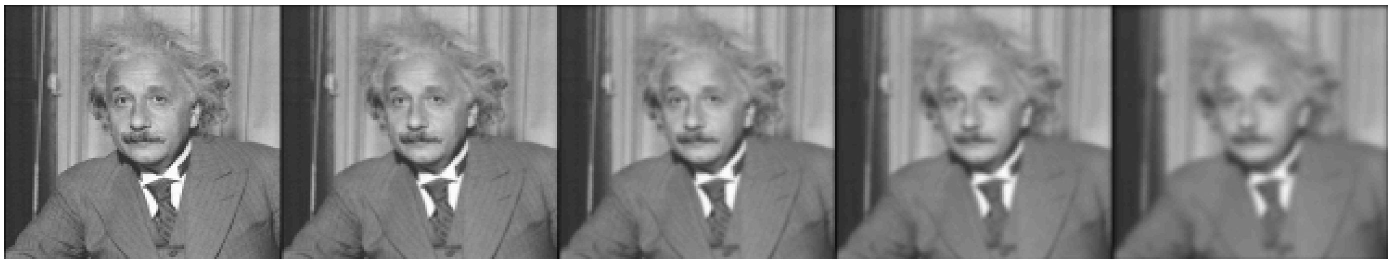
- How do we choose scale?



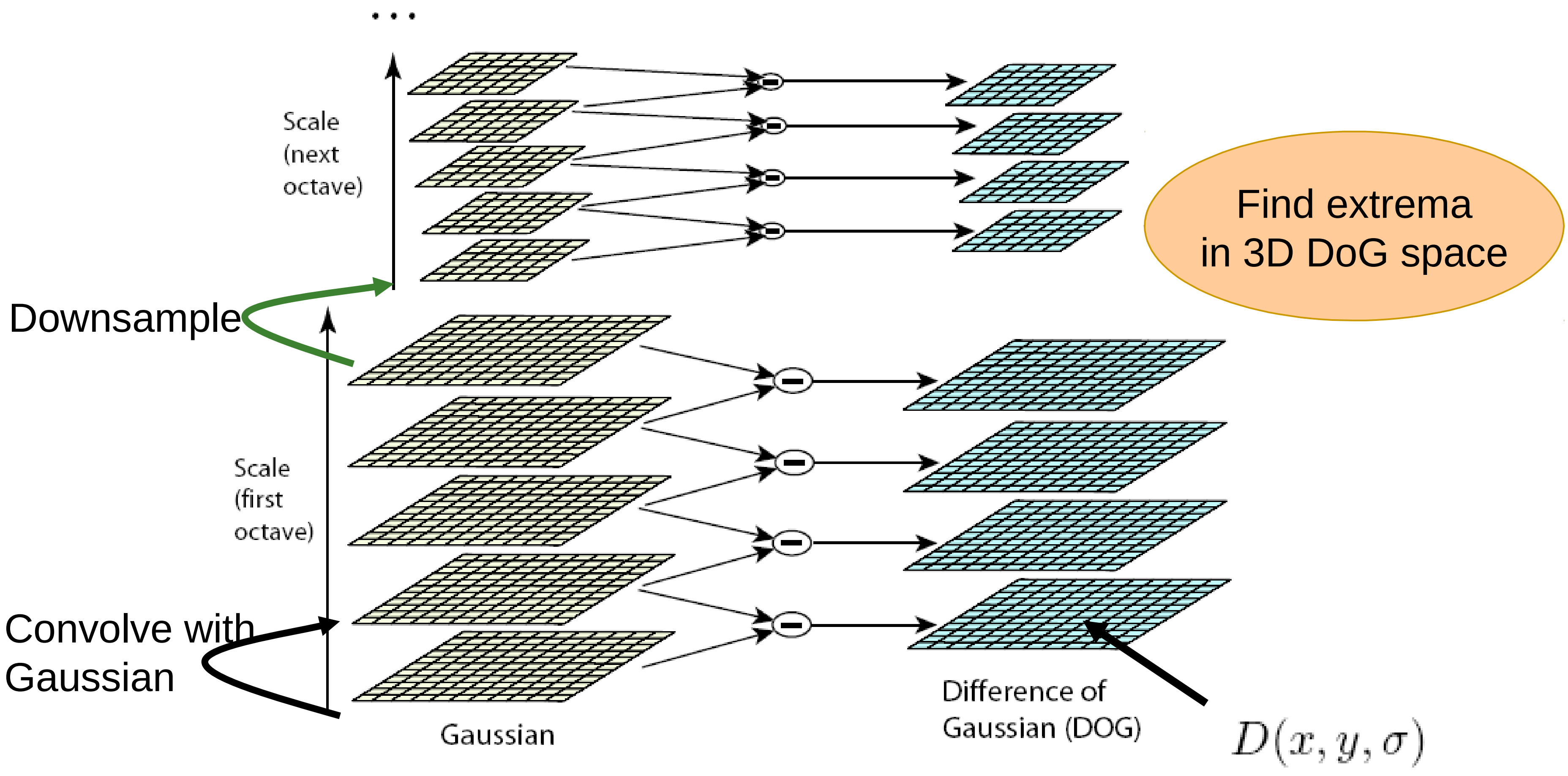


Difference-of-Gaussians

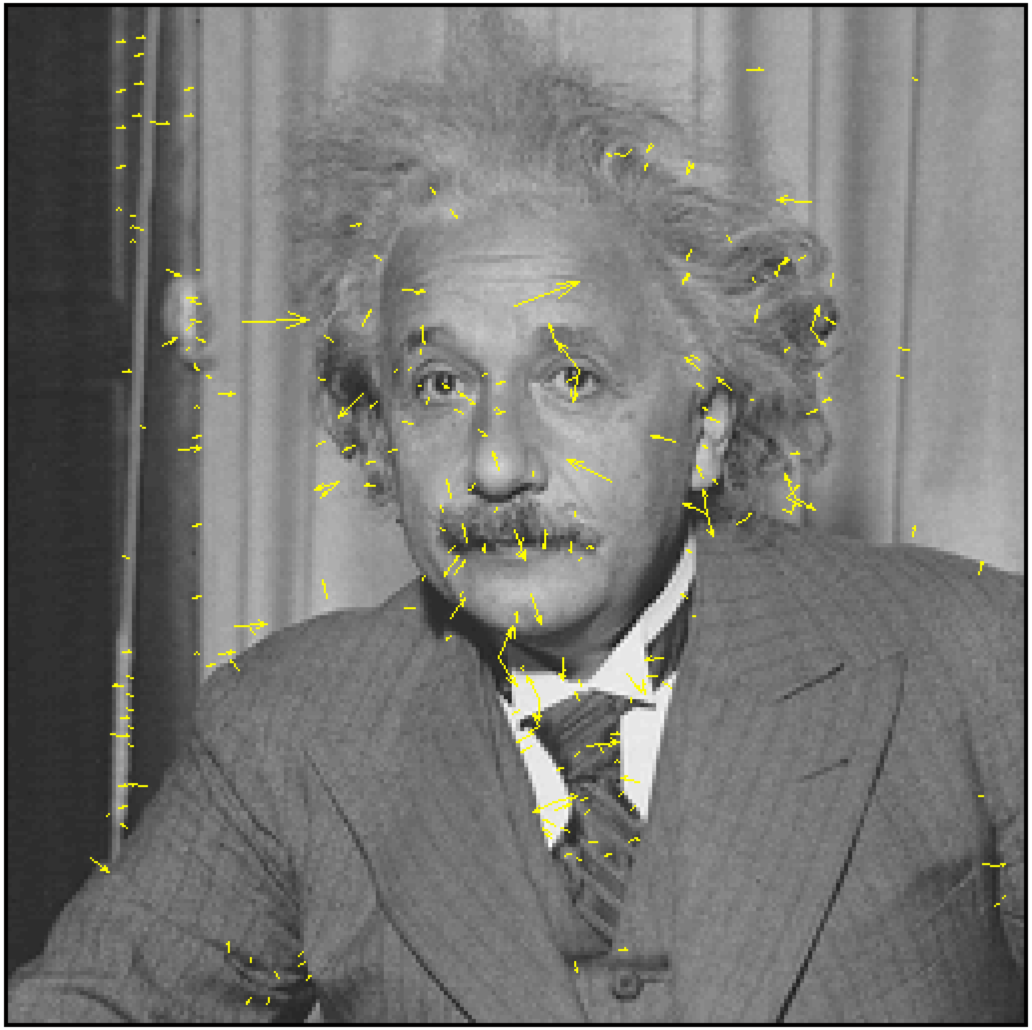




Finding Keypoints – Scale, Location

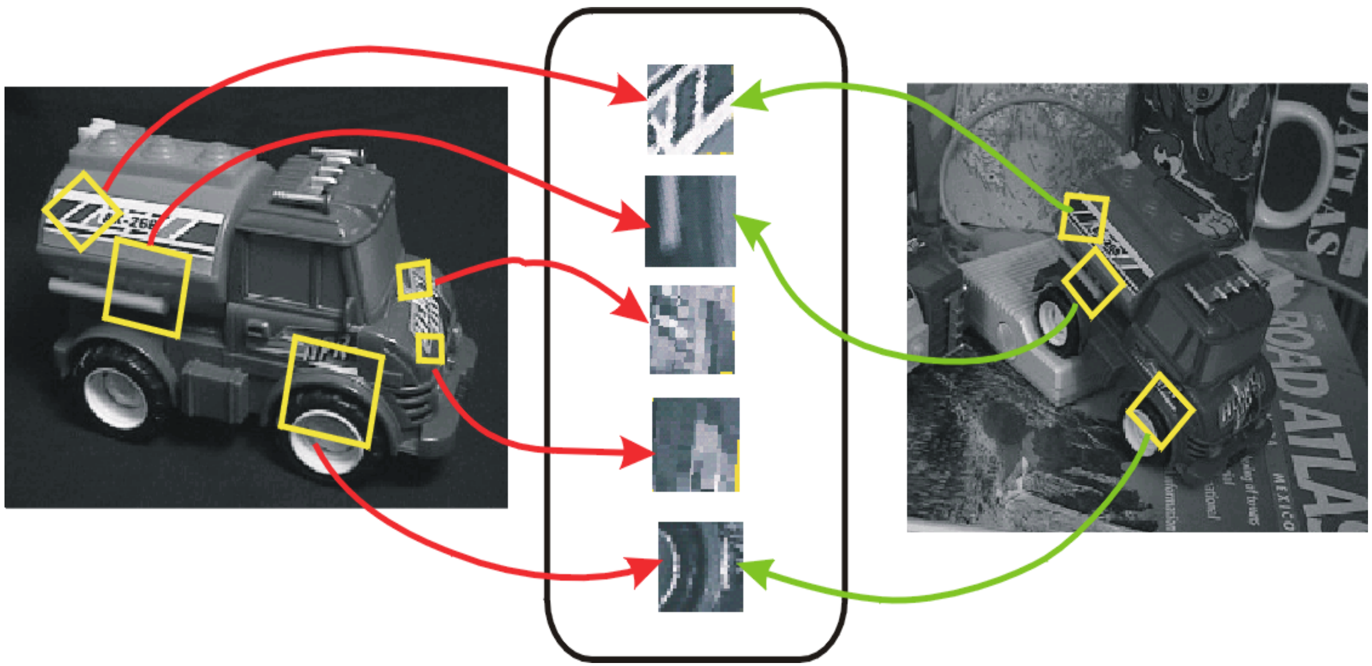


SIFT descriptor



Interest Point Descriptors

- Now that we can find interest points, how do we compare them?



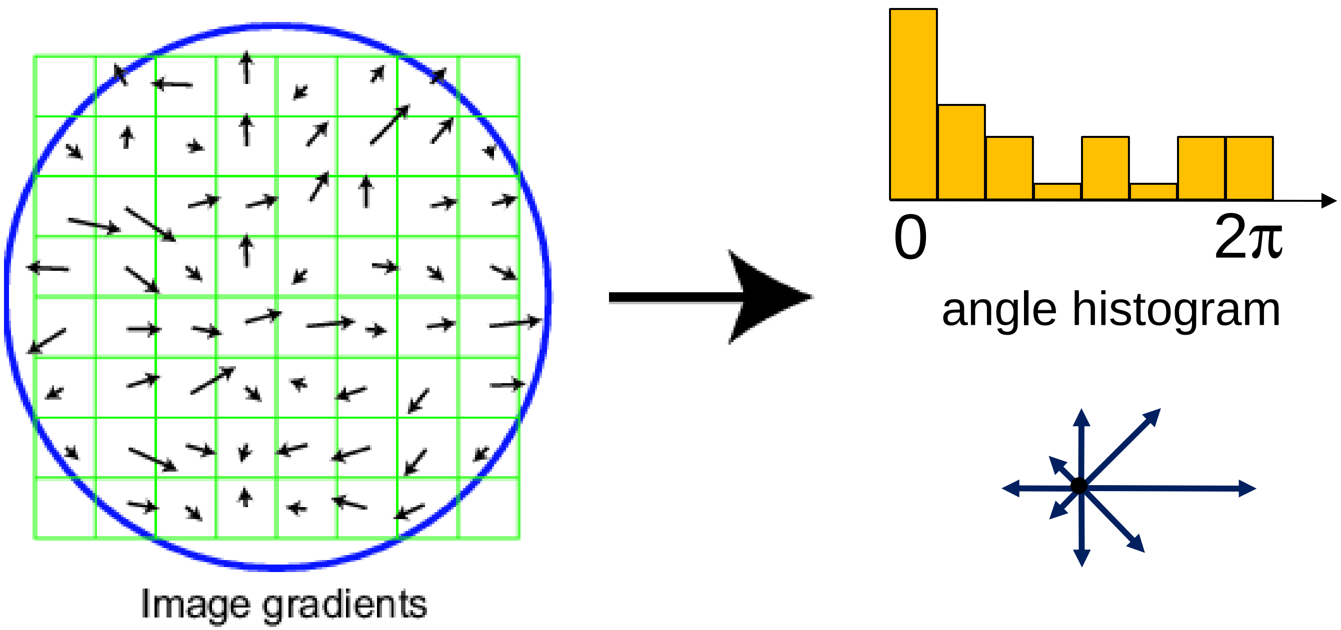
Interest Point Descriptors

- Now that we can find interest points, how do we compare them?
- Answer: compute a numerical feature descriptor describing the orientation, and scale of the interest point

SIFT descriptor

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Create histogram of edge orientations

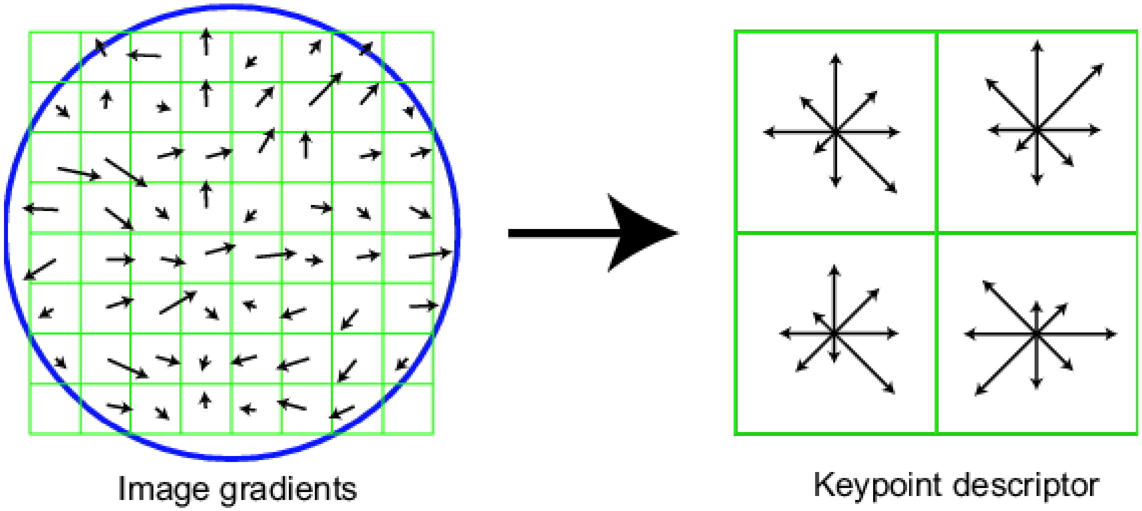


Adapted from slide by David Lowe

SIFT descriptor

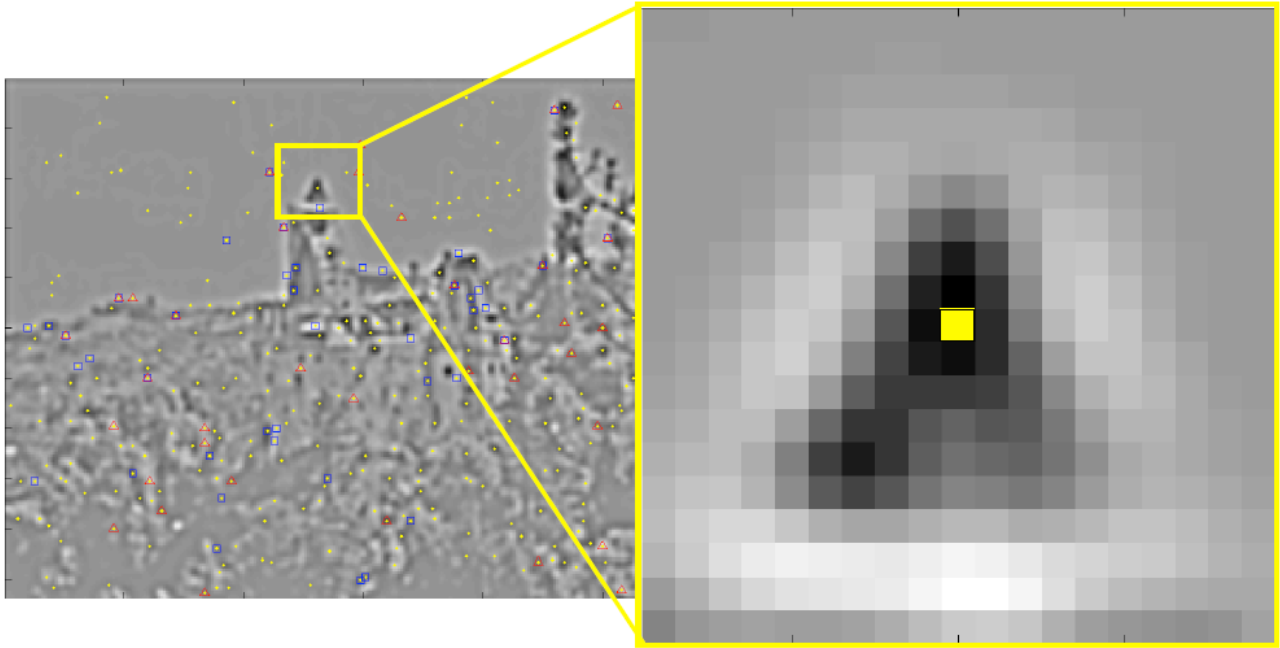
Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor



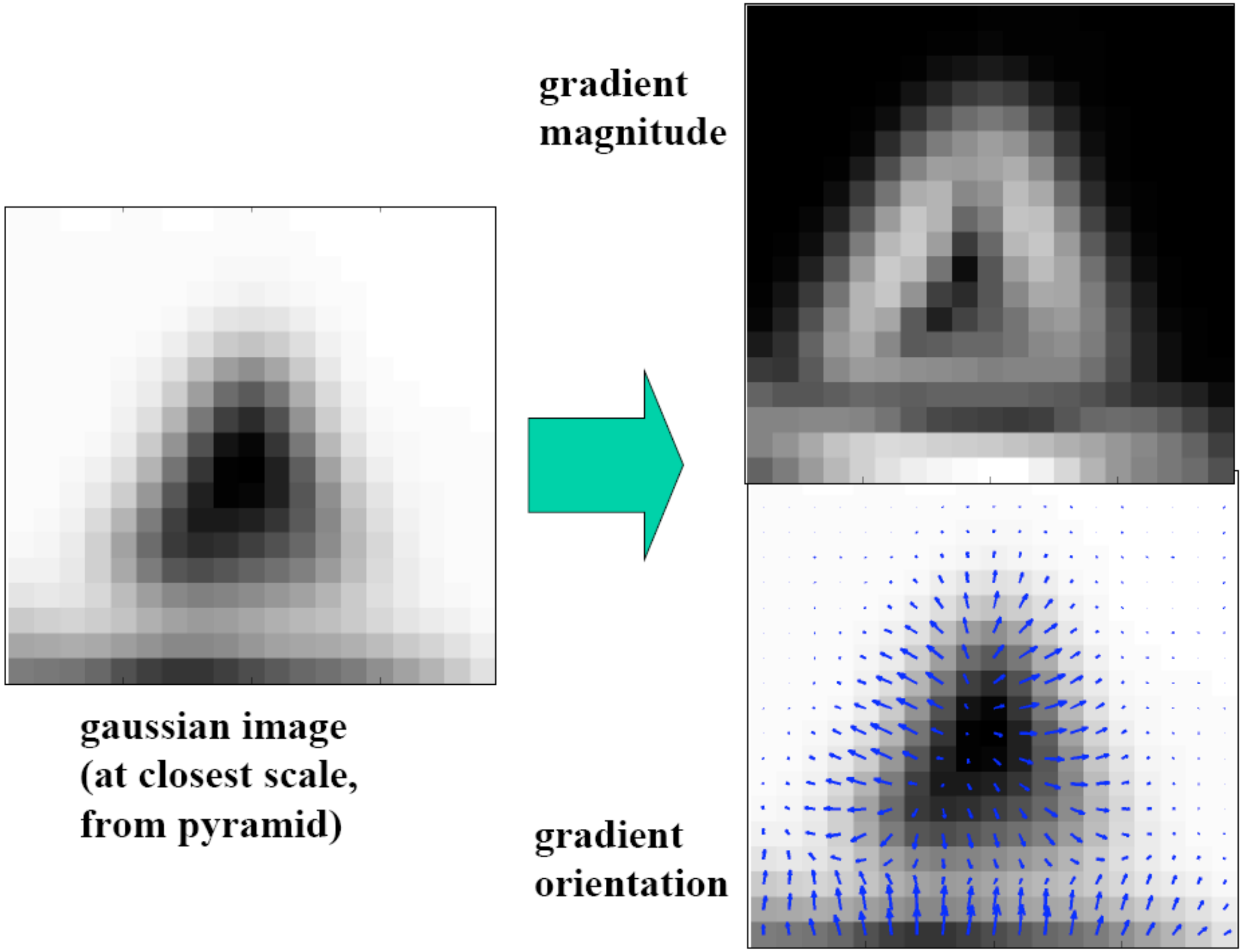
Adapted from slide by David Lowe

SIFT descriptor

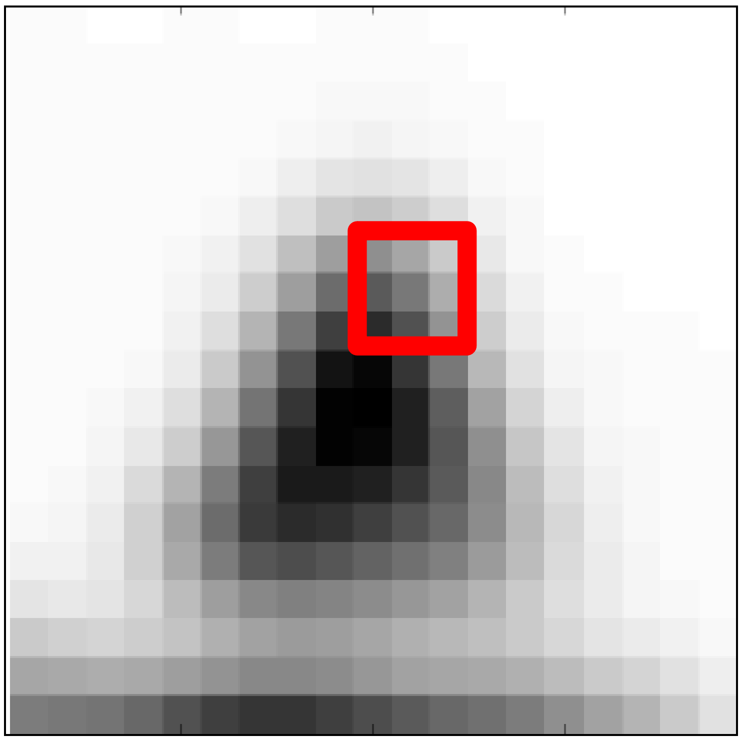


•Keypoint location = extrema location
•Keypoint scale is scale of the DOG image

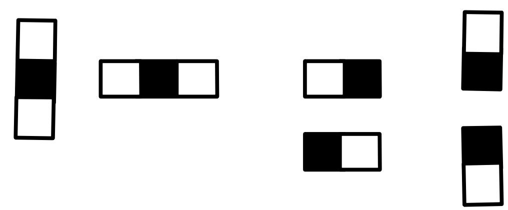
SIFT descriptor

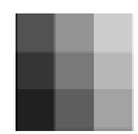


Computing Angle of Gradient



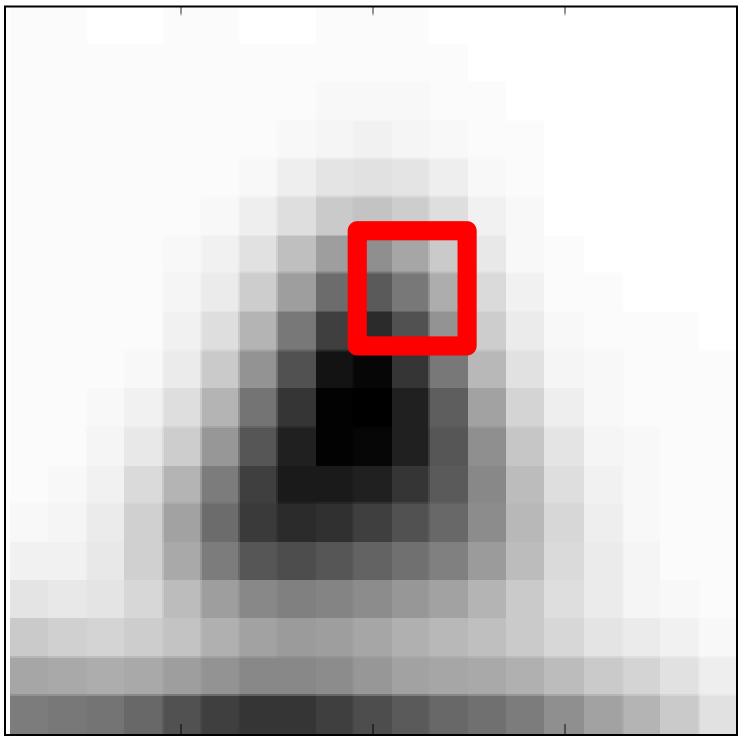
Angle and magnitude of gradient are computed using 1 and 2-side edge filters:



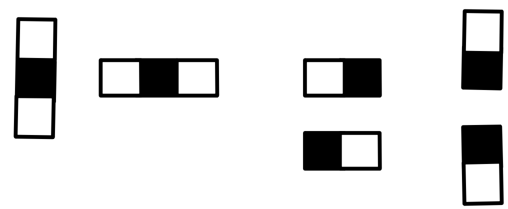


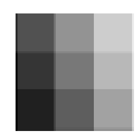
Patch

Computing Angle of Gradient



Angle and magnitude of gradient are computed using 1 and 2-side edge filters:

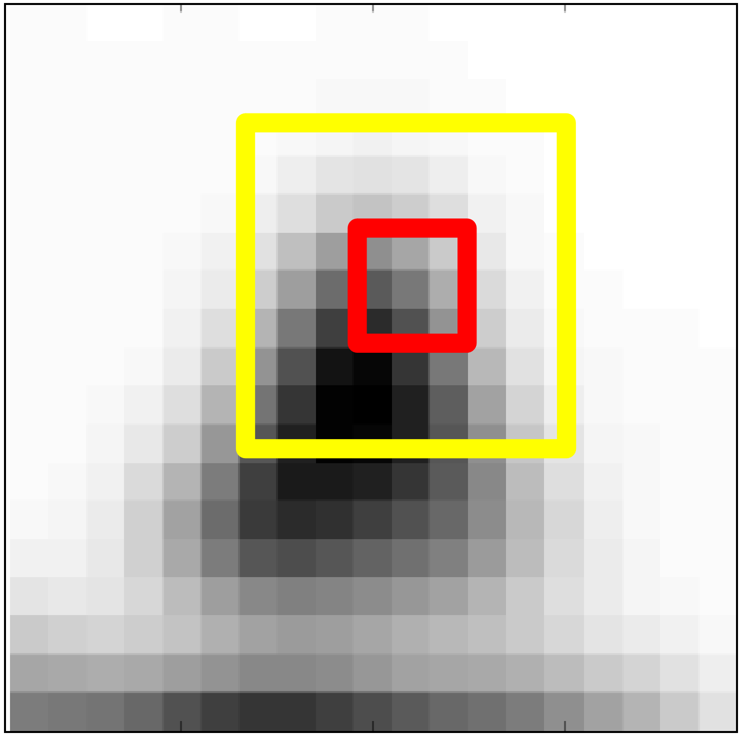




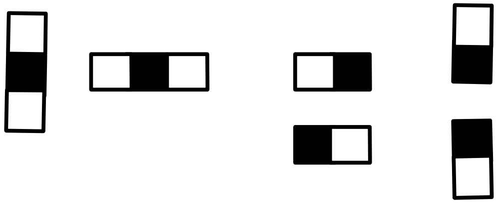
Patch

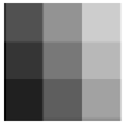
Angles have to be relative to reference frame. How do we choose it?

Computing Angle of Gradient



Angle and magnitude of gradient are computed using 1 and 2-side edge filters:



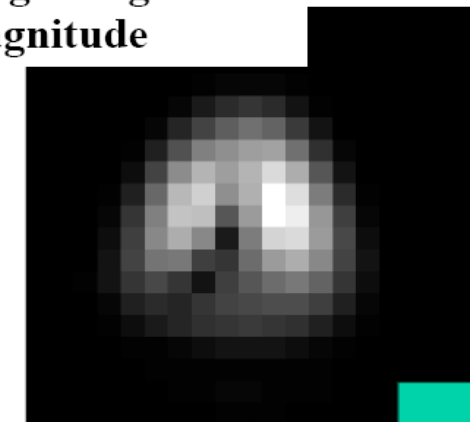


Patch

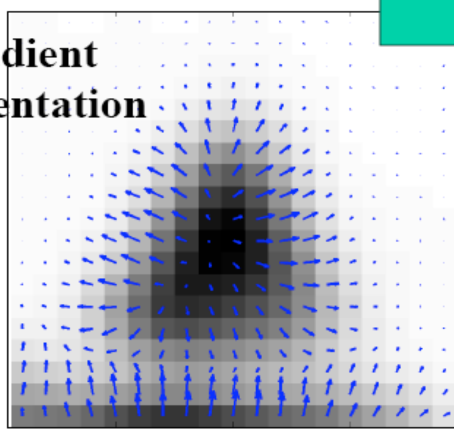
Angles have to be relative to reference frame. How do we choose it?

SIFT descriptor

weighted gradient
magnitude

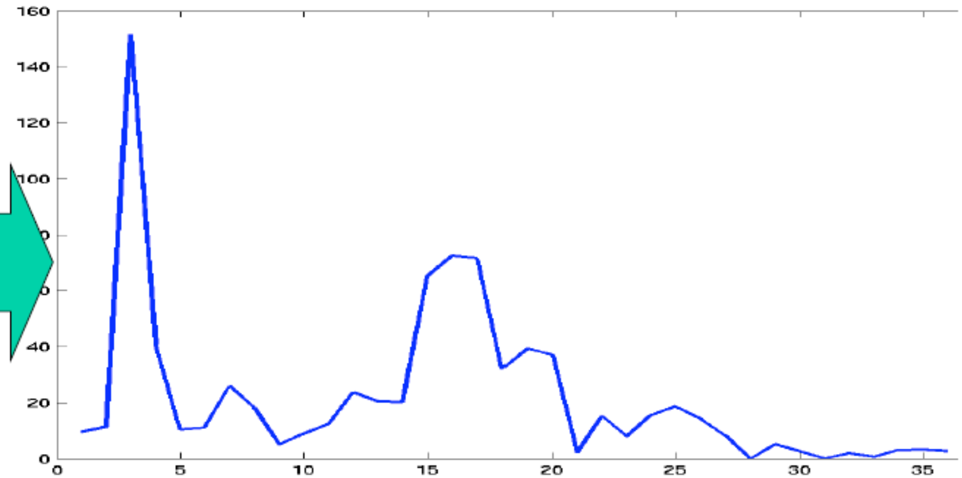


gradient
orientation



weighted orientation histogram.

Each bucket contains sum of weighted gradient magnitudes corresponding to angles that fall within that bucket.



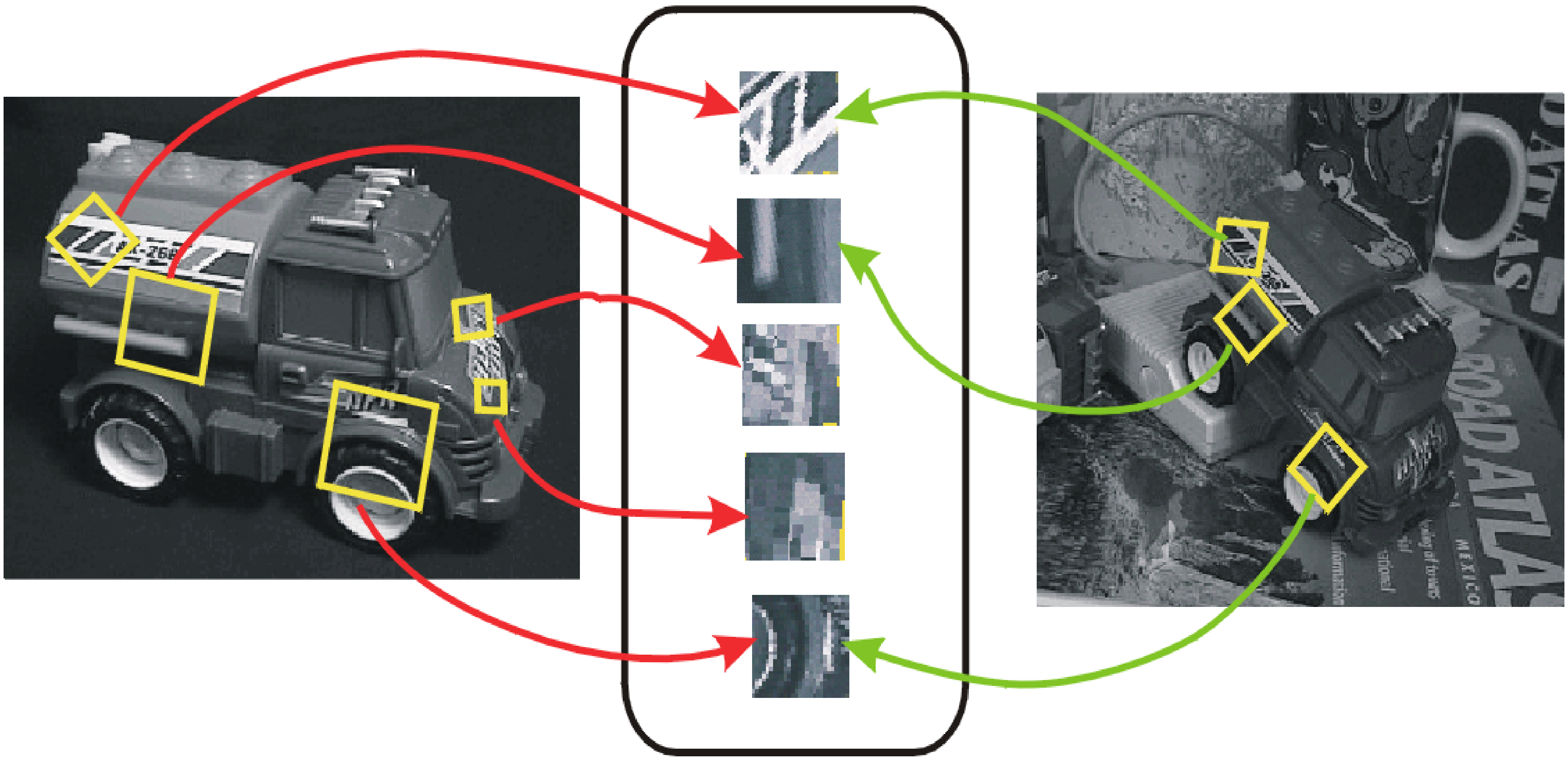
36 buckets

10 degree range of angles in each bucket, i.e.

0 <=ang<10 : bucket 1

10<=ang<20 : bucket 2

20<=ang<30 : bucket 3 …

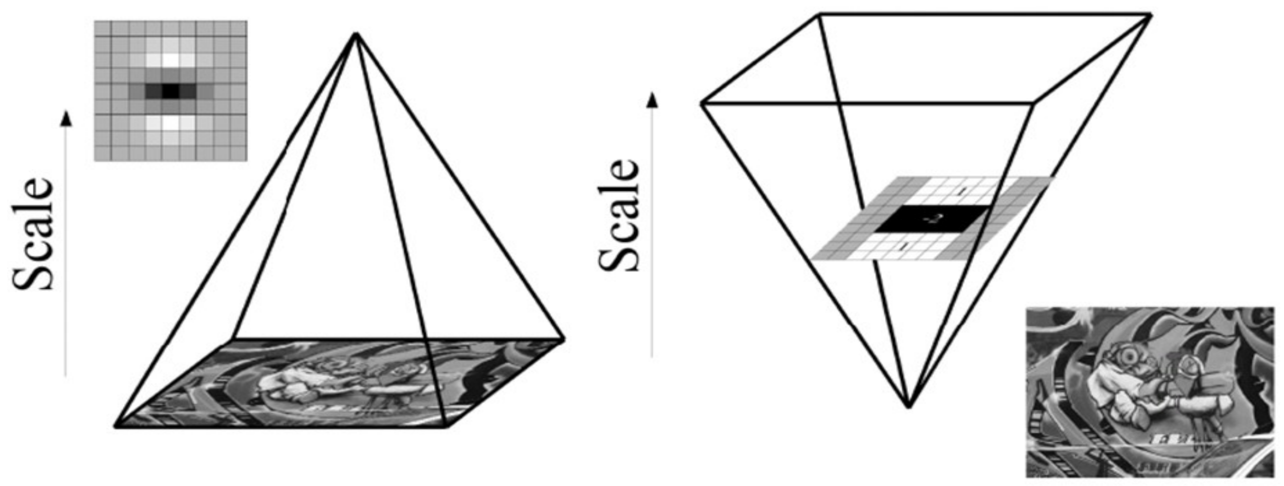

ROAD ATLAS

The problem with SIFT...

- Slow...
- Copyrighted!
  - Alternatives: SURF
  - OpenSURF:
    - http://opensurf1.googlecode.com/files/OpenSURF.pdf
  - Included in OpenCV 2.0+
  - OpenCV Tutorial:
    - http://achuwilson.wordpress.com/2011/08/05/object-detection-using-surf-in-opencv-part-1/

SURF

- Achieves quicker computation by scaling the filter rather than the image:



SIFT **SURF**

To summarize...

- Feature detectors:
  - Find interest points in image (e.g., using difference of Gaussians, Harris corner detection, etc.)
- Feature descriptors
  - Each detected feature can be represented by a numerical descriptor encoding orientation, scale, etc.

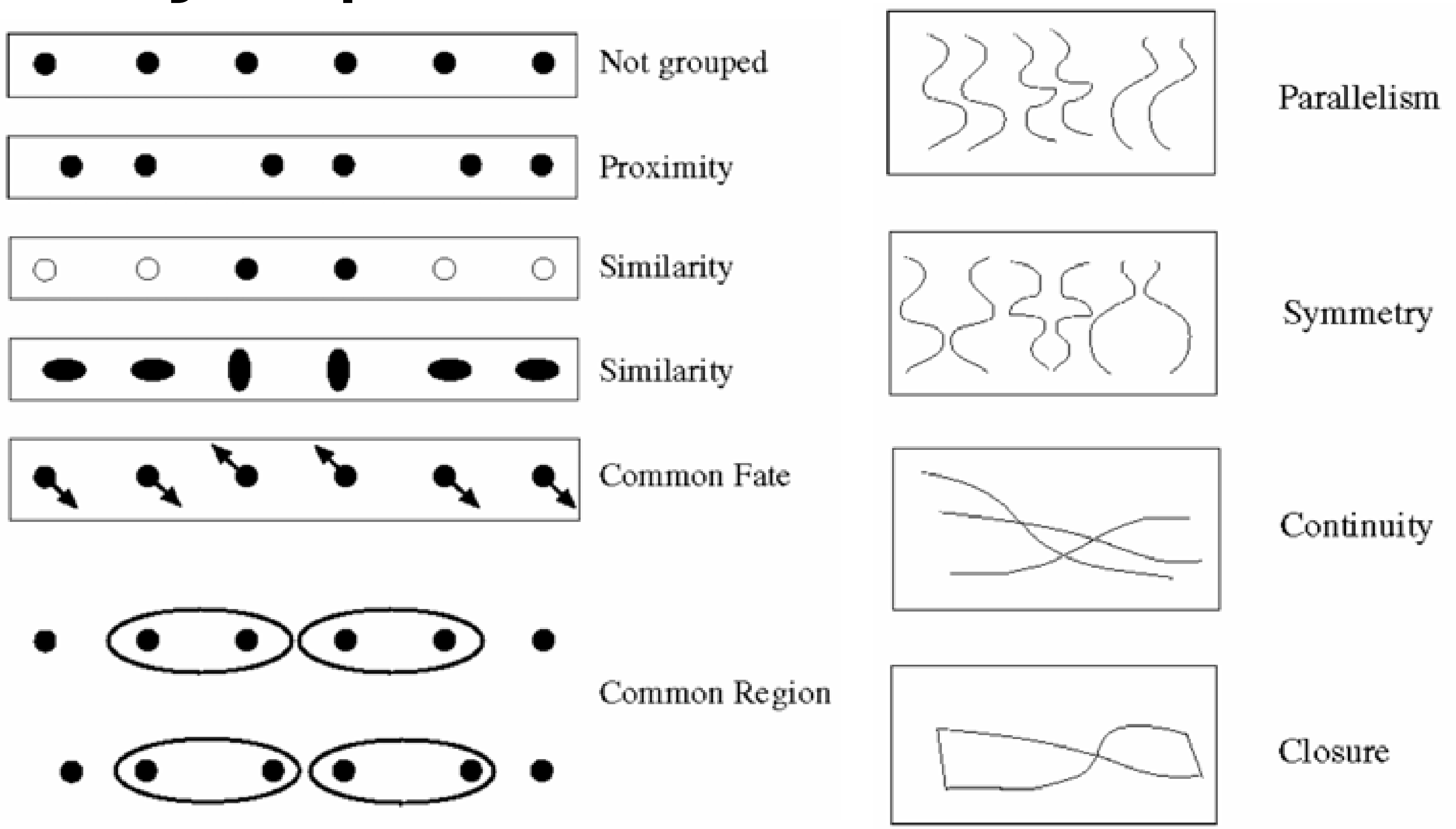
Applications

- Object Recognition using SURF:
  - https://www.youtube.com/watch?v=1xNSxVcwjjY
- Markerless AR using SURF:
  - https://www.youtube.com/watch?v=caFHvamMUTw
- Robot Localization and Mapping:
  - www.youtube.com/watch?v=cCp-xwkRD8E

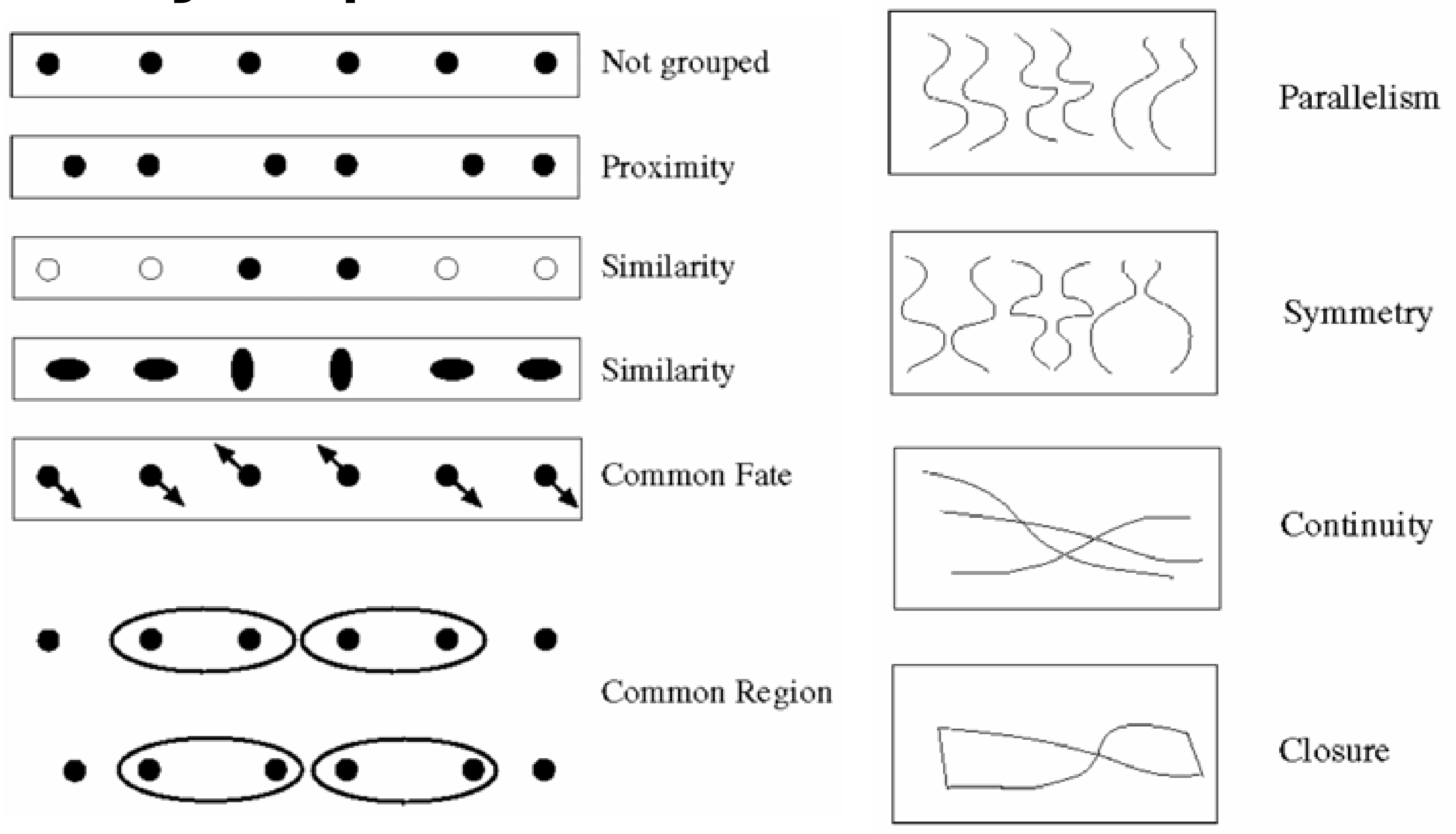
Optical Flow

- Interest key points and feature descriptors are great but suffer from one limitation:
  - They ignore time

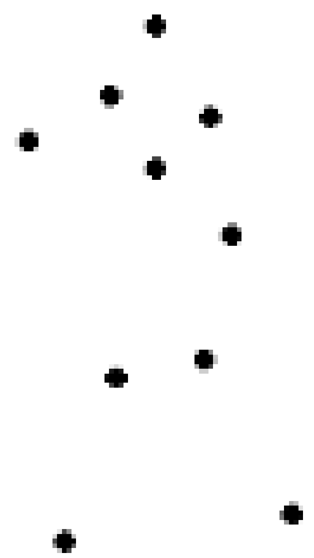
Why Optical Flow?



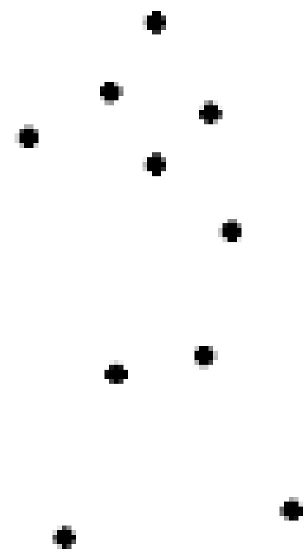
Why Optical Flow?



Why Optical Flow?



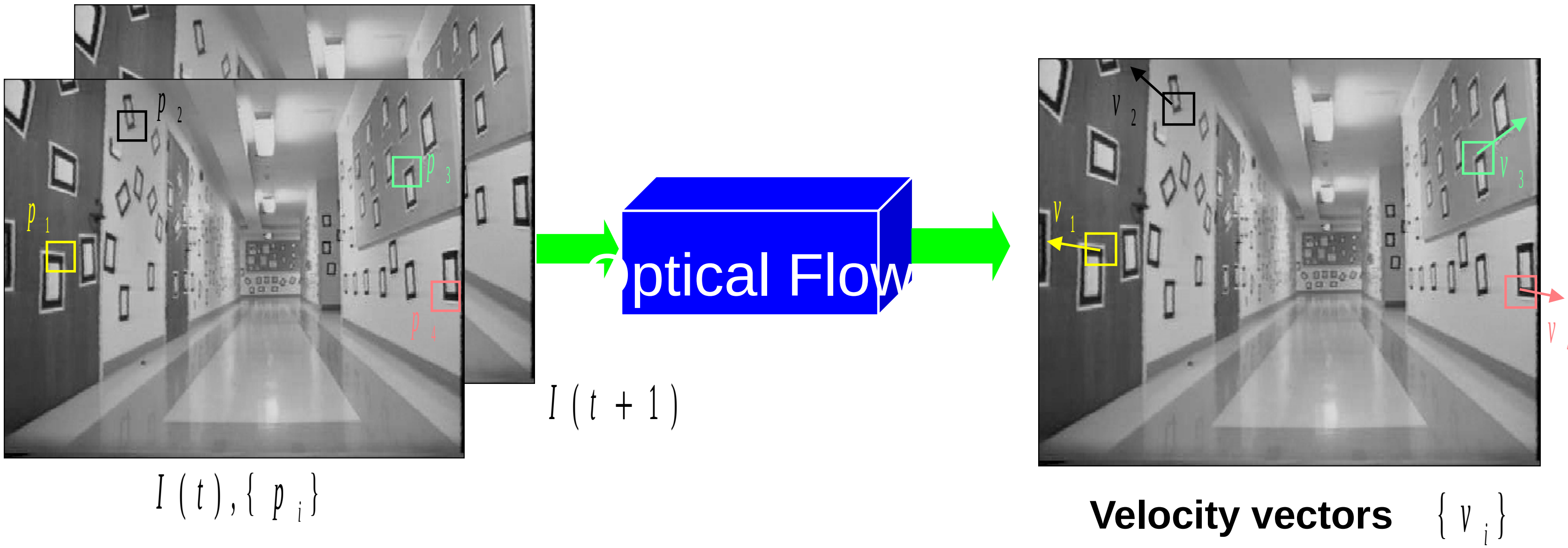
Why Optical Flow?



Optical Flow Video

- http://www.youtube.com/watch?v=o8NOabnZPIY

What is Optical Flow?

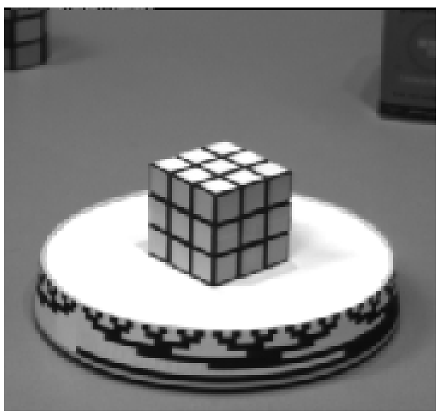


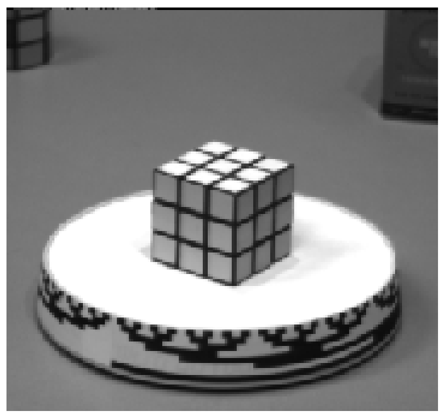
"Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image"

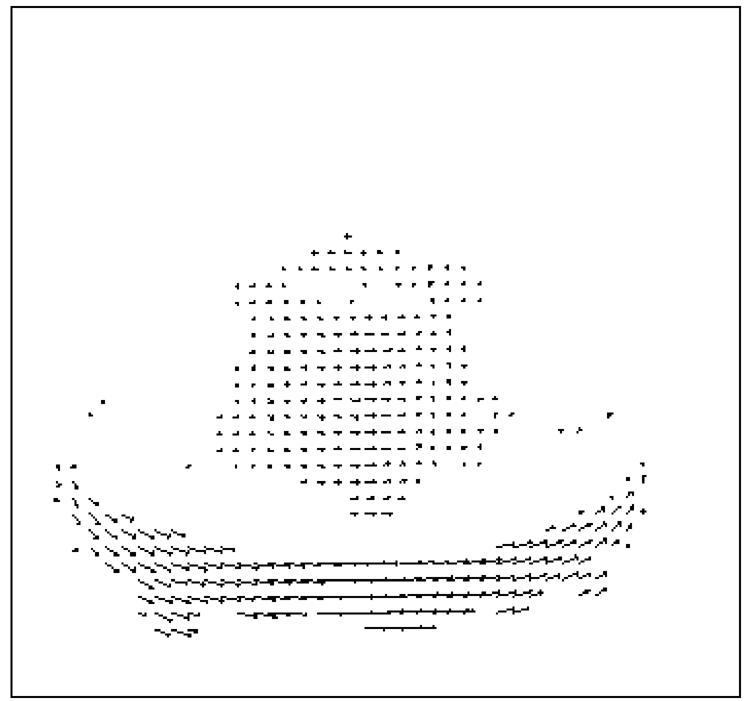
- Horn and Schunk, 1981

Motion Fields

- The motion field is the projection of the 3D scene motion into the image







Motion Fields and Camera Movement

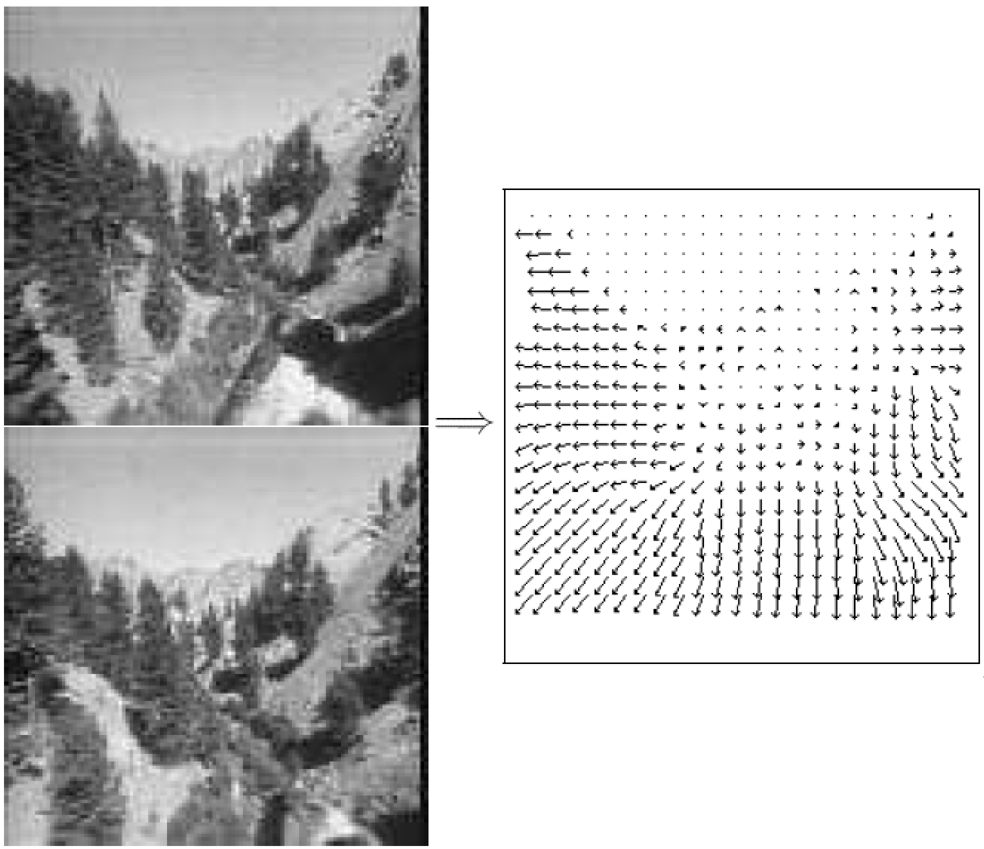


Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

Length of flow vectors inversely proportional to depth Z of 3D point

points closer to the camera move more quickly across the image plane

Figure from Michael Black, Ph.D. Thesis

Why do we want optical flow?

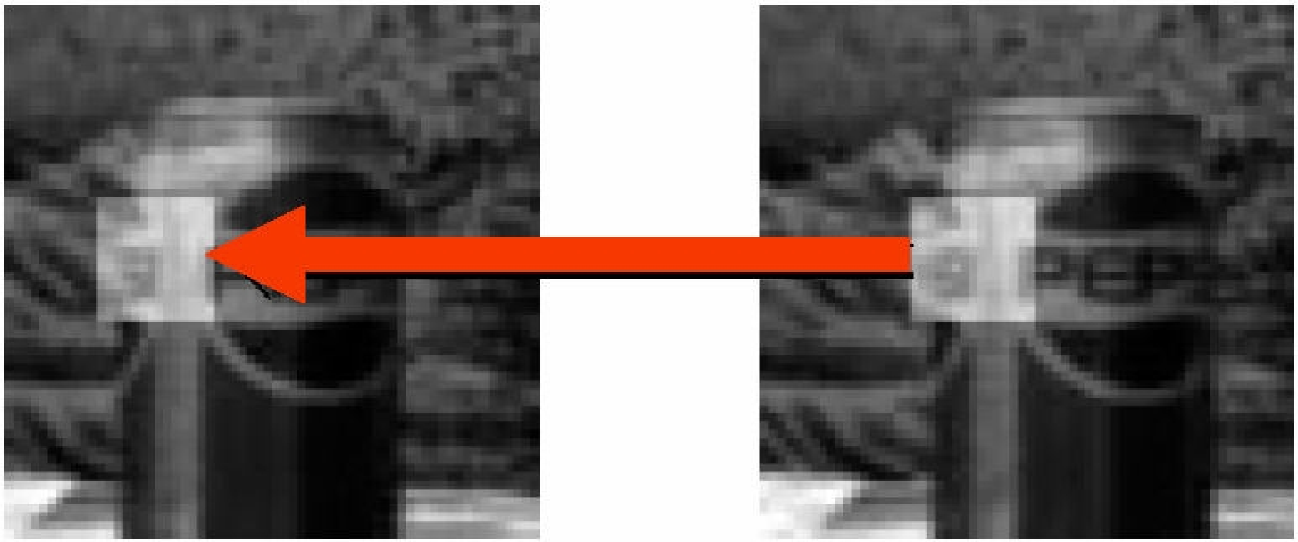
- For example, autonomous helicopters:
  - https://www.youtube.com/watch?v=V4r2HXGA8jw

Computing Optical Flow

- Given a set of points in an image, find those same points in another image
- Or, given point $[u_x, u_y]^T$ in image I_1 find the point $[u_x + \delta_x, u_y + \delta_y]^T$ in image I_2 that minimizes ε:

$$\varepsilon(\delta_x, \delta_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \left(I_1(x,y) - I_2(x+\delta_x, y+\delta_y) \right)$$

Optical Flow Assumptions



Assumption

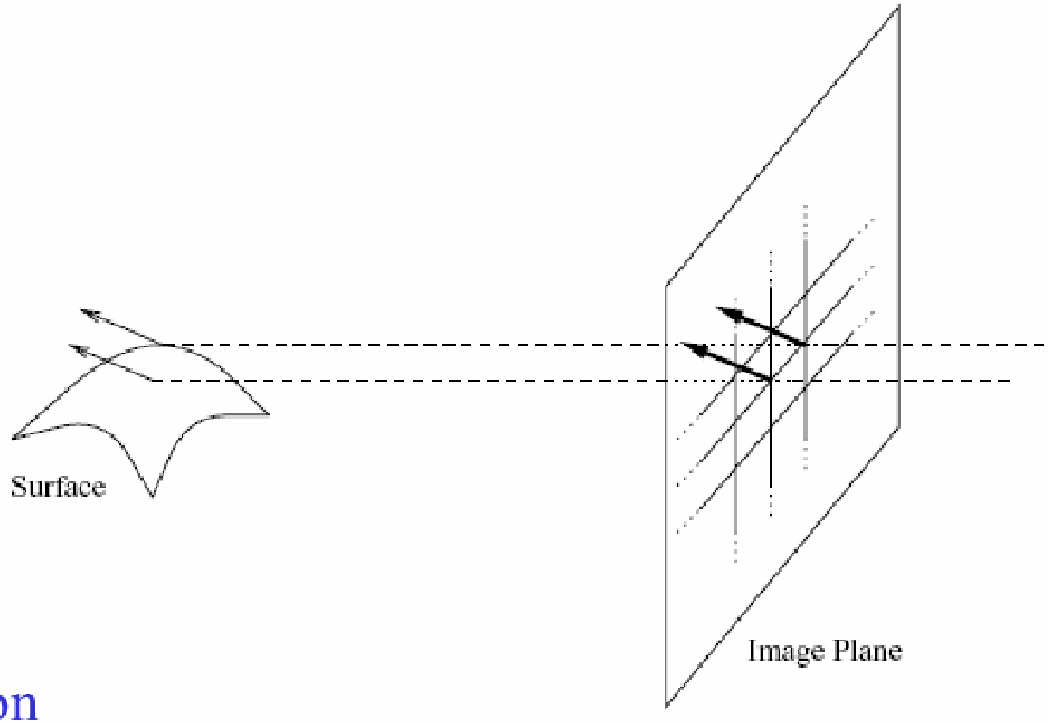
Image measurements (e.g. brightness) in a small region remain the same although their location may change.

$$I(x+u, y+v, t+1) = I(x, y, t)$$

(assumption)

* Slide from Michael Black, CS143 2003

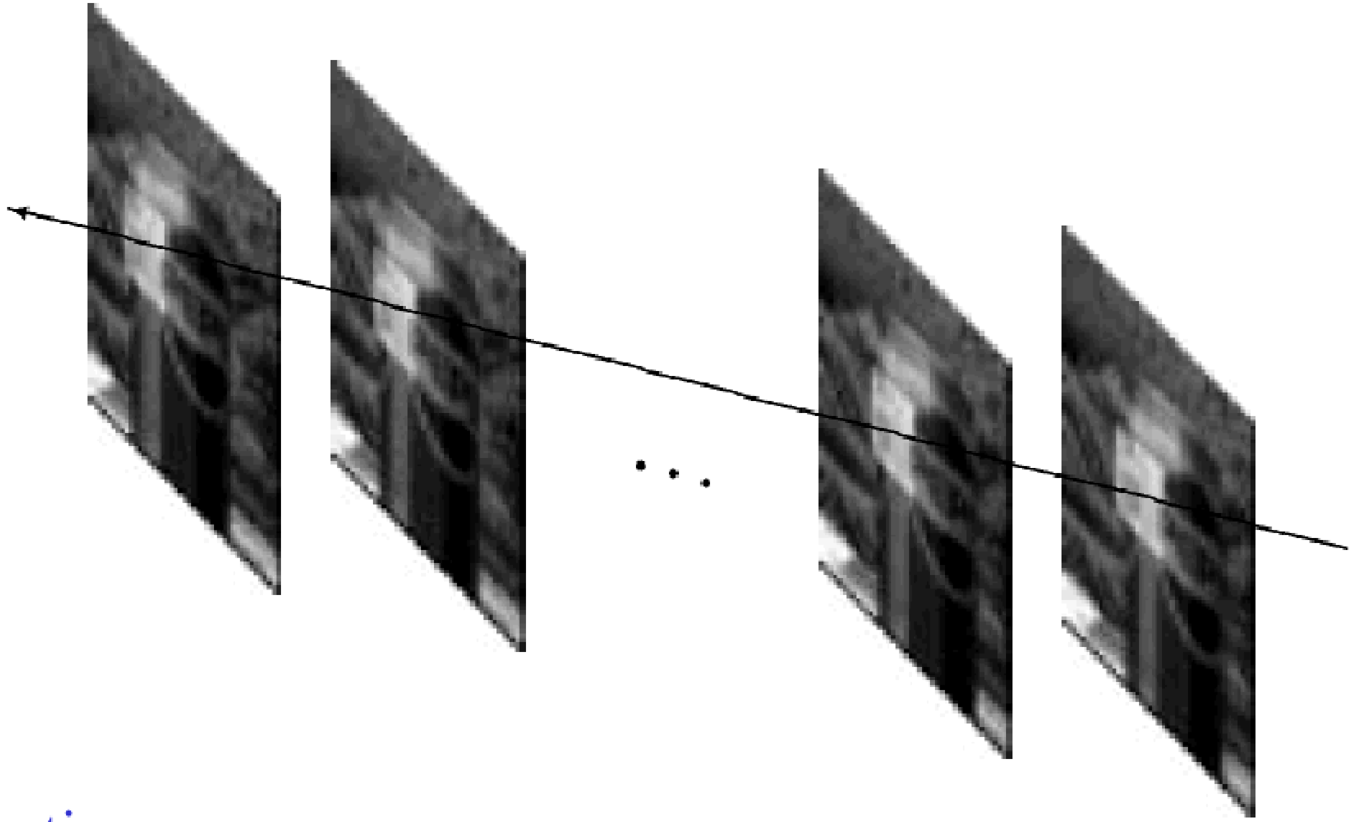
Spatial Coherence



Assumption

* Neighboring points in the scene typically belong to the same surface and hence typically have similar motions.
* Since they also project to nearby points in the image, we expect spatial coherence in image flow.

Temporal Persistence

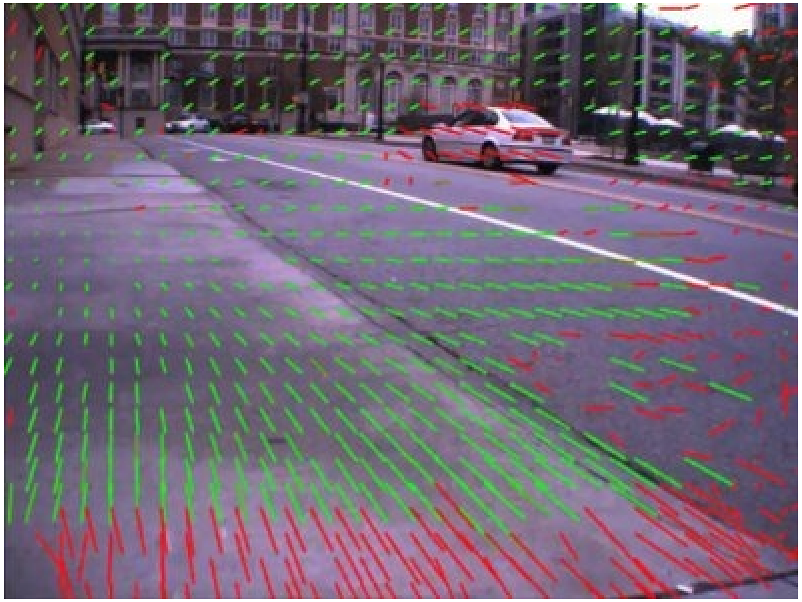


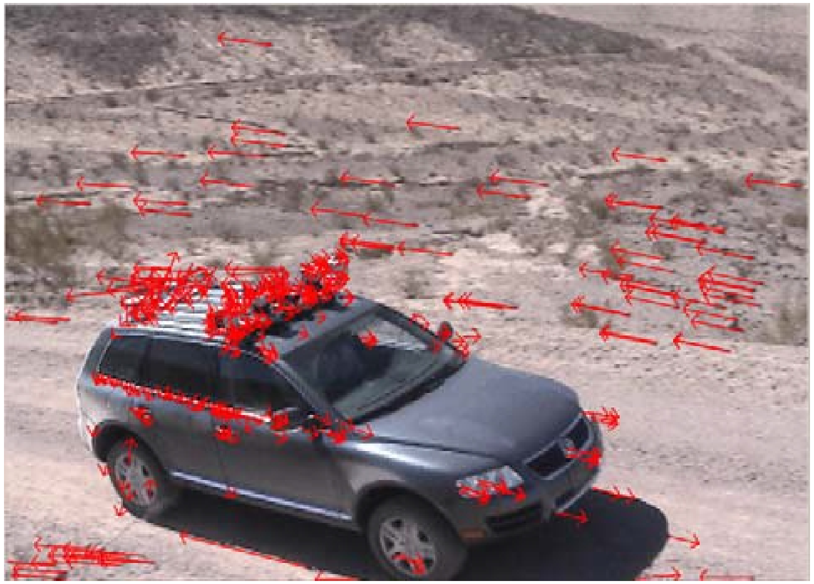
Assumption:

The image motion of a surface patch changes gradually over time.

* Slide from Michael Black, CS143 2003

Dense vs. Sparse Optical Flow





Applications

- Object Recognition using SURF:
  - https://www.youtube.com/watch?v=xzVXyrIRm30
- Markerless AR using SURF:
  - https://www.youtube.com/watch?v=caFHvamMUTw
- Robot Localization and Mapping:
  - www.youtube.com/watch?v=cCp-xwkRD8E

Code

- MATLAB:
  - Iterative Pyramidal LK Optical Flow
  - http://www.mathworks.com/matlabcentral/fileexchange/23142-iterative-pyramidal-lk-optical-flow
- OpenCV
  - http://robots.stanford.edu/cs223b05/notes/optical_flow_demo.cpp

THE END