The SIFT (Scale Invariant Feature Transform) Detector and Descriptor

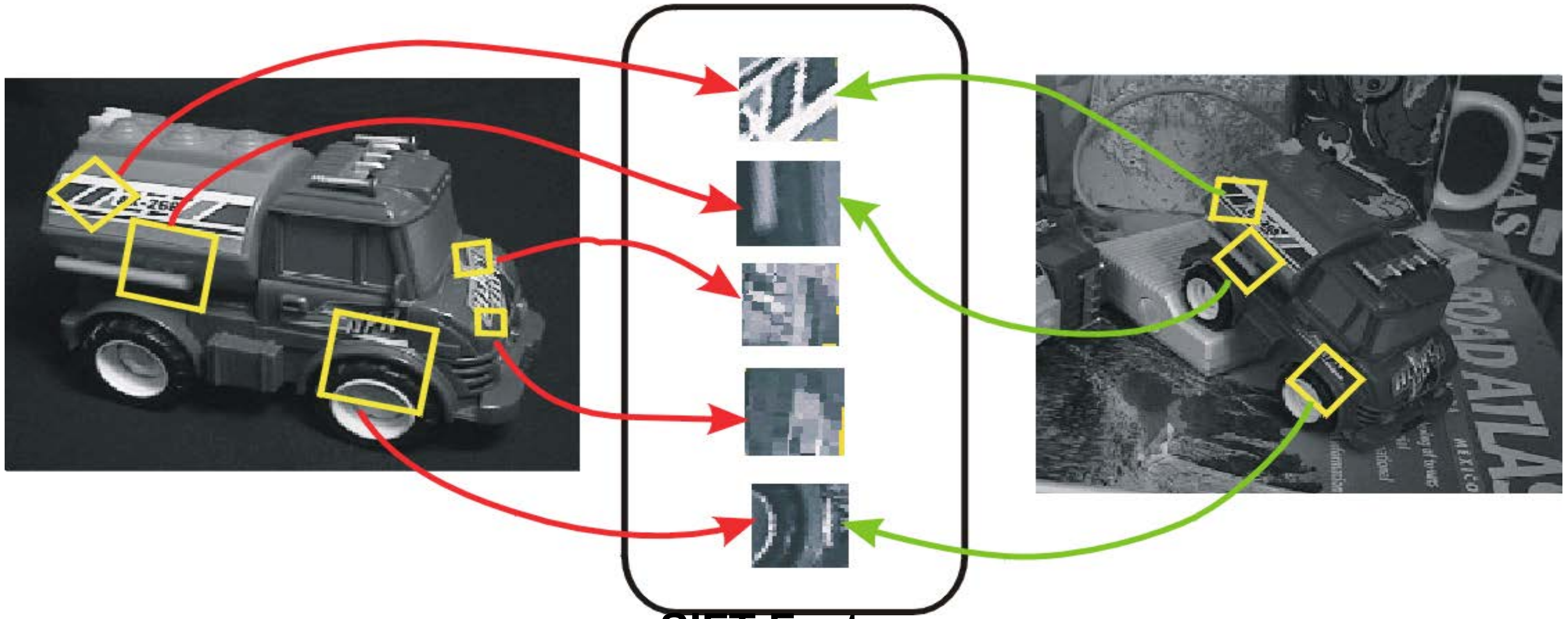
developed by David Lowe
University of British Columbia
Initial paper ICCV 1999
Newer journal paper IJCV 2004

SIFT: Motivation

- The Harris operator is not invariant to scale and correlation is not invariant to rotation[1].
- For better 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 was robust to the variations corresponding to typical viewing conditions. The descriptor is the most-used part of SIFT.

Idea of SIFT

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



SIFT Features

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 detrmine location and scale.
   Select keypoints based on a measure of stability.

3. Orientation assignment

   Compute best orientation(s) for each keypoint region.

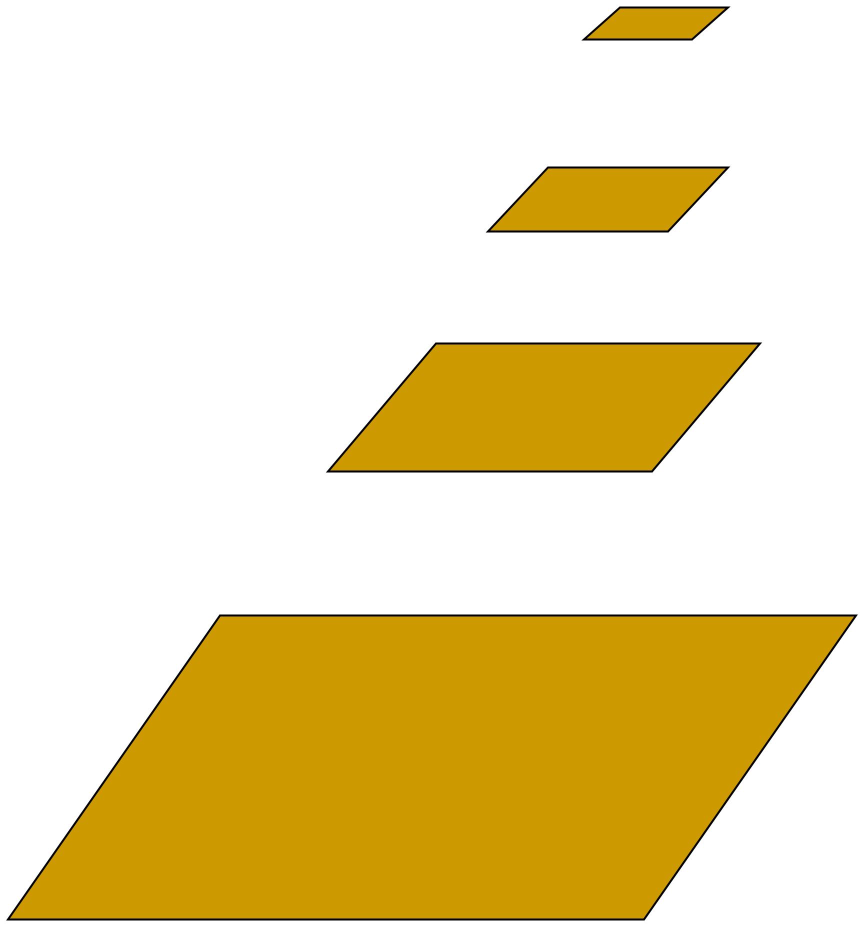
4. Keypoint description

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

1. Scale-space extrema detection

- Goal: Identify locations and scales that can be repeatably assigned under different views of the same scene or object.
- Method: search for stable features across multiple scales using a continuous function of scale.
- Prior work has shown that under a variety of assumptions, the best function is a Gaussian function.
- The scale space of an image is a function $L(x,y,\sigma)$ that is produced from the convolution of a Gaussian kernel (at different scales) with the input image.

Aside: Image Pyramids



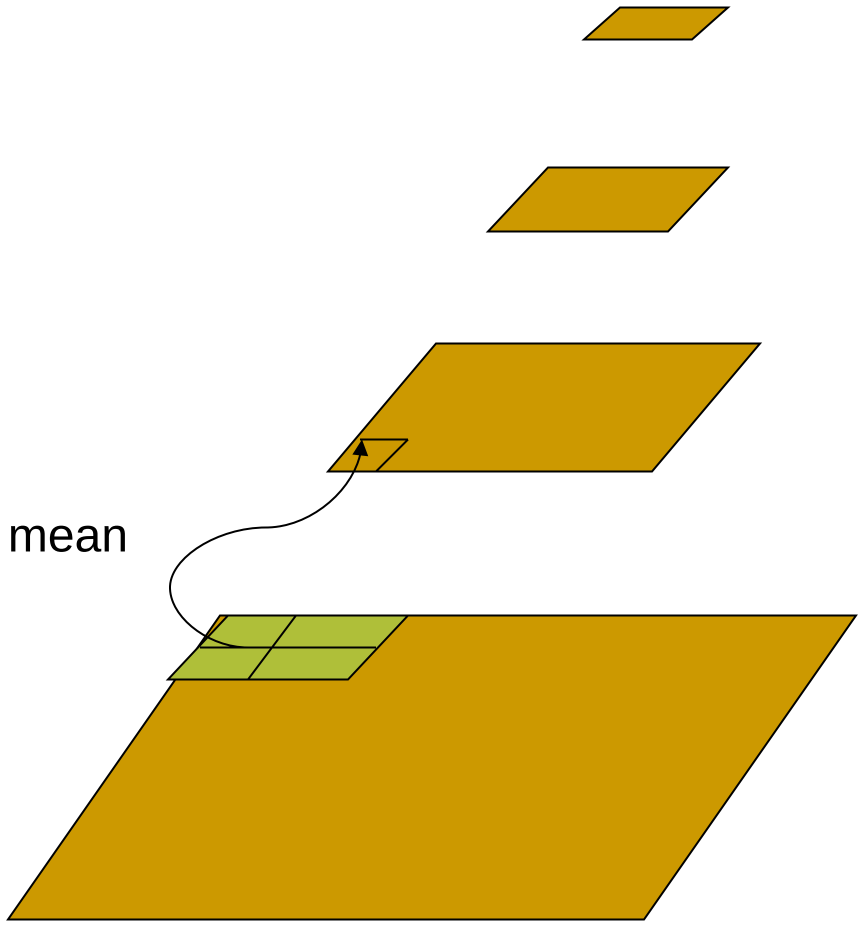
And so on.

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.

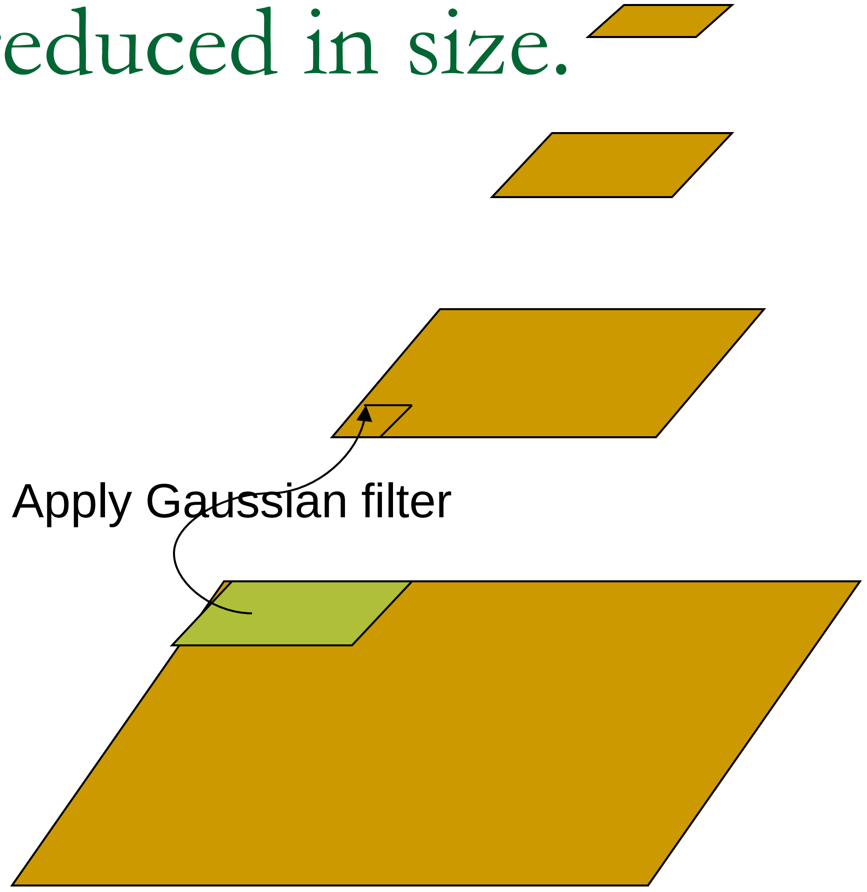
At 3rd level, each pixel is the mean of 4 pixels in the 2nd level.

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.

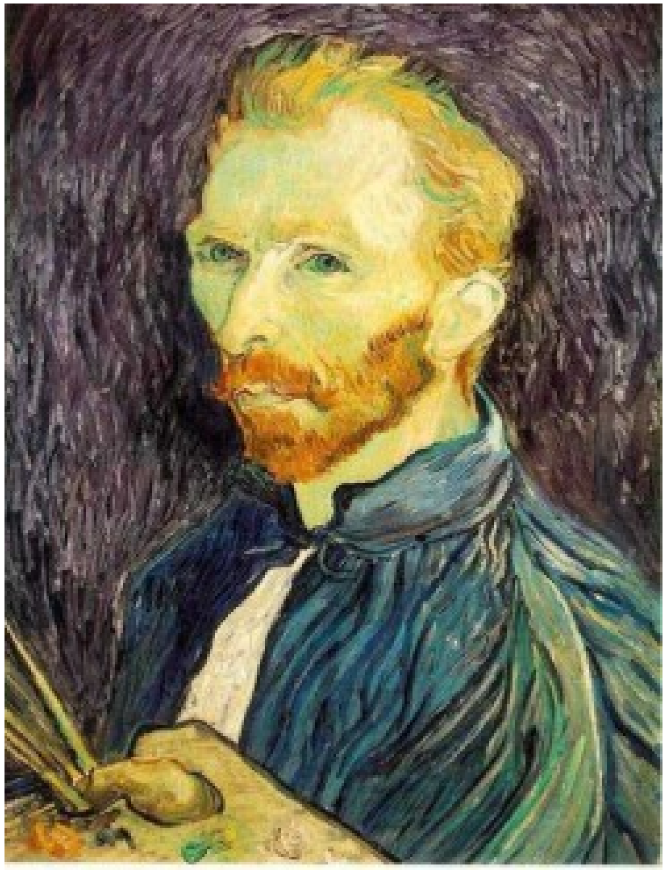


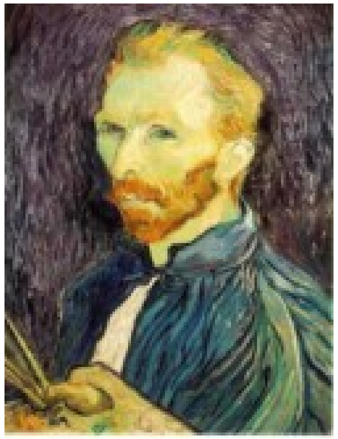
And so on.

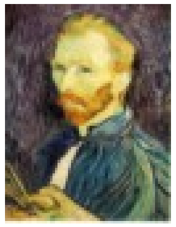
At 2nd level, each pixel is the result of applying a Gaussian mask to the first level and then subsampling to reduce the size.

Bottom level is the original image.

Example: Subsampling with Gaussian pre-filtering

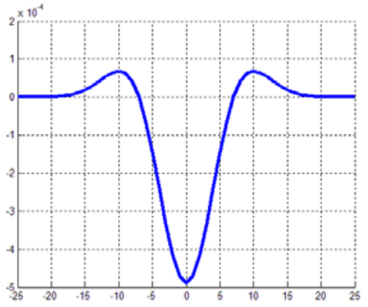

Gaussian 1/2

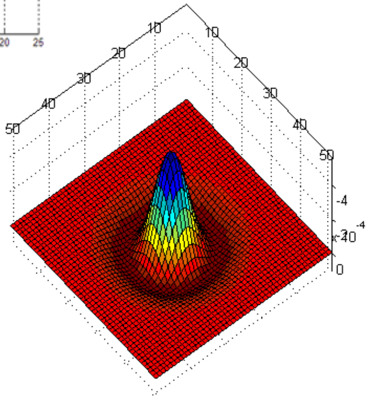

G 1/4

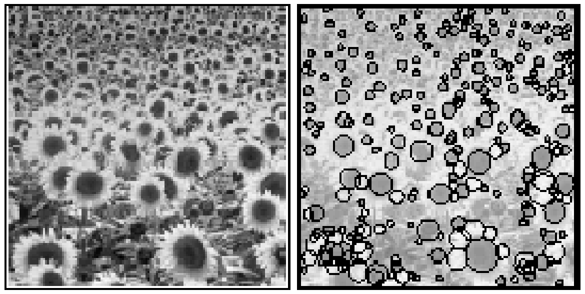

G 1/8

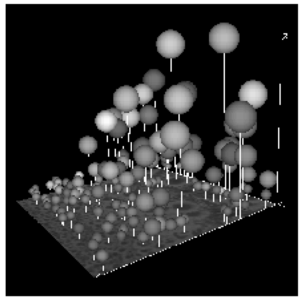
Lowe's Scale-space Interest Points

- **Laplacian of Gaussian** kernel
  - Scale normalised (x by scale^2)
  - Proposed by Lindeberg
- Scale-space detection
  - Find local maxima across scale/space
  - A good "blob" detector







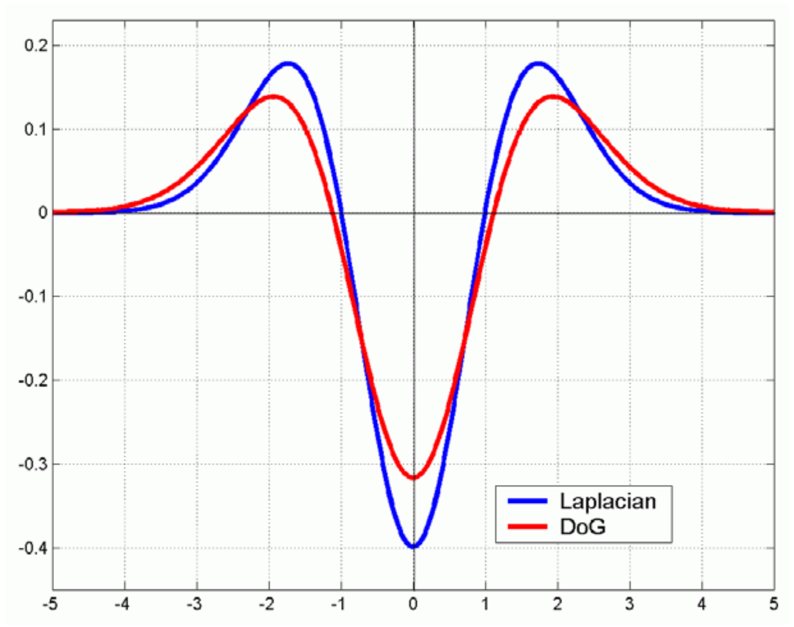


$$G(x,y,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}}$$

$$\nabla^2 G(x,y,\sigma) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

[T. Lindeberg IJCV 1998]

Lowe's Scale-space Interest Points: Difference of Gaussians



- Gaussian is an ad hoc solution of heat diffusion equation

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G.$$

- Hence

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 G.$$

- k is not necessarily very small in practice

Lowe's Pyramid Scheme

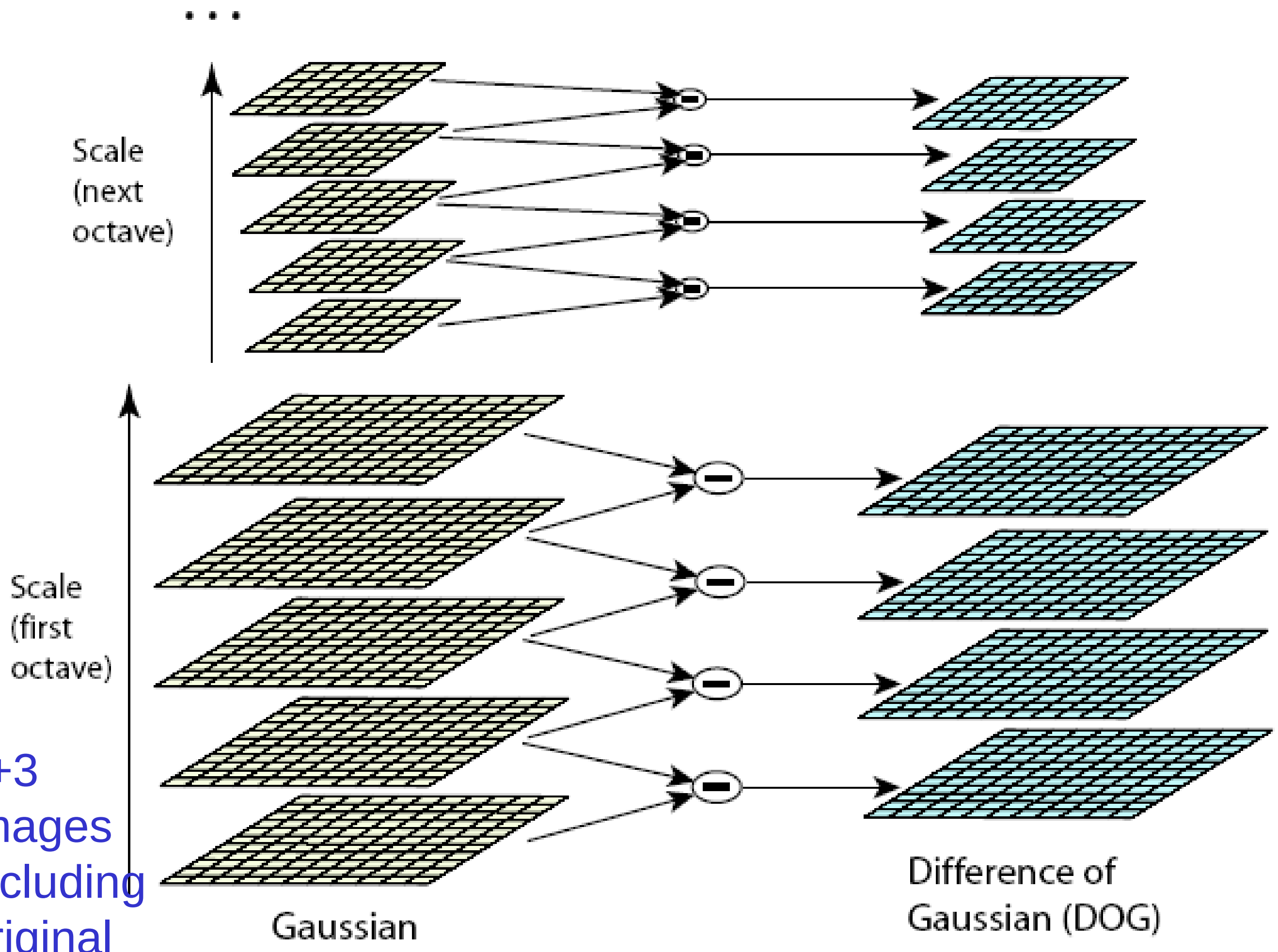
- Scale space is separated into octaves:
  - Octave 1 uses scale σ
  - Octave 2 uses scale 2σ
  - etc.

- In each octave, the initial image is repeatedly convolved with Gaussians to produce a set of scale space images.

- Adjacent Gaussians are subtracted to produce the DOG

- After each octave, the Gaussian image is down-sampled by a factor of 2 to produce an image ¼ the size to start the next level.

Lowe's Pyramid Scheme



s+2 filters

$\sigma_{s+1}=2^{(s+1)/s}\sigma_0$

.

.

$\sigma_i=2^{i/s}\sigma_0$

.

.

$\sigma_2=2^{2/s}\sigma_0$

$\sigma_1=2^{1/s}\sigma_0$

σ_0

s+3 images including original

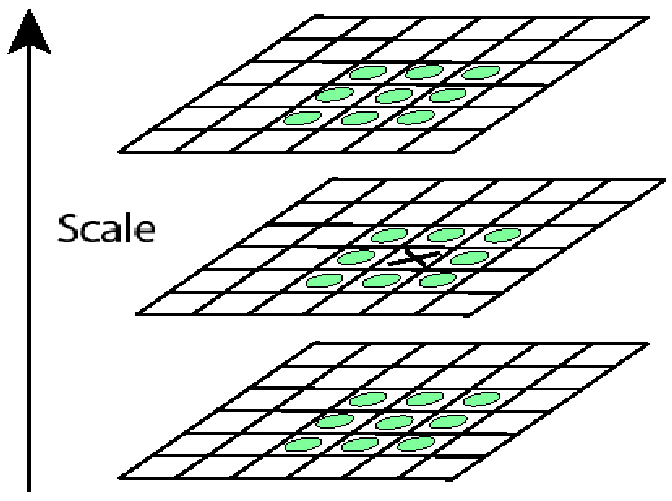
s+2 differ-ence images

The parameter **s** determines the number of images per octave.

2. Key point localization

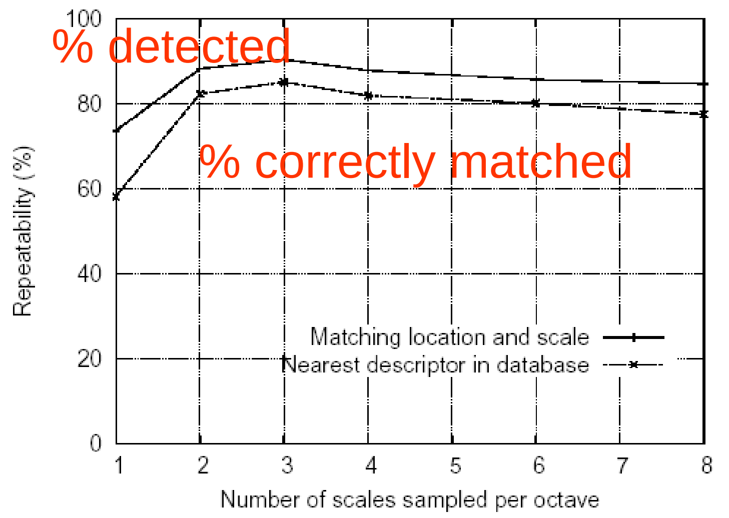
s+2 difference images.
top and bottom ignored.
s planes searched.

- Detect maxima and minima of difference-of-Gaussian in scale space
- Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below

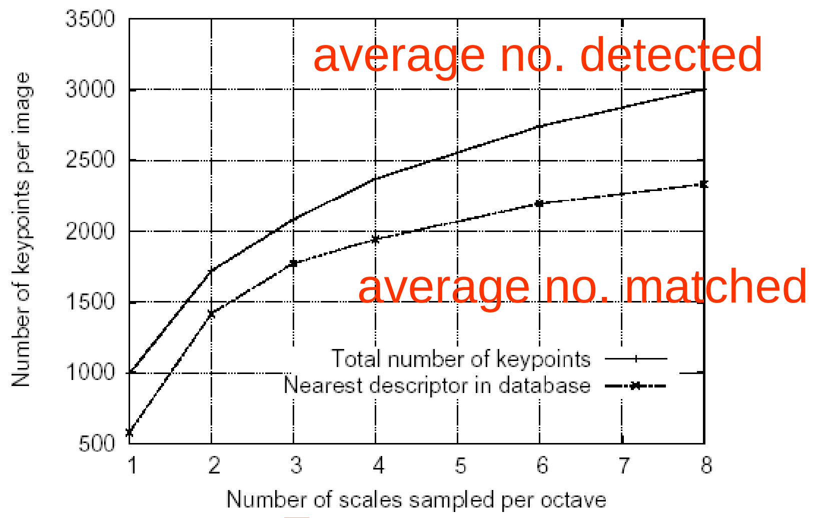


For each max or min found, output is the **location** and the **scale**.

Scale-space extrema detection: experimental results over 32 images that were synthetically transformed and noise added.



Stability



Expense

- Sampling in scale for efficiency
  - How many scales should be used per octave? S=?
    - More scales evaluated, more keypoints found
    - S < 3, stable keypoints increased too
    - S > 3, stable keypoints decreased
    - S = 3, maximum stable keypoints found

More Keypoint localization

- Once a keypoint candidate is found, perform a detailed fit to nearby data to determine
  - location, scale, and ratio of principal curvatures
- In initial work keypoints were found at location and scale of a central sample point.
- In newer work, they fit a 3D quadratic function to improve interpolation accuracy.
- The Hessian matrix was used to eliminate edge responses.

Eliminating the Edge Response

- Reject flats:
  - $|D(\hat{\mathbf{x}})|$ < 0.03
- Reject edges:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Let α be the eigenvalue with larger magnitude and β the smaller.

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

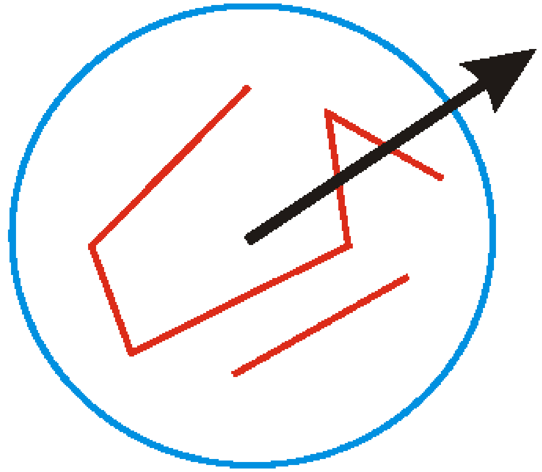
Let r = α/β.
So α = rβ

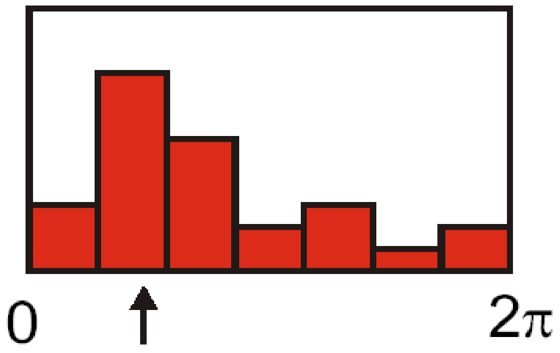
$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r},$$

$(r+1)^2/r$ is at a min when the 2 eigenvalues are equal.

  - r < 10
- What does this look like?

3. Orientation assignment

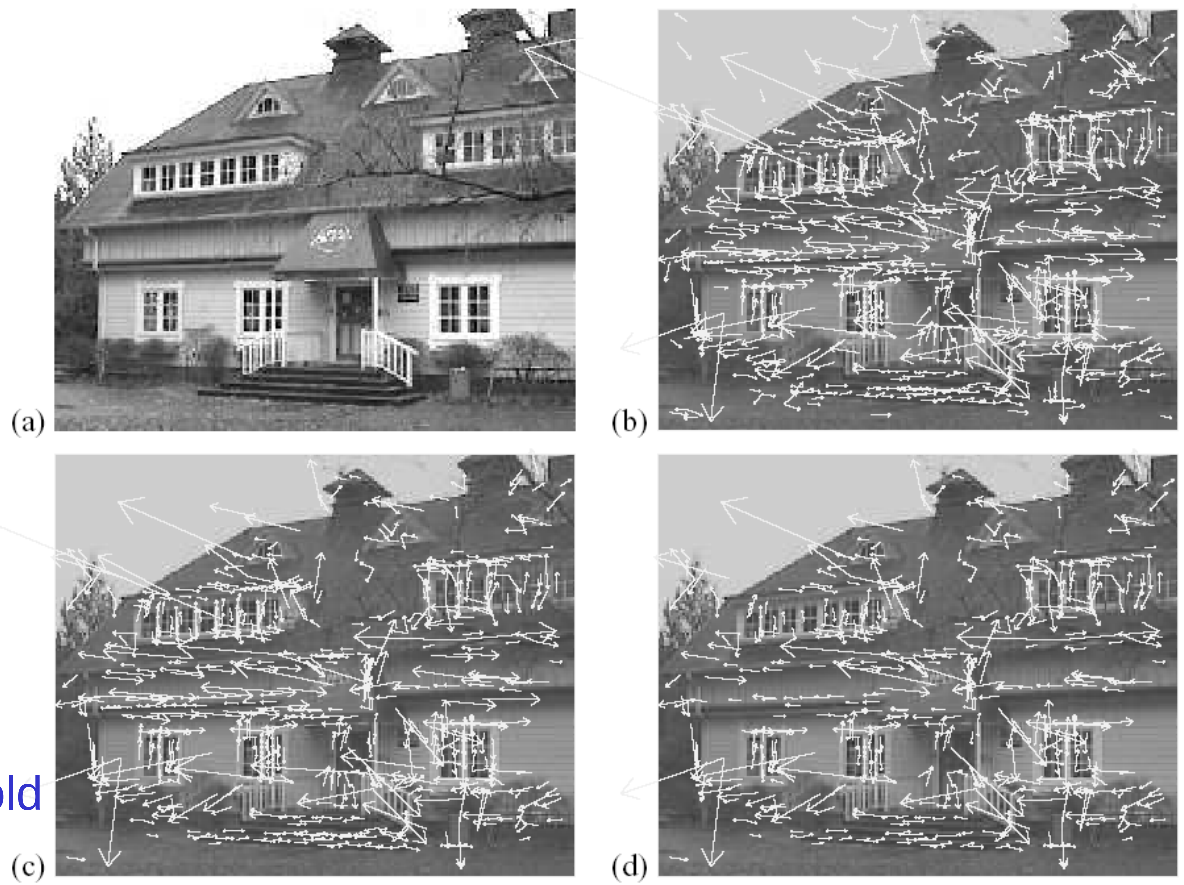




- Create histogram of local gradient directions at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale,orientation)

If 2 major orientations, use both.

Keypoint localization with orientation



233x189

832

initial keypoints

729

keypoints after gradient threshold

536

keypoints after ratio threshold

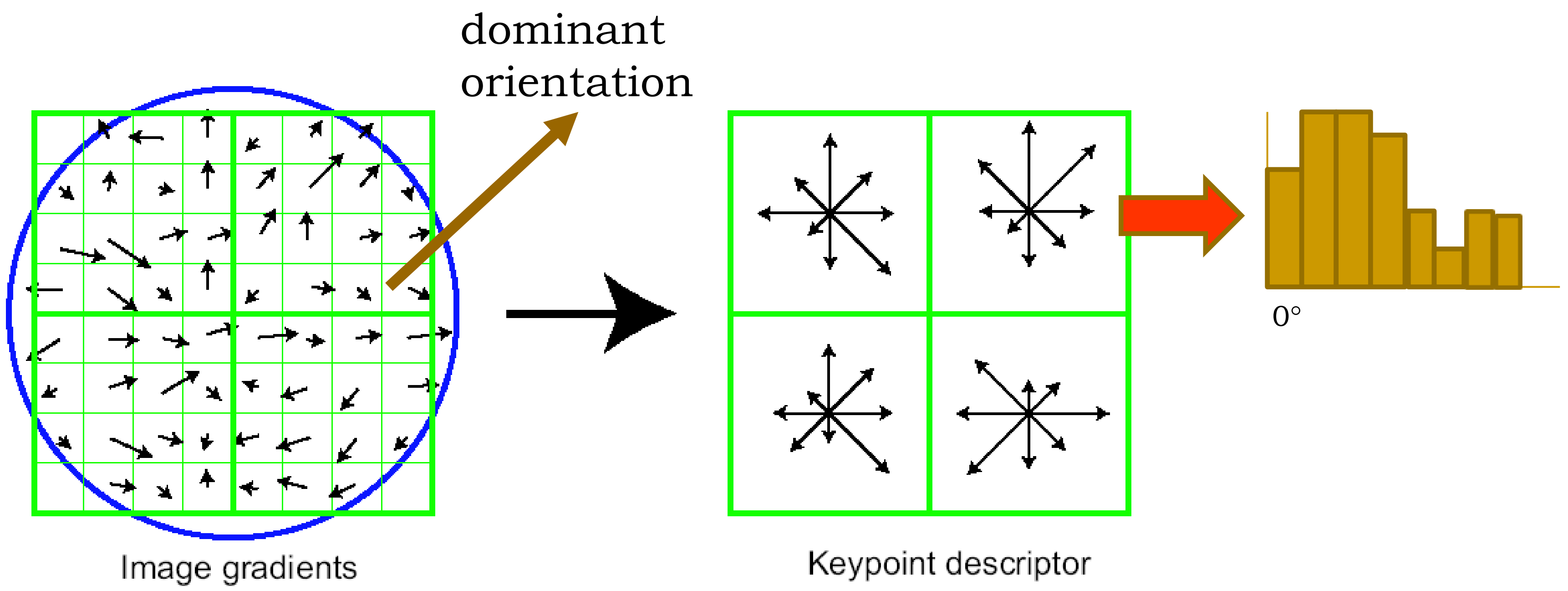
4. Keypoint Descriptors

- At this point, each keypoint has
  - location
  - scale
  - orientation
- Next is to compute a descriptor for the local image region about each keypoint that is
  - highly distinctive
  - invariant as possible to variations such as changes in viewpoint and illumination

Normalization

- Rotate the window to standard orientation
- Scale the window size based on the scale at which the point was found.

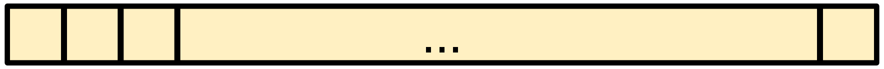
Lowe's Keypoint Descriptor (shown with 2 X 2 descriptors over 8 X 8)



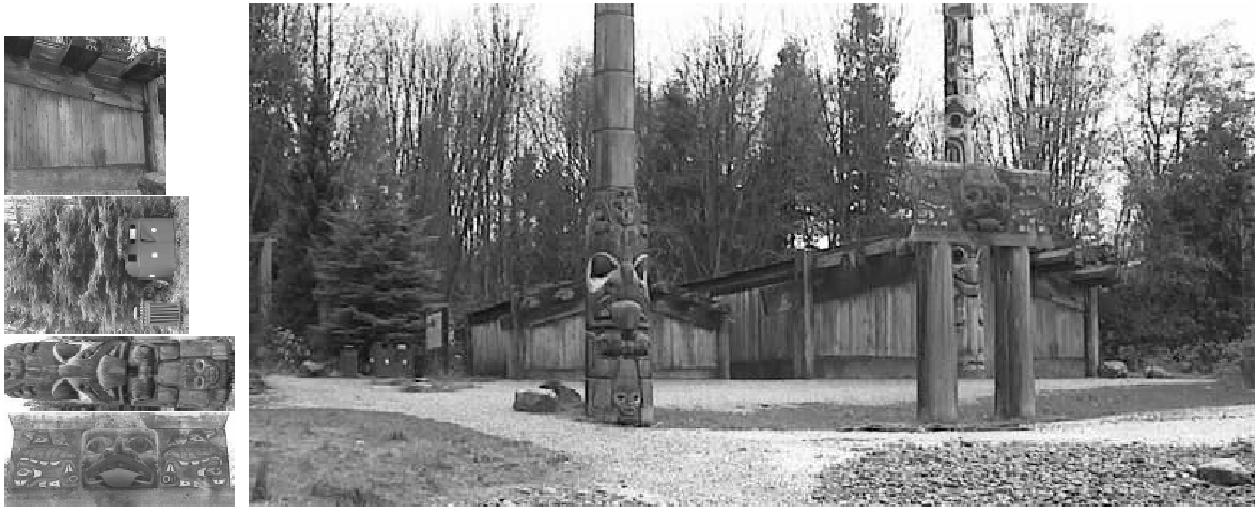
In experiments, **4x4 arrays of 8 bin histogram** is used, a total of 128 features for one keypoint

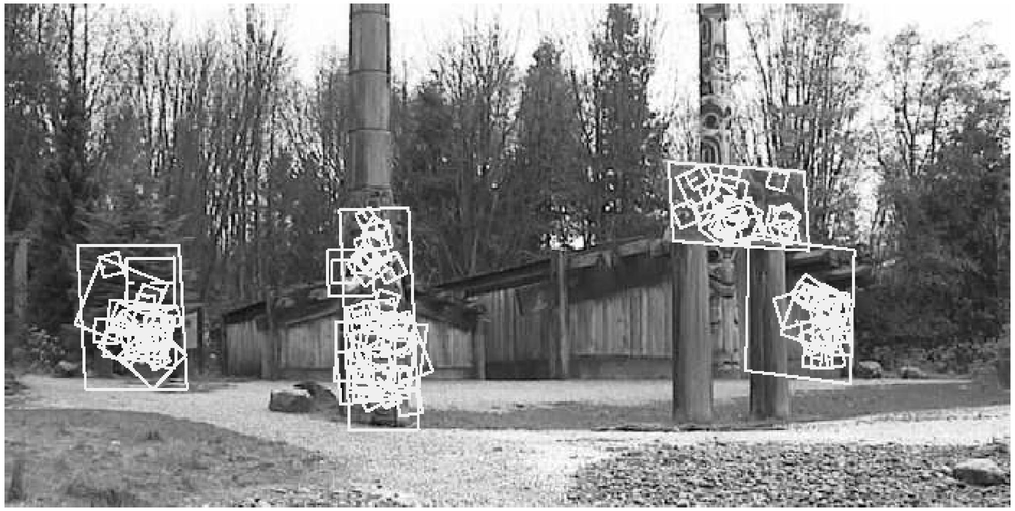
Lowe's Keypoint Descriptor

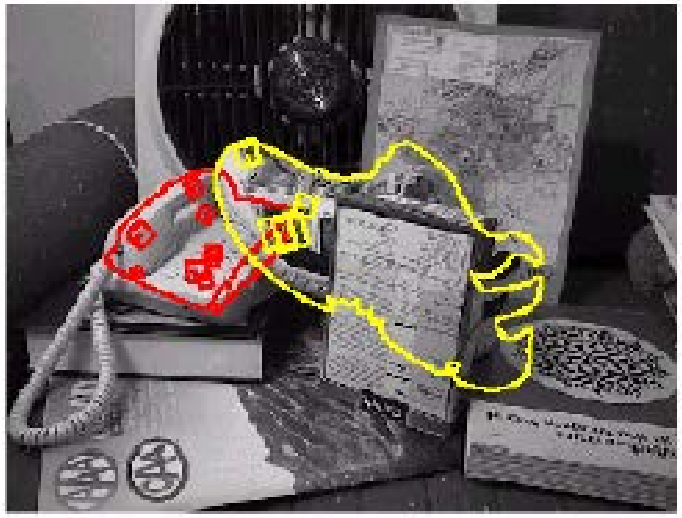
- use the normalized region about the keypoint
- compute gradient magnitude and orientation at each point in the region
- weight them by a Gaussian window overlaid on the circle
- create an orientation histogram over the 4 X 4 subregions of the window
- 4 X 4 descriptors over 16 X 16 sample array were used in practice. 4 X 4 times 8 directions gives a vector of 128 values.

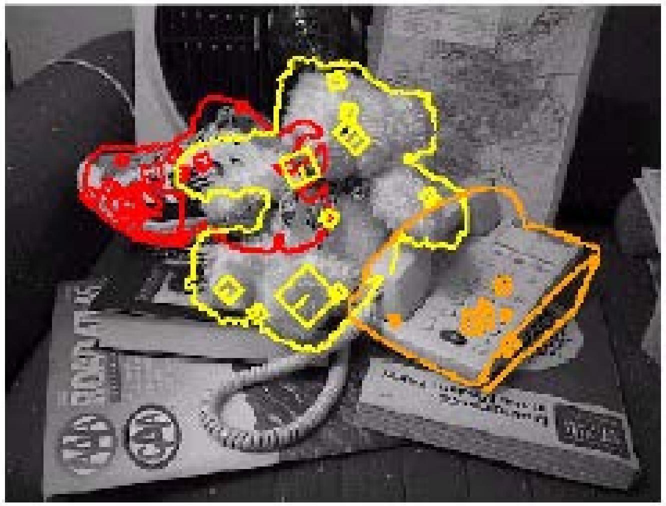


Using SIFT for Matching “Objects”









Uses for SIFT

- Feature points are used also for:
  - Image alignment (homography, fundamental matrix)
  - 3D reconstruction (e.g. Photo Tourism)
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - … many others

[Photo Tourism: Snavely et al. SIGGRAPH 2006]

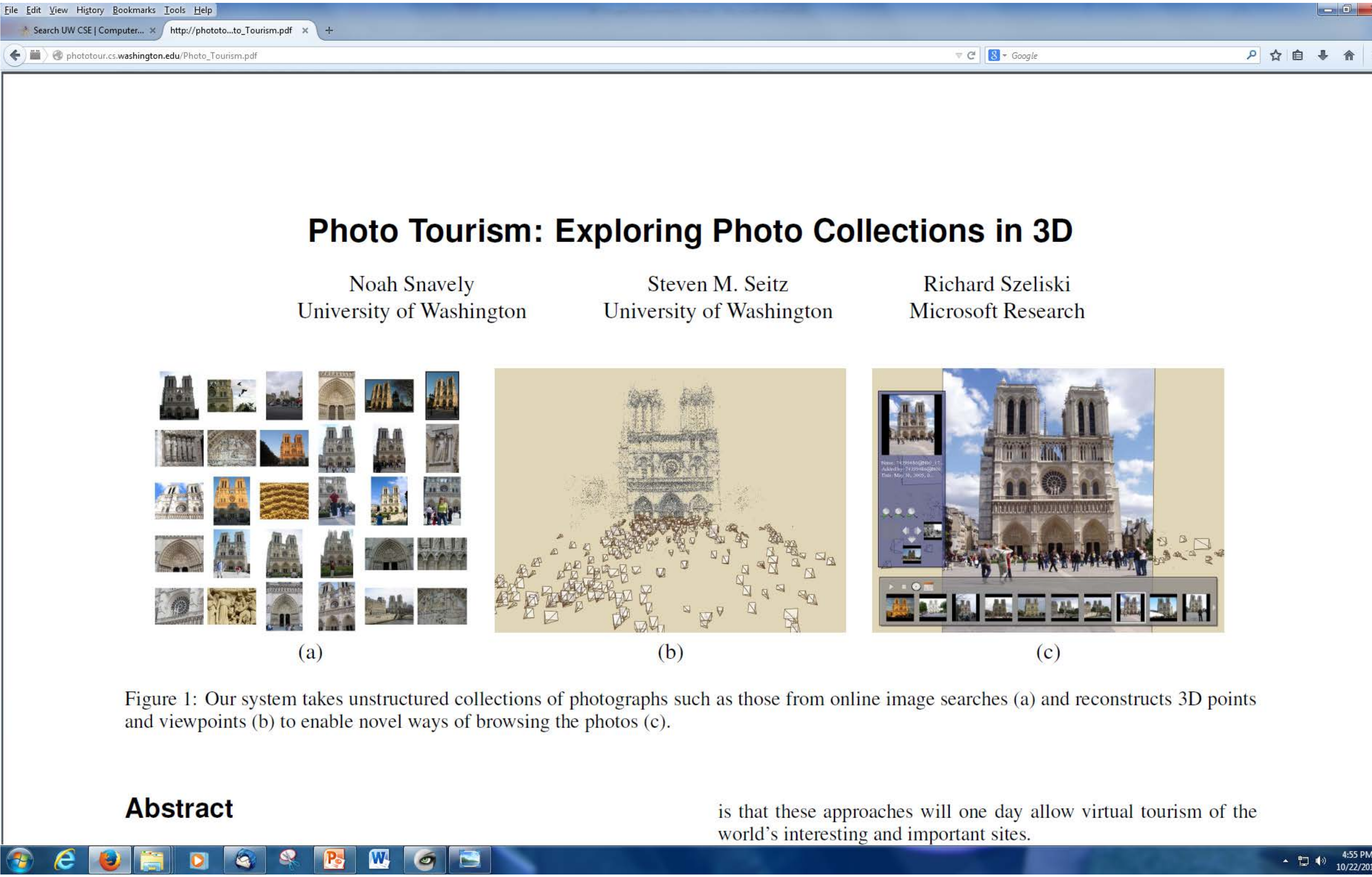
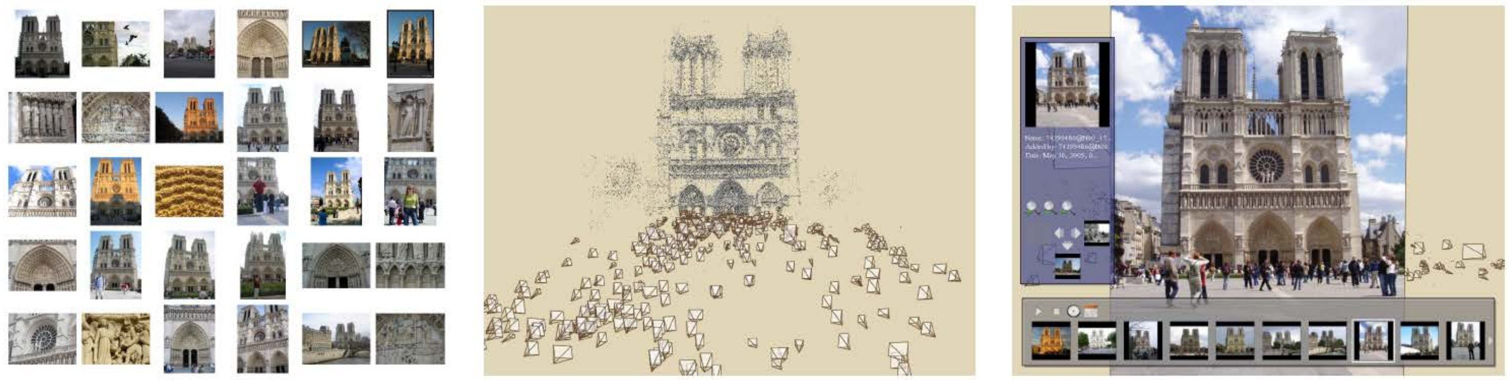

File Edit View History Bookmarks Tools Help
Search UW CSE | Computer...
http://phototo...to_Tourism.pdf
phototour.cs.washington.edu/Photo_Tourism.pdf
Google
4:55 PM
10/22/20

Photo Tourism: Exploring Photo Collections in 3D

Noah Snavely
University of Washington

Steven M. Seitz
University of Washington

Richard Szeliski
Microsoft Research



(a) (b) (c)

Figure 1: Our system takes unstructured collections of photographs such as those from online image searches (a) and reconstructs 3D points and viewpoints (b) to enable novel ways of browsing the photos (c).

Abstract

is that these approaches will one day allow virtual tourism of the world's interesting and important sites.

Another Descriptor

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs

CVPR 2005

Overview

1. Compute gradients in the region to be described
2. Put them in bins according to orientation
3. Group the cells into large blocks
4. Normalize each block
5. Train classifiers to decide if these are parts of a human

Details

- Gradients

  [-1 0 1] and $[-1\ 0\ 1]^T$ were good enough.

- Cell Histograms

  Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. (9 channels worked)

- Blocks

  Group the cells together into larger blocks, either R-HOG blocks (rectangular) or C-HOG blocks (circular).

More Details

- Block Normalization

They tried 4 different kinds of normalization.
Let υ be the block to be normalized and e be a small constant.

L2-norm: $f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}$

L2-hys: L2-norm followed by clipping (limiting the maximum values of v to 0.2) and renormalizing,

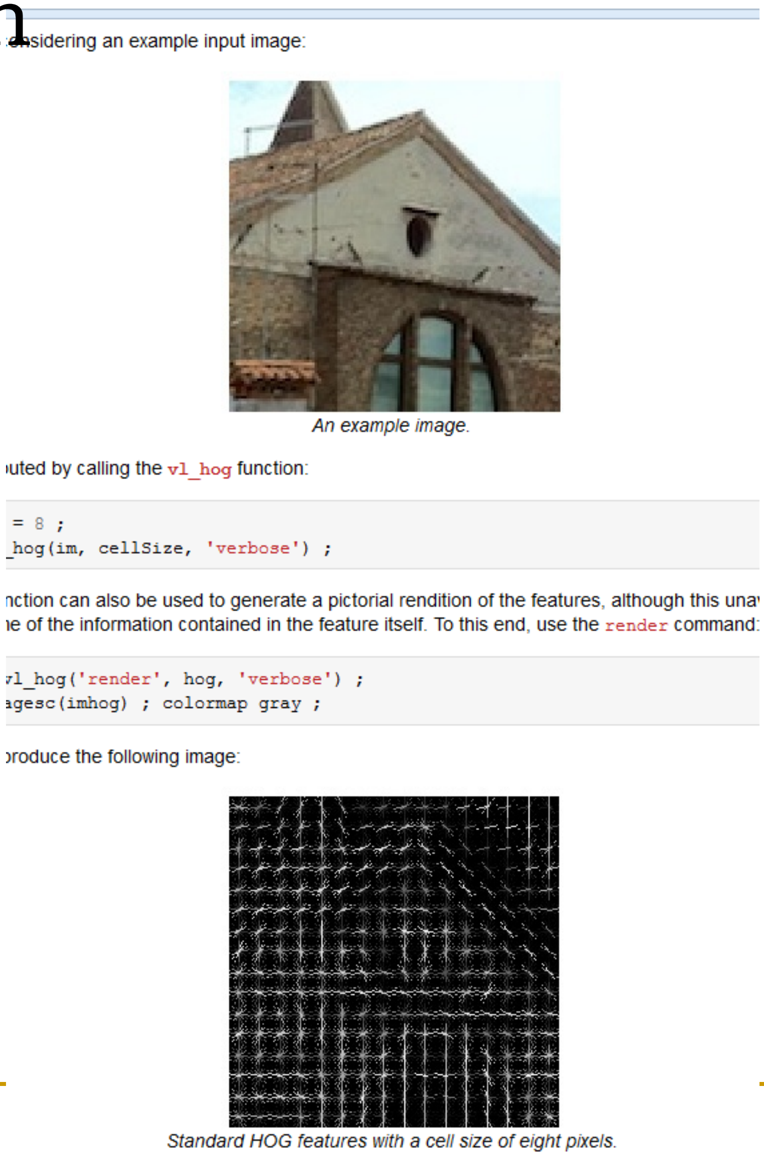
L1-norm: $f = \frac{v}{(\|v\|_1 + e)}$

L1-sqrt: $f = \sqrt{\frac{v}{(\|v\|_1 + e)}}$

R-HOG compared to SIFT Descriptor

- R-HOG blocks appear quite similar to the SIFT descriptors.
- But, R-HOG blocks are computed in dense grids at some single scale without orientation alignment.
- SIFT descriptors are computed at sparse, scale-invariant key image points and are rotated to align orientation.
- Much simpler, looking for objects, not points, no rotation.

Standard HOG visualization shows orientation



nsidering an example input image:

An example image.

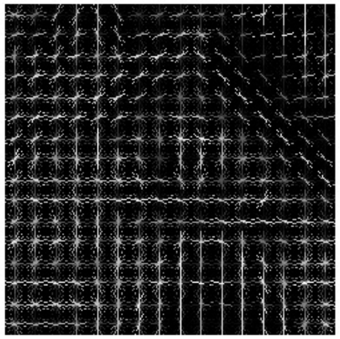
uted by calling the `vl_hog` function:

```
 = 8 ;
_hog(im, cellSize, 'verbose') ;
```

nction can also be used to generate a pictorial rendition of the features, although this una
ie of the information contained in the feature itself. To this end, use the `render` command:

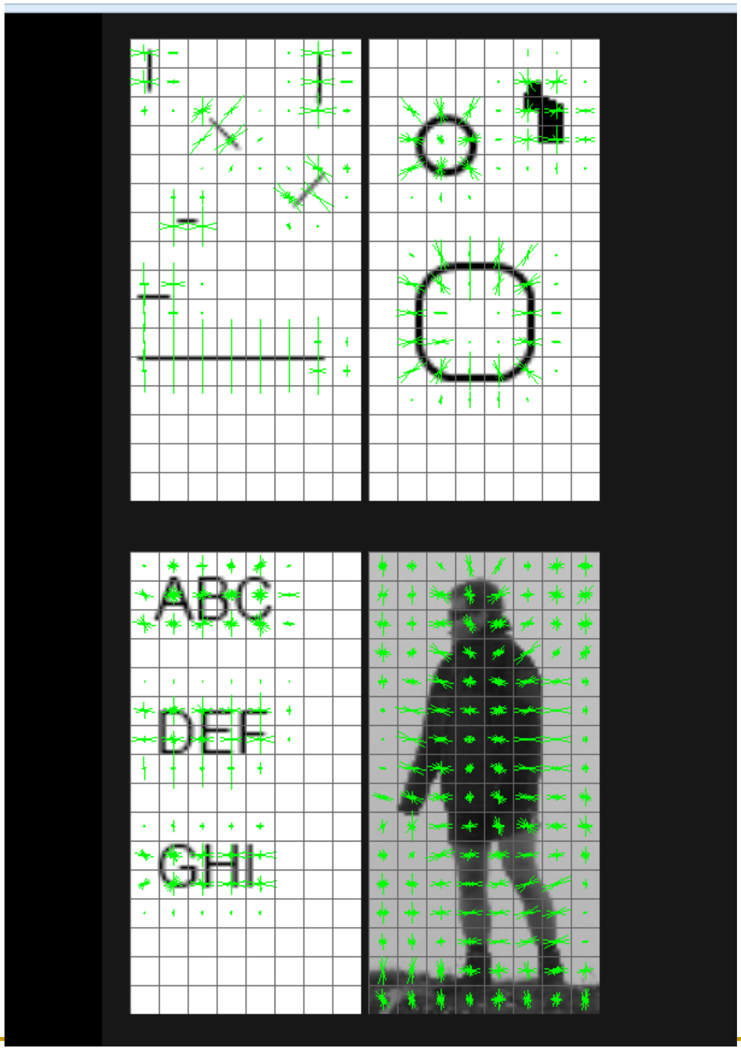
```
vl_hog('render', hog, 'verbose') ;
agesc(imhog) ; colormap gray ;
```

roduce the following image:

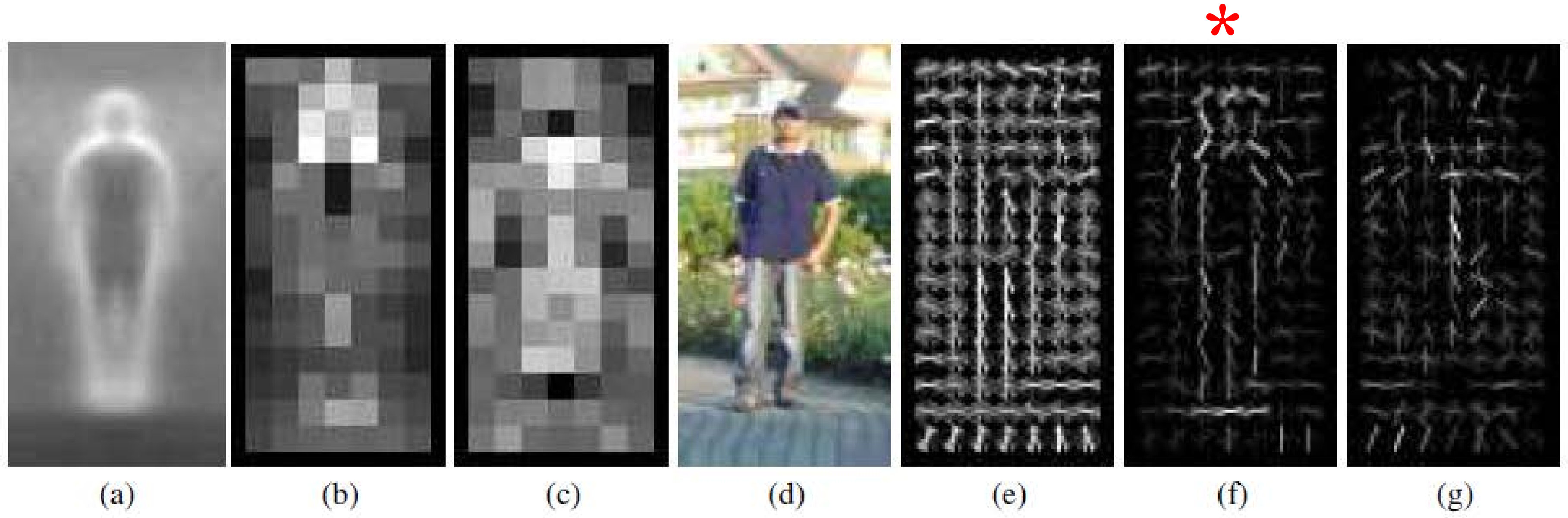


Standard HOG features with a cell size of eight pixels.

Some guy named Juergen's visualizations shows gradient vectors



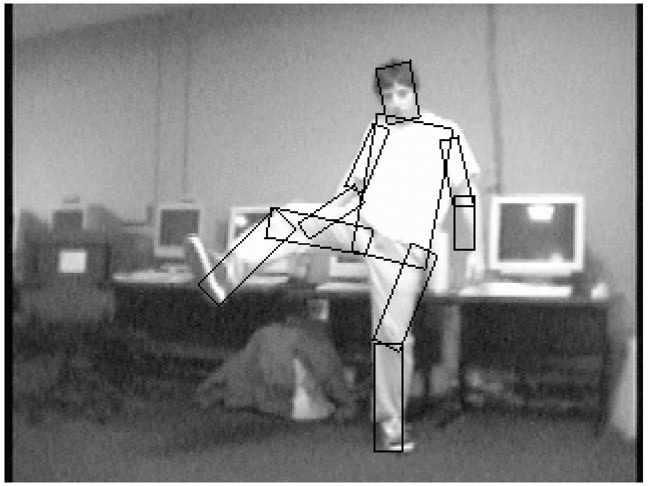
Pictorial Example of HOG for Human Detection



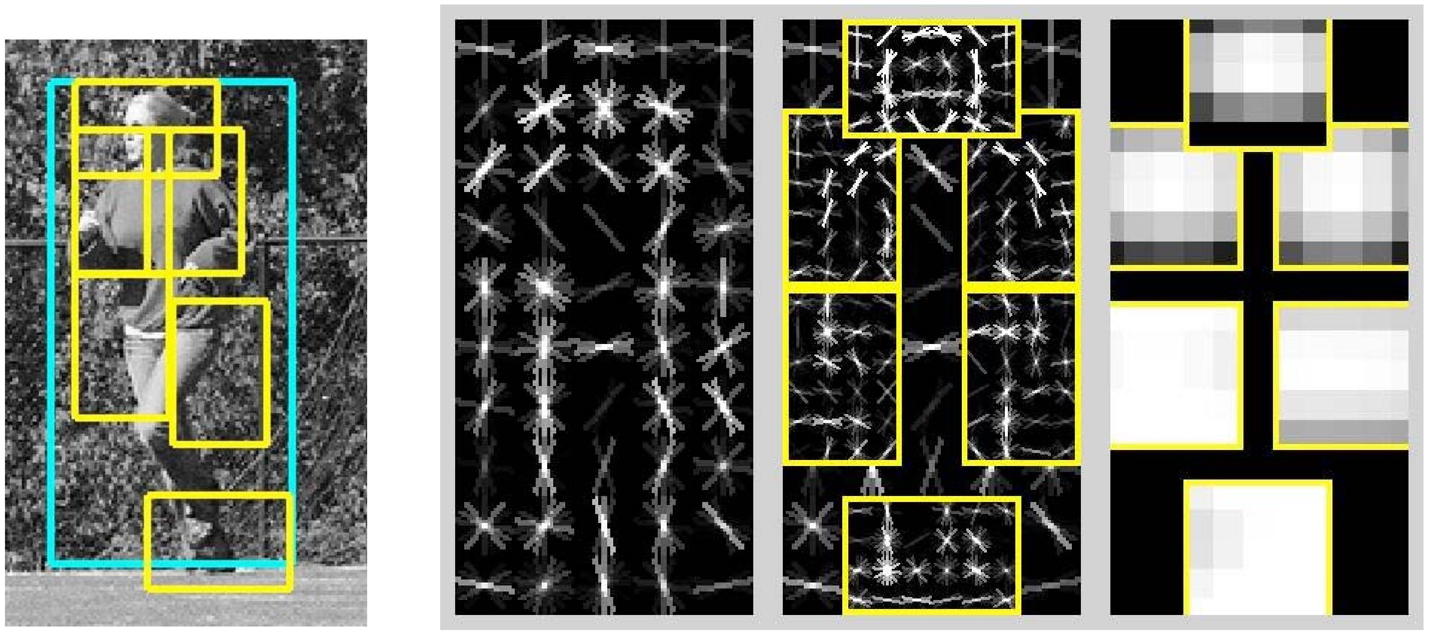
(a) average gradient image over training examples
(b) each "pixel" shows max positive SVM weight in the block centered on that pixel
(c) same as (b) for negative SVM weights
(d) test image
(e) its R-HOG descriptor
(f) R-HOG descriptor weighted by positive SVM weights
(g) R-HOG descriptor weighted by negative SVM weights

Felzenszwalb’s Deformable Models Won the 2007 Pascal Challenge

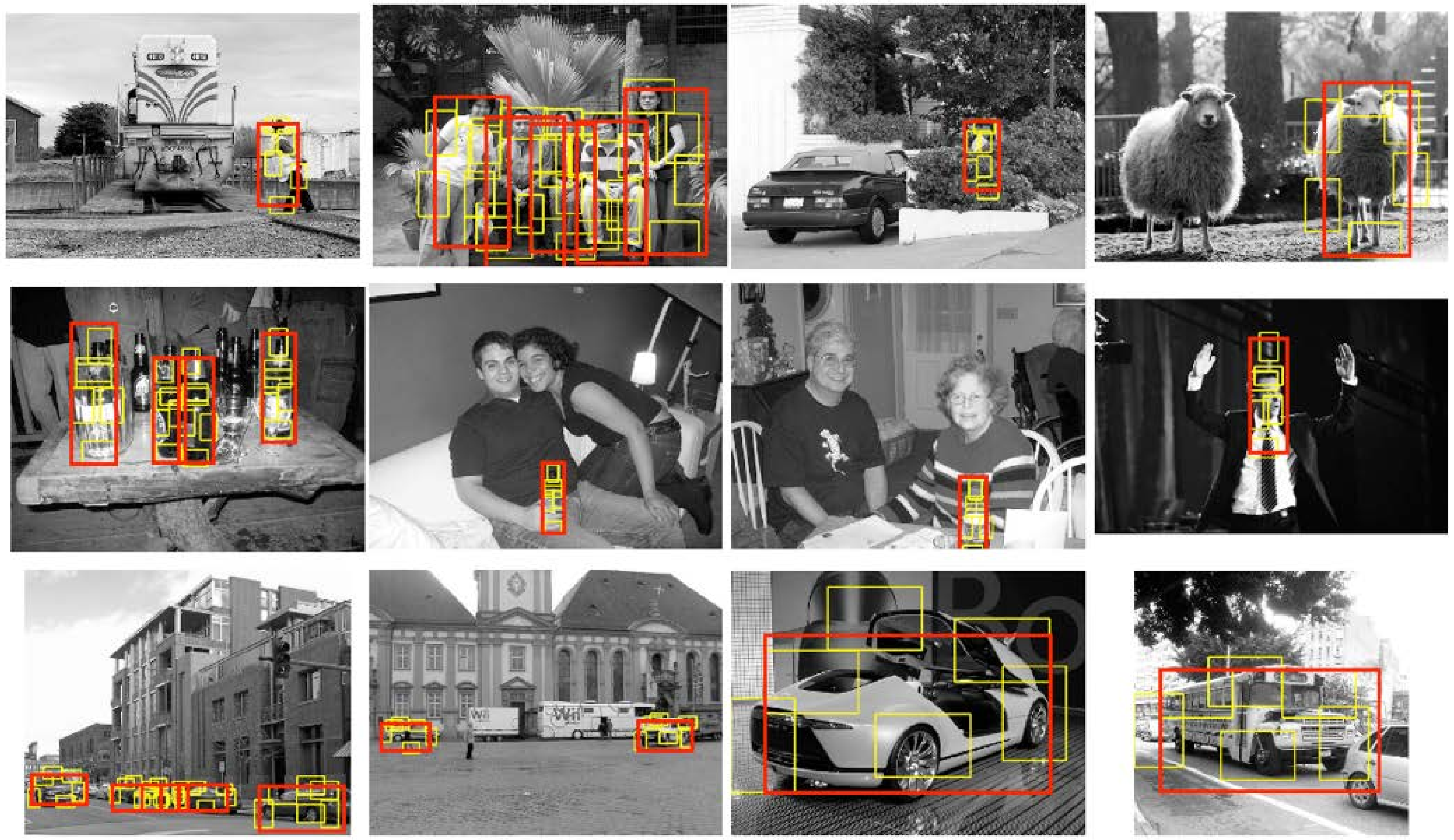
- Use HOG, but represent the object as a collection of parts, each with a HOG model.
- The parts can move, like the limbs of a human move.



Deformable Model for Human



Sample Results of Different Objects



More Results