Twenty-one matlab files are used

- sceneCompletion_starter.m
- make.m   ---- **from Miki Rubinstein**
- applyMask.m
- gistCompute.m
- gistComputeInput.m
- imresizecrop.m ---- **from Oliva**
- LMgist.m   ---- **from Oliva**
- colorDiff.m
- folderCompute.m
- sceneMatching.m
- computeGistSSD.m
- computeLocalContext.m
- localMatching.m
- computeContextSSD.m
- computeTextureSSD.m
- graphCut.m
- computeAdjMatrix.m
- computeGradientMagnitude.m
- maxflow.m ---- **from Miki Rubinstein**
- maxflow.mex.cpp -----   ---- **from Miki Rubinstein**
- maxflowmex.mex.maci64 ---- ---- **from Miki Rubinstein**
- edge4connected.m   ---- **from Miki Rubinstein**
- poissonBlend.m

The folder maxflow-v3.0 is from the Max-flow/min-cut code to compute the best graph cut between a source and target input. This code was **written by Vladimir Kolmogorov.**

There is also three .mat files:

- gistImgDB.mat
- folderDBInfo.mat
- colorDiffDB.mat

Those three files are the preprocessing steps needed on all the images (61301 exactly) in the database (folder db_images). It takes a long time to build those files. The user won't need to do it unless if the database changes.

To run the code, you need to open and run the sceneCompletion_starter.m file. The three .mat files are loaded.

The code needs a picture image and a mask of the same size to work. For that, I took test images and masks from the website of the authors:
http://graphics.cs.cmu.edu/projects/scene-completion/

Six tests are available in the folder images/img_mask. To pick one, the user only need to select at line 10, the number of his choice.

ex: testX = test4

The implementation is in seven steps:

- Call make.m function to be able to use the mex cpp file for the graph cut
- Application of the mask on the input = masked image (new input)
- Computation of the gist on the new input
- Scene matching process which gives back 10 best matches
- Computation of the local context
- Local matching for each scene match (bottleneck of the code - runs in parallel if available, but still very slow), which gives back a best patch
- Graph cut for each patch which gives back a new mask
- Poisson blending for each patch using the new mask, which gives back the final result

All the final images produced are saved in: images/outputs/textX

The masked images, matching scenes, local context, best patches, cut masks, cut patches are all saved under the images folder.

For each test, there two folders for each categories:

- textX_1 : for the test where the color differences were not scaled
- textX : for the test where the color differences were not scaled

The user should be able to see through messages in the console the current step being computed.