UNIVERSITY OF GHENT

DEPARTMANT OF TELECOMMUNICATION AND INFORMATION PROCESSING

SUMMER INTERNSHIP

COURSE: COMPUTER GRAPHICS

Assignment: Optimizing and making code for context-aware patch-based image inpainting more user friendly

|  |  |  |
| --- | --- | --- |
| Student |  | Mentor |
| Mihailo Drljača |  | Ph.D. Aleksandra Pižurica |

1. INTODUCTION

Image inpainting is an image processing task of filling in the missing region in an image in a visual plausible way. Applications are wide such as image restauration, photo editing, image coding and transmission etc. Context-aware patch-based image inpainting using Markov random field modelling is a novel method developed by Ph.D. Aleksandra Pižurica and Ph.D. Tijana Ružić. The main idea of this method is to employ contextual (textural) descriptors to guide and improve the inpainting process. Two most important contributions are:

* Context-aware patch selection strategy
* Interface method for global MRF-based inpainting

Context-aware patch selection strategy is not limited to global inpainting. It aims at improving and accelerating the search for candidate patches in patch-based methods in general.

In this particular realization of the method normalized texton histograms computed from Gabor filter responses are used as contextual descriptors. In general any other contextual descriptors can be used.

There are two strategies for dividing an image into regions based on the context:

* Division into fixed-size square non-overlapping blocks
* Division into blocks of adaptive size

In this particular realization of the inpaintig method division into blocks of adaptive size is used.

Interface method for global MRF-based inpainting uses novel optimization approach that makes it suitable in case of large number of labels.

Code is written in Matlab. Original version consists of 144 functions written in Matlab. There is many functions that are similar to each other because the code was written during the research. These similar functions are differing vary little because they were written to be optimal for a specific use.

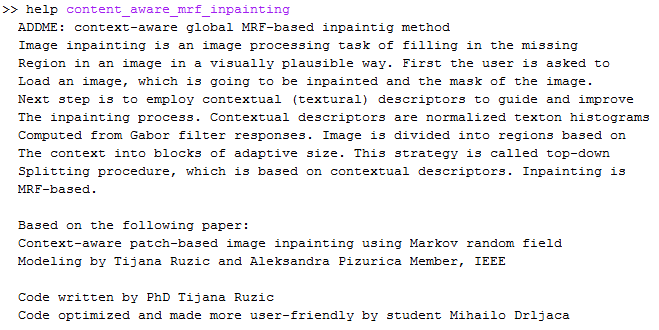
Task was to select and gather functions that are essential to the method so that optimized version of the program can be delivered to the various users. Besides that optimized version of the code is commented so that further revisions can be easily made. Parts of the code that are prompting user to interact with program makes it more user friendly. Every function that is called in the main function is assigned with an explanation that can be read by using help function in Matlab. Script is called **content\_aware\_mrf\_inpainting** and it also contains an explenation. Version of script file that is realized as a function called **content\_aware\_mrf\_inpainting\_func**.

1. SCRIPT FILE

Code is realized as script that is calling all functions. Structure of program is given below:

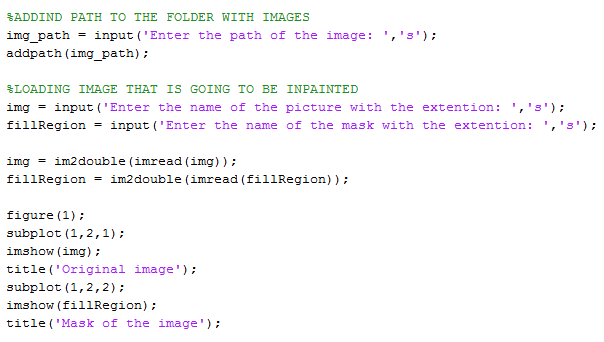
1. **content\_aware\_mrf\_inpaintings**
   1. **textonGenerateContrastNorm**
      1. **createGabor**
   2. **decomposition\_adaptive\_blocksize**
      1. **splitVertically**
      2. **splitHorizontally**
      3. **chiTestTexton**
   3. **textonHistBlock**
   4. **gist\_matches\_all\_adaptive\_blocksize\_weighted**
      1. **find\_neighbours\_adaptive**
   5. **find\_label\_pos\_per\_adaptive\_blocksize**
   6. **comp\_data\_gist\_ver4\_adaptive\_scaled\_weighted**
      1. **compdata.c**
   7. **label\_extraction\_gist\_weighted**
      1. **find\_first**
      2. **label\_pruning\_art\_fast**
      3. **find\_neighbours**
      4. **calculate\_weighted\_dif\_fast1**
         1. **overlap\_region**
         2. **calculatediffhelper.c**
      5. **upade\_priority\_scaled**
   8. **compute\_pot\_art**
      1. **find\_neighbours**
      2. **comp\_matrix\_pot\_art**
         1. **overlap\_region**
         2. **calculatepotentionalhelper.c**
   9. **compute\_cost\_art**
      1. **getpatch**
      2. **ssd3**
   10. **NCMP**
       1. **find\_neighbours**
       2. **normalise\_new\_inf**
   11. **output\_art\_mincut**
       1. **binomialFilter**
       2. **getpatch**
       3. **dpmain**
          1. **dp1**
          2. **dp2**
          3. **randmin**
       4. **dp**
       5. **rconv2**
   12. **show\_order**

Script file is called **content\_aware\_mrf\_inpainting**. After defining the name of the script, explenation of the script file is given. It can be called with **help** function. (0) After the explenation all used variables are described.

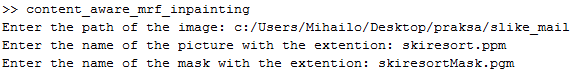


1. Matlab command window

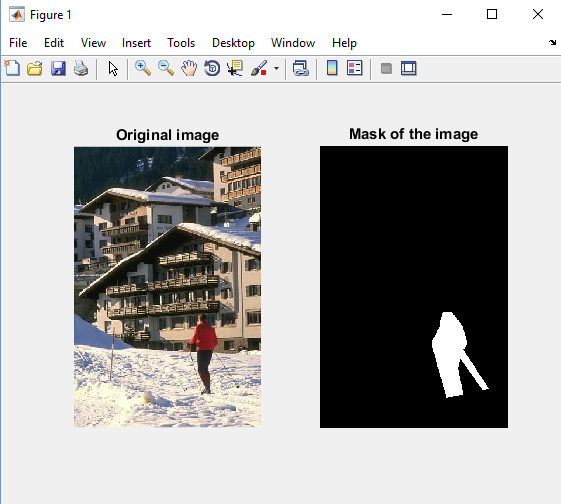
Before creating contextual descriptors image and mask have to be loaded. User is prompt to insert path to the mask and the image. Then user has to insert its name with an extension. Matlab function **input** is being used to prompt the user. Functions **imread** and **im2double** are used to load the image and mask. After that the mask and the image are plotted. (Picture 2.4)



1. Part of the code that is responsible for loading image and mask

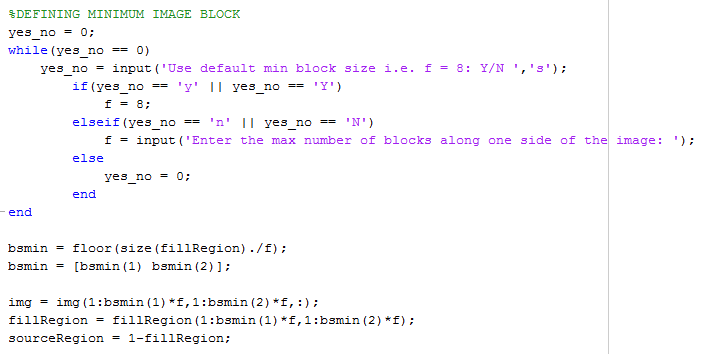


1. User interface – Matlab command window

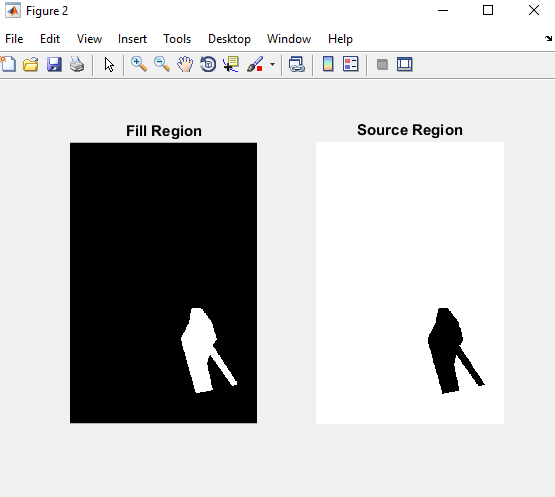


1. Plotted image and mask

Next step is to define minimum block size of the image. Default parameter is *f* = 8. User is prompt to decide weather he is going to use default value or not. Length and width are divided by *f* and stored into *bsmin* parameter. Source and fill region are plotted. (Picture 2.6)



1. Part of the code that is responsible for defining minimum block size

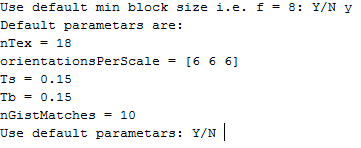


1. Plotted source and fill region

For context-aware patch algorithm following parameters have to be loaded.

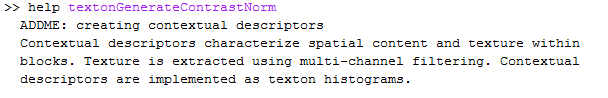
* nTex – number od textons
* orientationsPerScale - number of orientations over 3 scales
* Ts - block similarity threshold used while decomposing image
* Tb - block similarity threshold while context aware patch selection is being performed
* nGistMatches - max number of blocks from where the patches are considerd

Once again user can choose between default and personalized values.



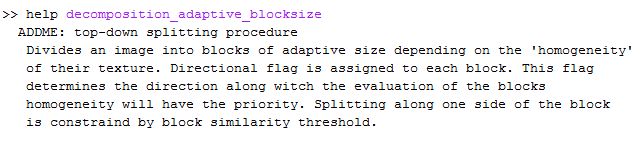
1. Parameters displayed in Matlab command widow

After loading of the parameters contextual descriptors are being calculated. Function that is responsible for this part of the method is **textonGenerateContrastNorm**. Further explenation of the function is given in Matlab help selection. (Picture 2.8)



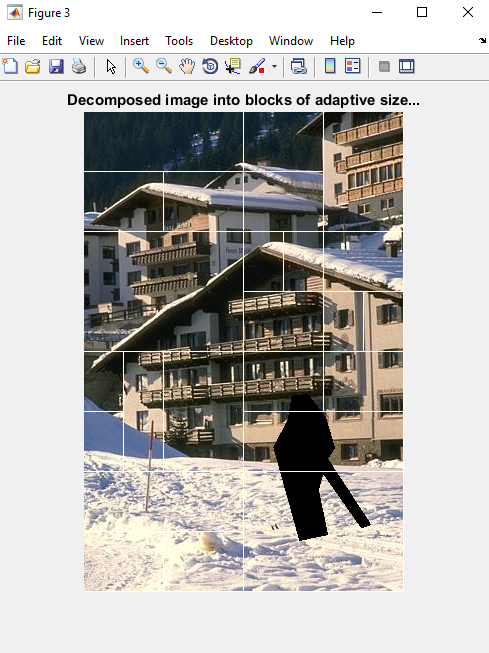
1. Matlab command window

Next step is to divide image into blocks of adaptive size. Function that is responsible for division is **decomposition\_adaptive\_blocksize**. (Picture 2.9)



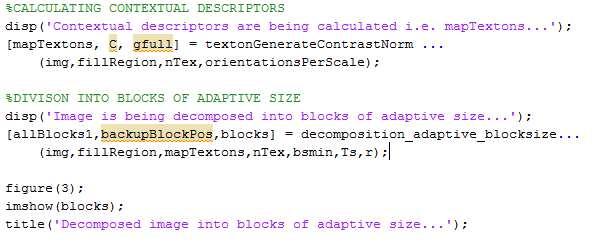
1. Matlab command window

After decomposition processed image is displayed. (Picture 2.10)



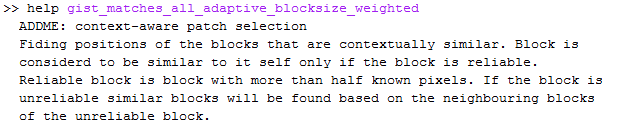
1. Decomposed image

Part of the code that is responsible for previously described process is shown below.

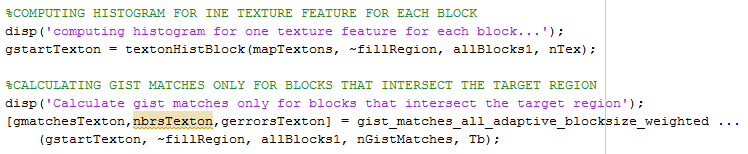


1. Part of the code responsible for creating contextual descriptors and dividing image into blocks of adaptive size

To perform inpainting it is obligatory to find positions of the contextually similar blocks. Functions that are employed in this section are **textonHistBlock** and **gist\_matches\_all\_adaptive\_blocksize\_weighted**.



1. Matlab command window

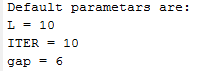


1. Part of the code responsible for finding positions of the blocks that are contextually similar

For MRF- based inpainting following parameters have to be loaded.

* L – number of labels kept after pruning
* ITER- number of iterations
* gap - determines the patch size as 2gap+1

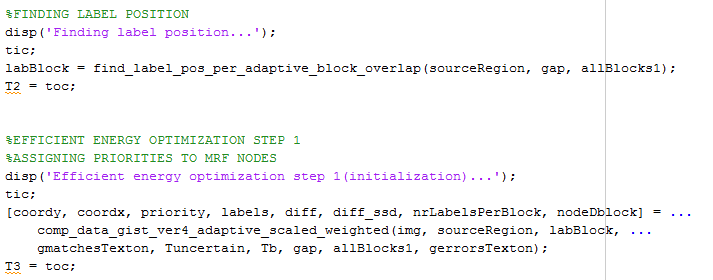
Once again user can choose between default and personalized values.



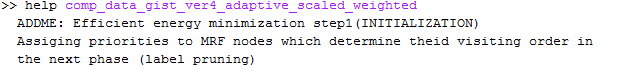
1. Parameters displayed in Matlab command widow

Part of the code that is responsible for patch-based inpainting process uses memory efficient optimization method. This method consists of three steps. Initialization, label pruning and inference.

Function responsible for initialization is **comp\_data\_gist\_ver4\_adaptive\_scaled\_weighted**. Before executing initialization it is necessary to find all labels positions with function called **find\_label\_pos\_per\_adaptive\_block\_overlap.**

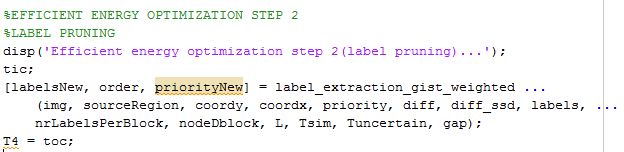


1. Part of the code responsible for initialization of energy efficient optimization

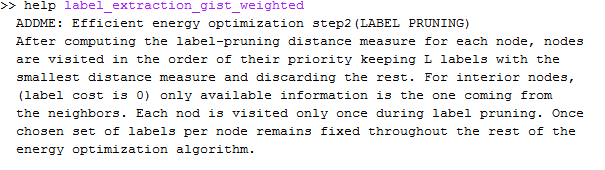


1. Matlab commad prompt window

Second step label pruning is realized with **label\_extraction\_gist\_weighted.**

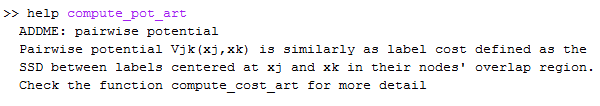


1. Part of the code responsible for label pruning

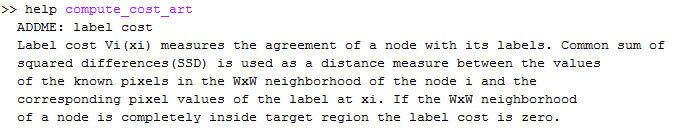


1. Matlab command prompt

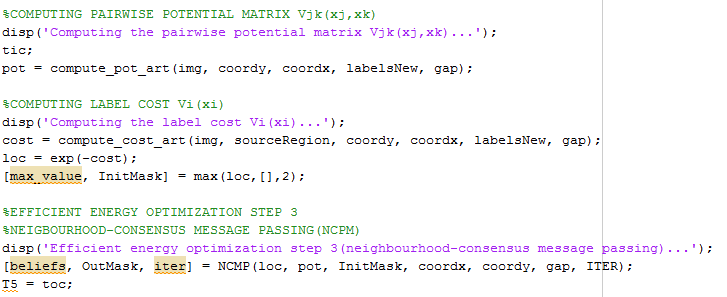
Third and last step of energy optimization is neighbor consensus message passing. Function is call **NCMP**. Before calling NCMP label cost and pairwise potential have to be calculated. Function that calculates label cost is called **compute\_cost\_art.** Inside **compute\_cost\_art** there is **ssd3** function that calculates common sum of squared differences. Pairwise potential is calculated with **compute\_pot\_art** function.



1. Matlab commad prompt

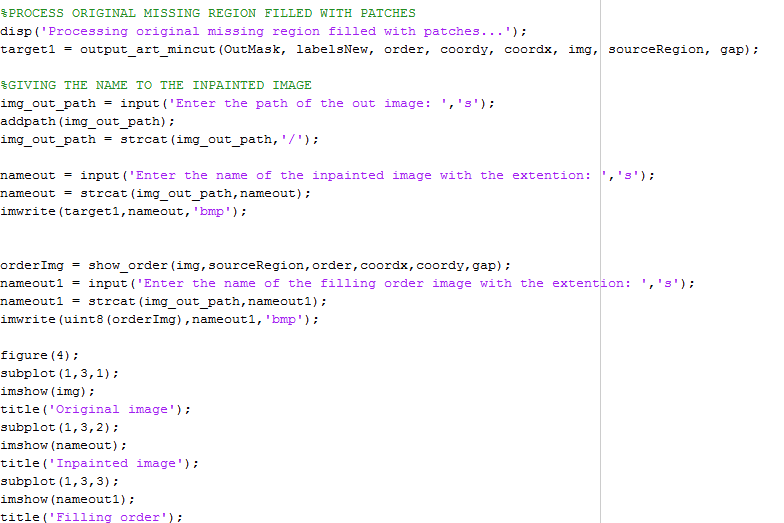


1. Matlab commad prompt

****

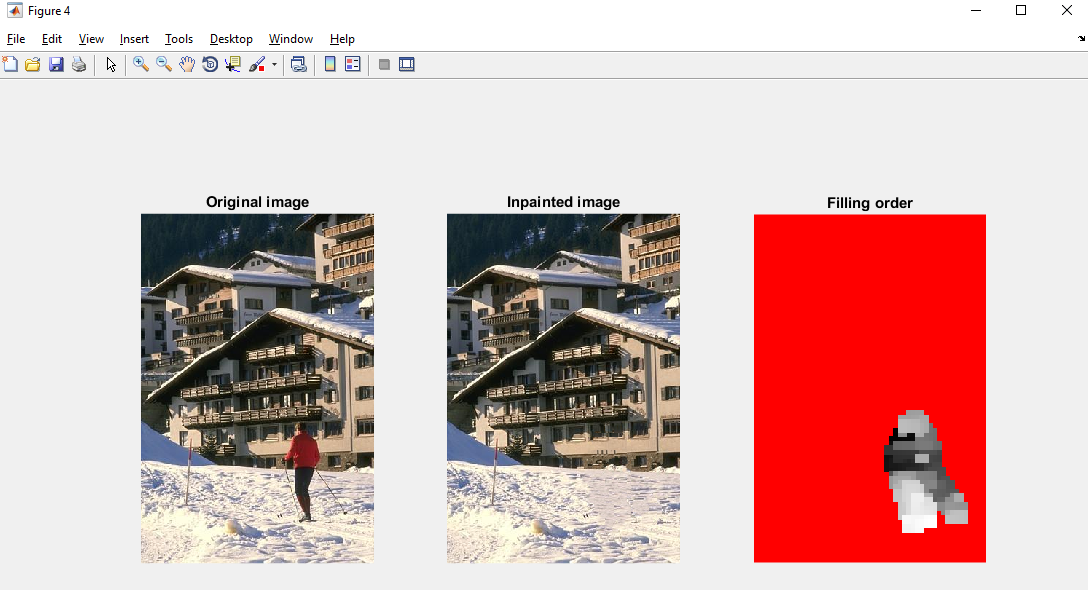
1. Part of the code responsible for final step of energy efficient optimization

When the processing of missing region filled with patches of the original image is done, function **output\_art\_mincut**, name and path of the inpaited image and image representing filling order have to be given.



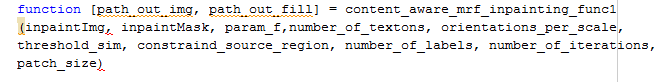
1. Part of the code responsible for storing inpainted image

After finishing the inpaintig process original image, ipainted image and filling order image are displayed.

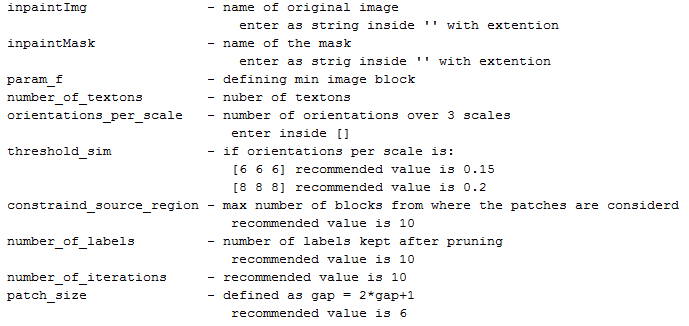


1. Original image, inpainted image and filling order image
2. FUNCTION

In addition to script file function called **content\_aware\_mrf\_inpainting\_func** is realized. All of the parametars that were prompted from the user in scrip file are now requested as parametars of the function. Function returns path of the inpainted image and filling order image.



1. Function **content\_aware\_mrf\_inpainting\_func**

****

1. Parametars of the function

Based on the following paper: T. Ružic and A. Pižurica, “Context-aware patch-based image inpainting using Markov random field modeling,” IEEE Transactions on Image Processing, vol. 24, no. 1, pp. 444-456, Jan 2015