# Image Segmentation Using Normalized Cuts

A graph theoretical approach to image segmentation lab section

Manuel Pagliai, Lorenzo Sarti

May 22, 2013

- The code is organized in different blocks, according to different techniques.
- The main file consists in a switch statement operating over different algorithms

- The main function can be called as follow:

```
main(imageName, SI, SX, radius, cutValue,
 methodName, HSVConvertion, nOfClusters)
```

  - imageName is the name of the image in the folder Images without extension.
  - SI is the value of sigma controlling the sensitivity to colour
  - SX is the value of sigma controlling the proximity
  - radius is the threshold beyond that the proximity is zero
  - cutValue is the threshold value for the cut
  - methodName is the algorithm chosen
  - YUVConvertion is a boolean for converting the image in YUV space
  - nOfClusters is the number of clusters desired

- Your assignment for this lab session is to run the algorithm with different values of cutValue.
- Notice the different results obtained by changing this parameter.

- You can test your program, by running

```
main('img11', 0.01, 1.7, 5, x, 'nCut', false);
```

The resulting partitions are saved inside the folder `./Results/nCut/img11`

- What you should have noticed by changing the parameters:

1. Increasing the value of SI, the colour weights more in the similarity measure.
2. With a bigger value of SI, it is more likely to get many small regions of uniform colour.
3. Decreasing the value of SI, i.e. decreasing the value of the weight in the similarity measure, it is more likely to get less regions but with colour that might be non uniform.
4. Setting the radius too small discards too many neighbouring pixels, while setting the radius too big, other than increasing the complexity of the algorithm, also accounts for pixels too distance to the one considered.