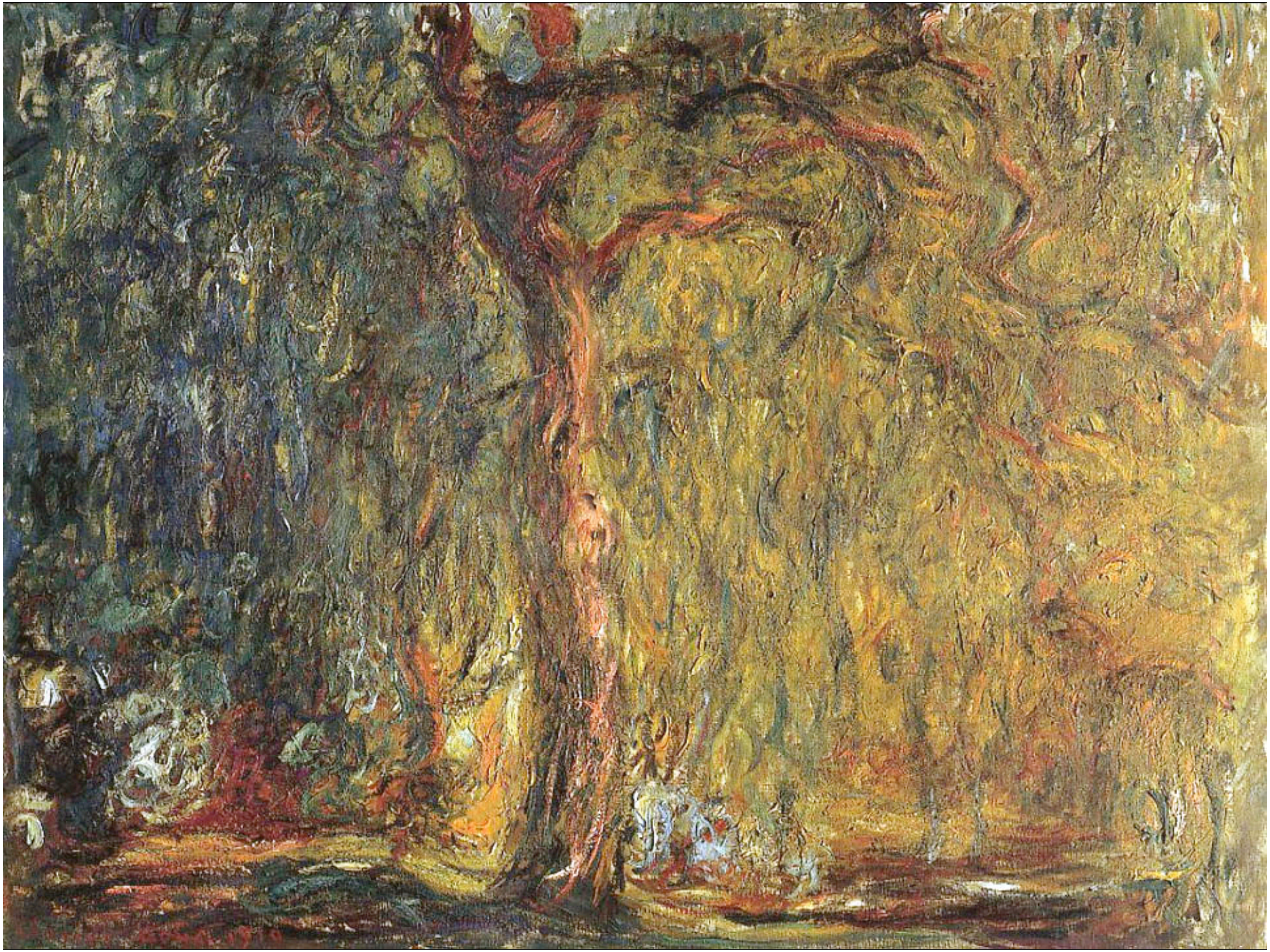


Reconnaissance d'objets et vision artificielle

Josef Sivic (josef.sivic@ens.fr)
http://www.di.ens.fr/~josef

INRIA - Equipe-projet WILLOW
ENS/INRIA/CNRS UMR 8548
Laboratoire d'Informatique
Ecole Normale Supérieure, Paris

The rest of the class today

- Overview of course assignments and the final project (J. Sivic)
- Overview of research at WILLOW and internship possibilities (I. Laptev)

Course assignments

http://www.di.ens.fr/willow/teaching/recvis10/

Assignments

There will be four programming assignments representing 70% of the grade.

1. Scale invariant feature detection. *Hand out date: Oct 5. Due date: Oct 19.*
2. Stitching photo mosaics. *Hand out date: Oct 12. Due date: Nov 2.*
3. Bag-of-features image classification. *Hand out date: Nov 2. Due date: Nov 16.*
4. Simple face detector. *Hand out date: Nov 16. Due date: Nov 30.*

Final project

The final project will represent 30% of the grade.

The topics will be announced in the class and on this page. Talk to us if you want to define your own project (before you start working on the project). Joint projects with other courses are possible, but please talk to instructors of both courses (before you start working on the project).

Hand out date: Nov 2. Project presentations and reports due on: Dec 14.

Course schedule (subject to change)

Lecture	Date	Topic
1	Sep 28	Introduction (J. Ponce, I. Laptev, J. Sivic)
2	Oct 5	Instance-level recognition I. - Local invariant features (C. Schmid) *Assignment 1 out.*
3	Oct 12	Instance-level recognition II. - Correspondence, efficient visual search (J. Sivic) *Assignment 2 out.*
4	Oct 19	Very large scale image indexing. Bag-of-feature models for category-level recognition (C. Schmid) ***Assignment 1 due.***
5	Oct 26	Sparse coding and dictionary learning for image analysis (J. Ponce)
6	Nov 2	Part-based models and pictorial structures for object recognition (J. Sivic) **Assignment 2 due.** *Assignment 3 out. Final projects out.*
7	Nov 9	Motion and human actions I. (I. Laptev)
8	Nov 16	Motion and human actions II. (I. Laptev) ***Assignment 3 due.*** Assignment 4 out.
9	Nov 23	Neural networks; Optimization methods (J. Ponce)
10	Nov 30	Category-level localization; Face detection and recognition (C. Schmid) ***Assignment 4 due.***
11	Dec 7	Multiple object categories; Context; Recognizing large number of object categories; Segmentation (I. Laptev, J. Sivic)
12	Dec 14	Final project presentations and evaluation (J. Sivic, I. Laptev) ***Final project reports due.***

A1: Scale-invariant feature detection

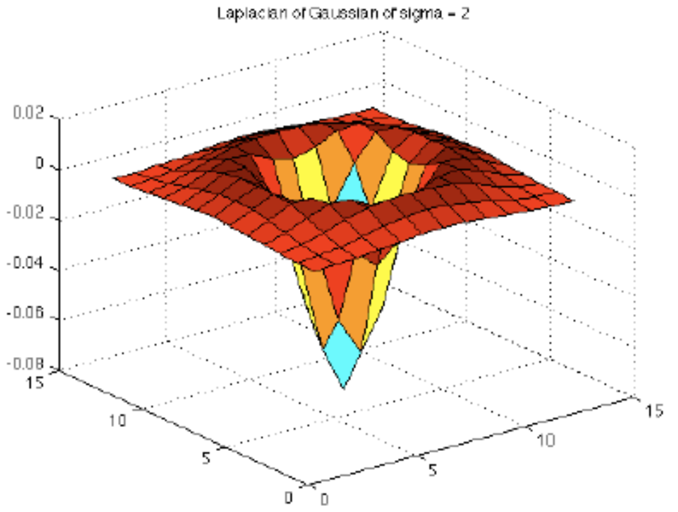
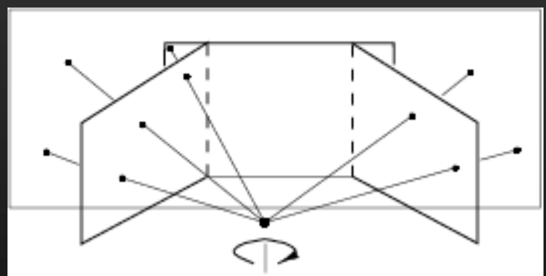


Figure 1: Laplacian of the Gaussian

$$\Delta G(x,y) = \frac{1}{2\pi\sigma^6} e^{\frac{-(x^2+y^2)}{2\sigma^2}} (x^2 + y^2 - 2\sigma^2)$$

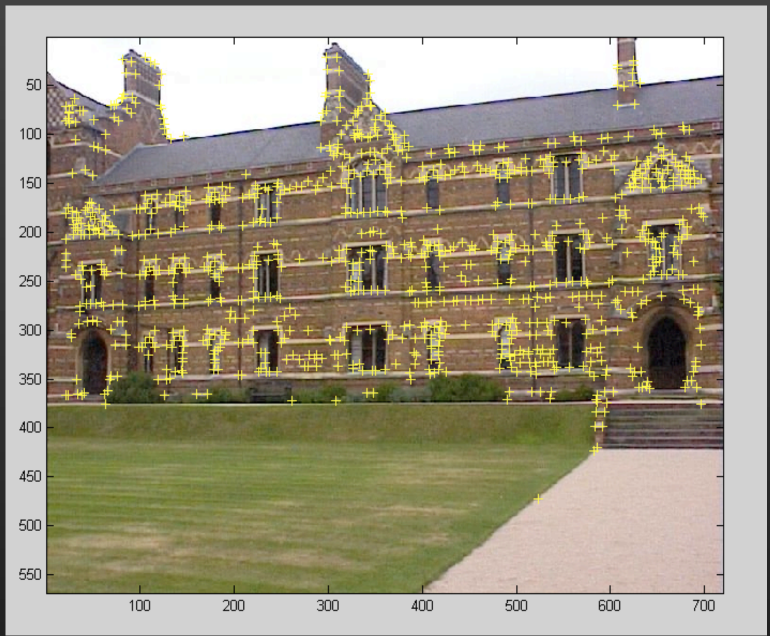
A2: Stitching photo mosaics

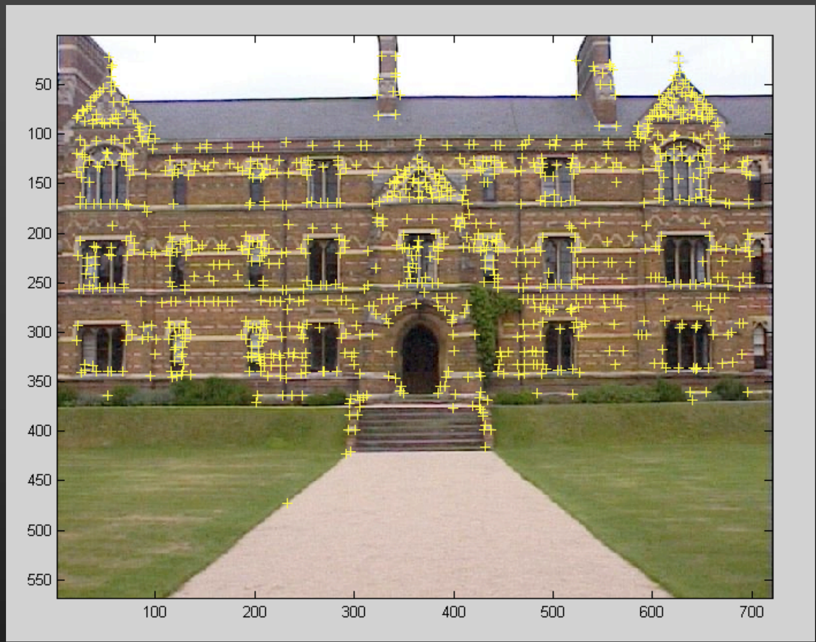
The goal of the assignment is to automatically stitch images acquired by a panning camera into a mosaic.



A2: Stitching photo mosaics

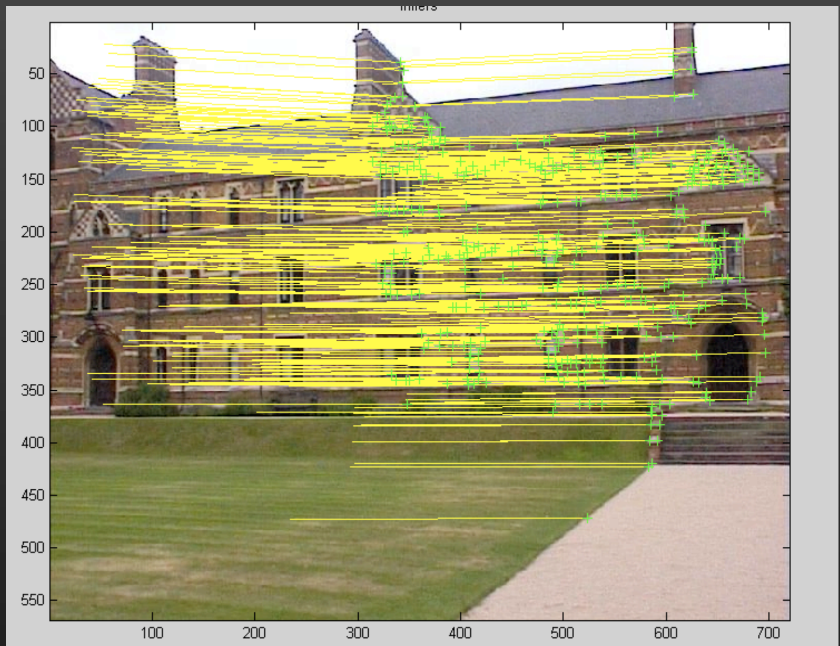
1. Detect feature points (Assignment 1)
2. Match feature points and estimate planar homography between images
3. Warp and composite images into a common reference frame

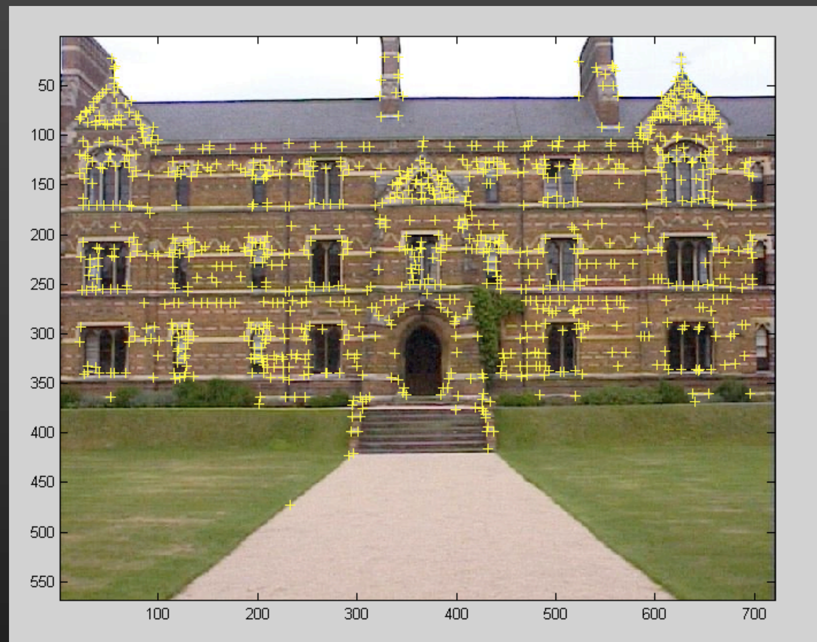




A2: Stitching photo mosaics

1. Detect feature points (Assignment 1)
2. Match feature points and estimate planar homography between images
3. Warp and composite images into a common reference frame





A2: Stitching photo mosaics

1. Detect feature points (Assignment 1)
2. Match feature points and estimate planar homography between images
3. Warp and composite images into a common reference frame

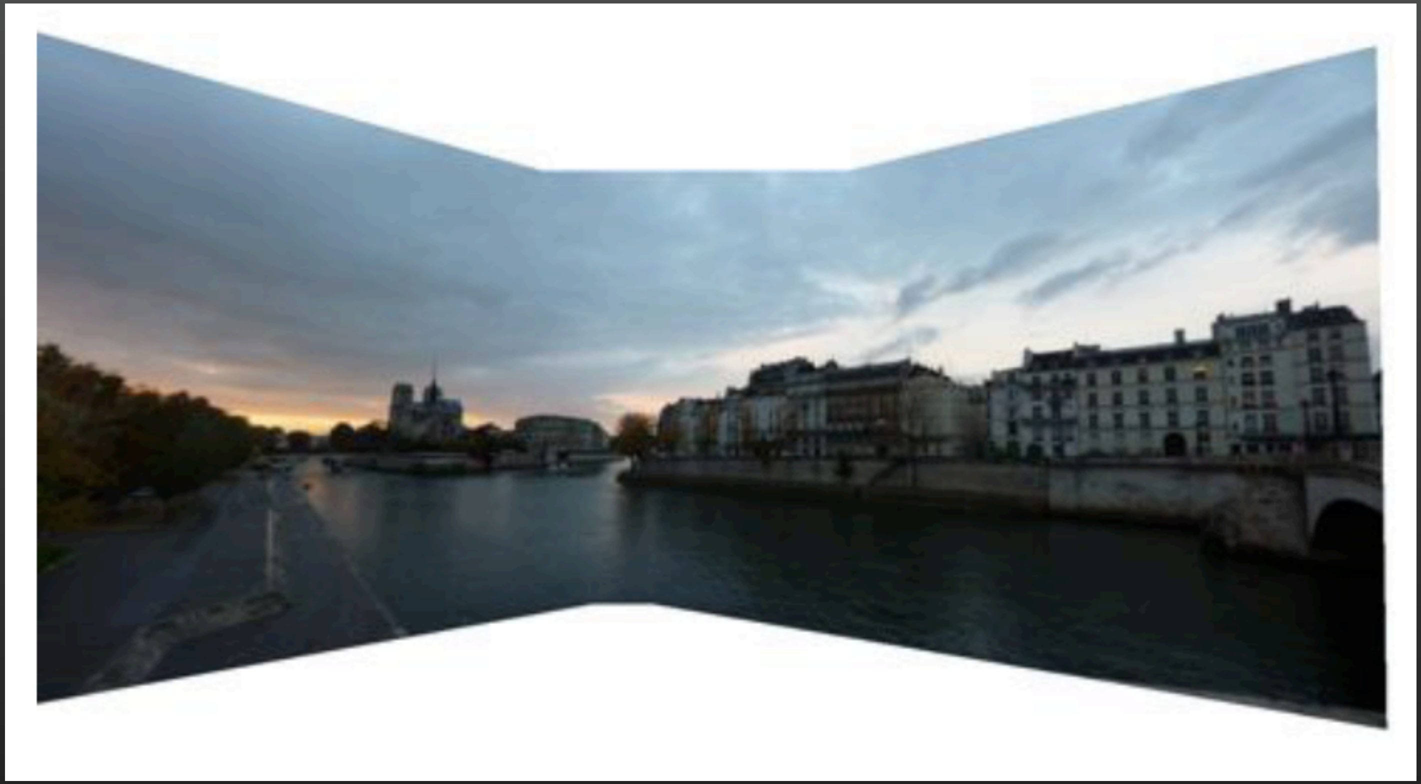
A2: Stitching photo mosaics

Stitch your own images!

Images courtesy: Jose Lezama

A2: Stitching photo mosaics

Stitch your own images!



Images courtesy: Jose Lezama

A3: Bag-of-features image classification

The goal of this assignment is to implement a category-level image classifier to perform four-class image classification into “airplanes”, “motorbikes”, “faces” and “cars” categories.

Training and test images will be provided.

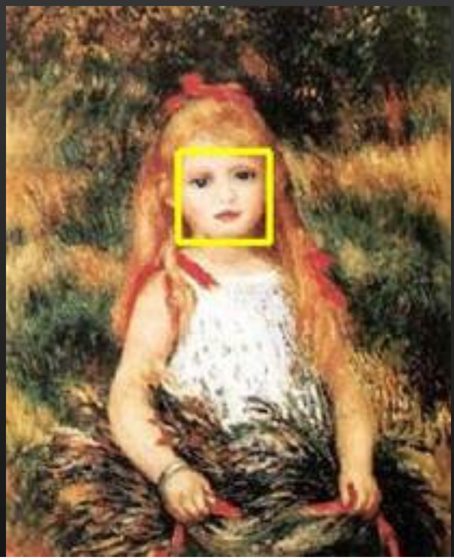
The goal is to achieve at least 90% accuracy and compare performance with a baseline image representation.

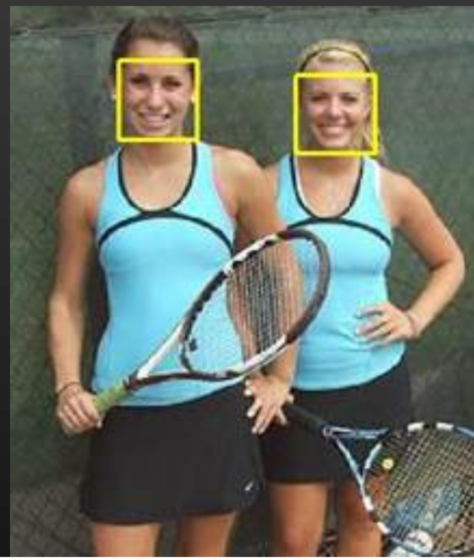
A4: Simple face detector

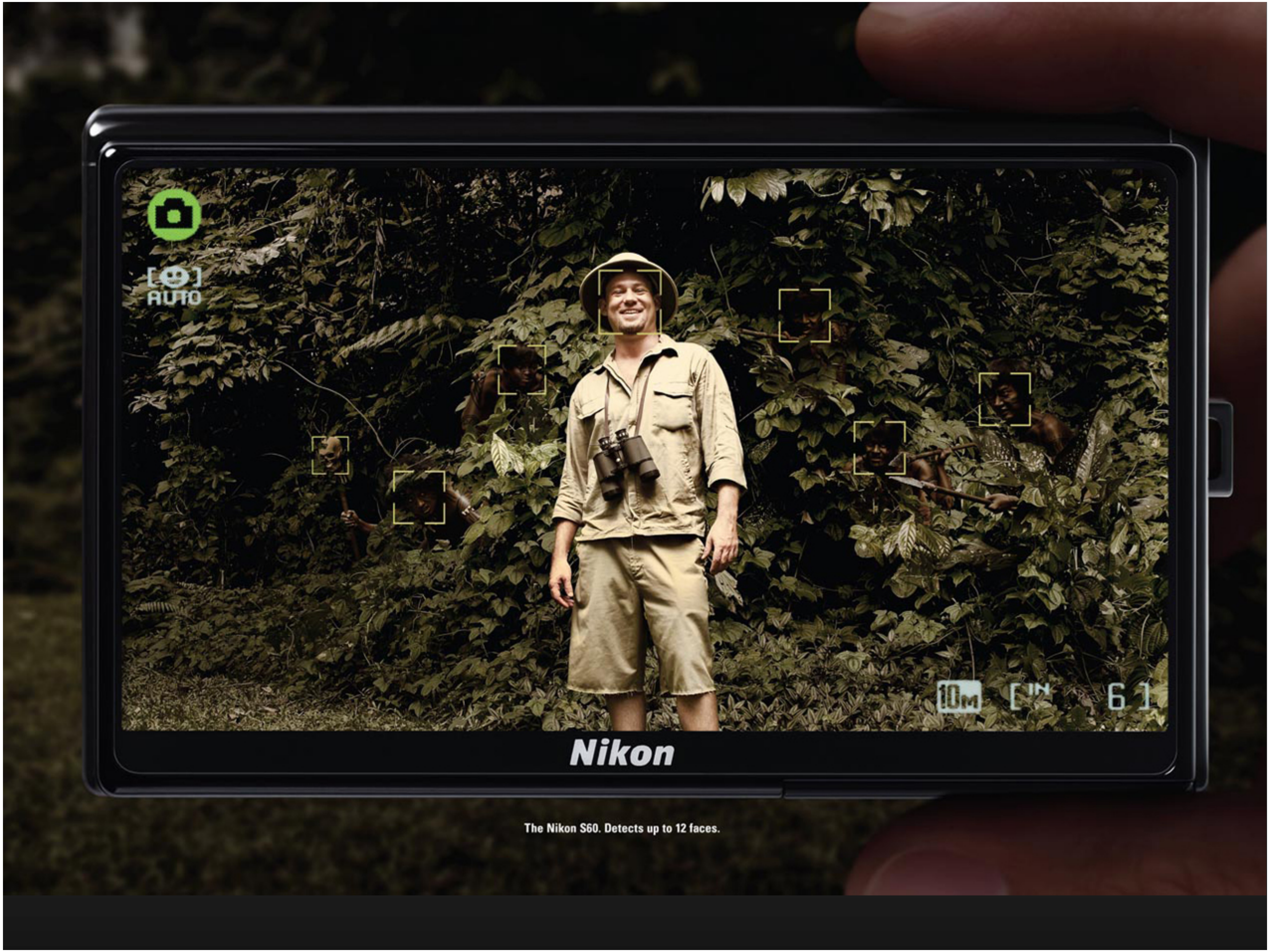
You will develop a simple "window-scanning" face detector and analyze its performance on real images.

The detector will be based on a linear Support Vector Machine classifier.

Training and test images will be provided.






AUTO
10M
IN
6
Nikon
The Nikon S60. Detects up to 12 faces.

Assignments

Assignments will be implemented in Matlab

We provide a comprehensive starting package including (1) detailed instructions, (2) usually some starting code and (3) datasets for each assignment.

Assignment 1: Scale-invariant blob detection

Jean Ponce, Ivan Laptev, Cordelia Schmid and Josef Sivic
(adapted from Svetlana Lazebnik, UNC)

Due date: October 27th 2009

The goal of the assignment is to implement a Laplacian blob detector as discussed in the lectures.

Algorithm outline:

Jean Ponce, Ivan Laptev, Cordelia Schmid and Josef Sivic
(adapted from Svetlana Lazebnik, UNC)

Due date: October 27th 2009

The goal of the assignment is to implement a Laplacian blob detector as discussed in the lectures.

Algorithm outline:

1. Build a Laplacian scale space, starting with some initial scale and going for n iterations:
   1. Generate a scale-normalized Laplacian of Gaussian filter at a given scale "sigma".
   1. Filter image with the scale-normalized Laplacian.
   2. Save square of Laplacian filter response for current level of scale space.
   3. Increase scale by a factor k.
2. Perform non-maximum suppression in scale space.
3. Display resulting circles at their characteristic scales.

Tips

- You have to choose the initial scale, the factor k by which the scale is multiplied each time, and the number of levels in the scale space. Reasonable value for the initial scale is sigma=2 pixels, using 10 to 15 levels for the scale pyramid. The multiplication factor should depend on the largest scale at which you want regions to be detected (try e.g. k=2^(0.25)).

- You can use the Matlab function `fspecial` for generating the scale-normalized Laplacian of Gaussian filter at a given scale "sigma":

```
filt_size =  2*ceil(3*sigma)+1; % filter size
LoG       =  sigma^2 * fspecial('log', filt_size, sigma);
```

- You can use the Matlab function `imfilter` to convolve the image with the filter, e.g.:

```
imFiltered = imfilter(im, LoG, 'same', 'replicate');
```

accommodate a different data type or a matrix of different dimensions. Here is how you would use it:

```
scale_space    = cell(n,1); %creates a cell array with n "slots"
scale_space{i} = my_matrix; % store a matrix at level i
```

- **On the non-maximum suppression in scale space.** The goal is to find pixels, which are local maxima in the scale-space. This amounts to finding pixels with the filter response (strictly) greater than its 26 (3x3x3 neighbourhood) scale-space neighbours, considering also the adjacent scales as illustrated in figure 2 of David Lowe's paper.

- To perform non-maximum suppression in scale space, you may first want to do non-maximum suppression in each 2D slice separately. For this, you may find functions `nlfilter`, `colfilt` or `ordfilt2` useful. To extract the final nonzero values (corresponding to detected regions), you may want to use the `find` function.

- You also have to set a threshold on the squared Laplacian response above which to report region detections. A reasonable value is 0.001, but you should play around with different values and choose one you like best.

- To display the detected regions as circles, you can use this function (or feel free to write your own). Don't forget that there is a multiplication factor that relates the scale at which a region is detected to the radius of the circle that most closely "approximates" the region.

- If you decide to experiment with different implementation choices for building the scale space and nonmaximum suppression, you may want to compare these choices according to their computational efficiency. To time different routines, use `tic` and `toc` commands.

- Apply the detector to greyscale versions of colour images. In Matlab, you can convert a color image to greyscale using

```
Im_gray = mean(Im_rgb,3);
```

  Make sure the image values are scaled between 0 and 1.

- If you are new to MATLAB, have a look at the very useful tutorials (from Martial Hebert at CMU). They will give you overview of basic necessary functions: basic operations, programming, working with images.

- You may also want check this code for an example of a Harris interest point detector: Sample Harris detector code.

Test images

Here are four images to test your code. Also run your code on two images of your own choice.

What to hand in

· You should implement the Laplacian blob detector as described in the algorithm outline above.

· Motivated students can also implement the efficient difference of Gaussian approximation described in section 3 of David Lowe's paper. However, this is not required and will not be marked.

You should prepare a (very brief) report including the following:

· A figure showing the 2D Laplacian of Gaussian filter (use Matlab functions `surf(LoG)` or `mesh(LoG)` to visualize the filter).

· Short answers to the following two questions: (i) What does it mean when a filter is separable? (ii) Is the Laplacian of Gaussian filter separable?

· For image "sunflowers.jpg" show the squared Laplacian response for three different scales (sigma=2, 4 and 8 pixels).

· Show a graph of the Laplacian response across scales, i.e. the scale "sigma" on the x-axis and the squared Laplacian filter response on the y-axis, for two image points corresponding to: (i) the large sunflower in the foreground (image coordinates from top-left corner [row=327, column=128] and (ii) the small sunflower in the background (image coordinates [row=62, column=130]). On which scale is the strongest response for the two points?

· Show the output of your detector (i.e. detected regions shown as circles of different sizes overlaid over the image) on all test images.

Instructions for formatting and handing-in assignments:

· At the top of the first page of your report include (i) your name, (ii) date, (iii) the assignment number and (iv) the assignment title.

· The report should be a single pdf file and should be named using the following format: A#_lastname_firstname.pdf, where you replace # with the assignment number and "firstname" and "lastname" with your name, e.g. **A1_Sivic_Josef.pdf**.

· Zip your code and any additional files (e.g. additional images) into a single zip file using the same naming convention as above, e.g. **A1_Sivic_Josef.zip**. We do not intend to run your code, but we may look at it and try to run it if your results look wrong.

· Short answers to the following two questions: (i) What does it mean when a filter is separable? (ii) Is the Laplacian of Gaussian filter separable?

· For image "sunflowers.jpg" show the squared Laplacian response for three different scales (sigma=2, 4 and 8 pixels).

· Show a graph of the Laplacian response across scales, i.e. the scale "sigma" on the x-axis and the squared Laplacian filter response on the y-axis, for two image points corresponding to: (i) the large sunflower in the foreground (image coordinates from top-left corner [row=327, column=128] and (ii) the small sunflower in the background (image coordinates [row=62, column=130]). On which scale is the strongest response for the two points?

· Show the output of your detector (i.e. detected regions shown as circles of different sizes overlaid over the image) on all test images.

Instructions for formatting and handing-in assignments:

· At the top of the first page of your report include (i) your name, (ii) date, (iii) the assignment number and (iv) the assignment title.

· The report should be a single pdf file and should be named using the following format: A#_lastname_firstname.pdf, where you replace # with the assignment number and "firstname" and "lastname" with your name, e.g. **A1_Sivic_Josef.pdf**.

· Zip your code and any additional files (e.g. additional images) into a single zip file using the same naming convention as above, e.g. **A1_Sivic_Josef.zip**. We do not intend to run your code, but we may look at it and try to run it if your results look wrong.

Send your report, code, and your two additional test images to **Josef Sivic <Josef.Sivic@ens.fr>**.

Helpful resources

· Useful Matlab tutorials (from Martial Hebert at CMU): basic operations, programming, working with images.

· Sample Harris detector code.

· Nice Slides by Svetlana Lazebnik on feature detection describing also scale invariant blob detection (slides 32—49).

· Blob detection 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-

Final projects

• The final project will represent 30% of the grade.

• The topics will be announced in the class and on the course page.

• Talk to us if you want to define your own project (before you start working on the project).

•Joint projects with other courses are possible, but please talk to instructors of both courses (before you start working on the project).

• You are encouraged to work alone but you can also form groups of two or three people. Each group will submit a single final report. Reports will be in a standard conference format submission and max 3 pages long.

• Hand out date: Nov 2.

• Project presentations (~10mins per group) and reports due on: Dec 14.

Final projects

Two suggested topics building on assignments 3 and 4:

1. Image classification of 100 object classes.

Final projects

Two suggested topics building on assignments 3 and 4:

2. Object detection and localization

Any questions?