TOPICS IN VIRTUAL REALITY:
MATHEMATICAL METHODS FOR VISUAL COMPUTING

EXERCISE 2 - MLS FOR IMAGE MANIPULATION

Handout date: 27 September 2017
Submission deadline: 16 October, 17:59
Demo date: TBA

General Rules

Plagiarism note. Copying code (either from other students or from external sources) is strictly prohibited! We will be using automatic anti-plagiarism tools, and any violation of this rule will lead to expulsion from the class.

Late submissions will not be accepted, except in case of serious illness or emergency. In that case please notify the teaching assistants.

Software. All exercises of this course use the MATLAB programming language. The MATLAB distribution is available from `http://kftp.kaist.ac.kr/` for KAIST students. See our MATLAB tutorial slides for hints or specific functions that could be useful for your implementation.

What to hand in. Submit a .zip file of your solution on KLMS. The *zip file* must be called "MathMethods17-Ex*-firstname-familyname.zip" (replace * with the assignment number, e.g. MathMethods17-Ex01-John-Smith.zip). The .zip file MUST contain *a single folder* with the following data inside:

- A folder named "code" containing your MATLAB code
- A README file (in pdf format) containing a description of what you have implemented and instructions for running it, as well as explanations/comments on your results.
- Screenshots of all your results with associated descriptions in the README file.

Grading. The homeworks count for 70% of your total grade. Your homework submission will be graded according to the quality of the images/results produced by your program, the conformance of your program to the expected behaviour of the assignment, and your understanding of the underlying techniques used in the assignment. The submitted code must produce exactly the same images included in your submission.

To ensure fairness of your grade, you will be asked to briefly present your work to the teaching assistants. Each student will have about 5 minutes to demo their submission and explain in some detail what has been implemented, report potential problems and how they tried to go about solving them, and point the teaching assistants to the code locations where the various key points of the assignments have been implemented. See above for the scheduled demo date for this particular assignment.

Goal of this exercise

In this exercise you will apply what you learned about the Moving Least Squares (MLS) for image manipulation. You will read and implement parts of recent research papers that used MLS.

1. EXERCISE PART 1: Image Deformation Using Moving Least Squares

This task is to write a matlab program to deform images based on moving least squares as presented in the paper "Image deformation using moving least squares" by Schaefer et al. [06] (provided with the exercise). In this paper, the authors propose a method to smoothly deform images using different kinds of transformations and allow the user to manipulate sets of points and segments to specify the deformations. For this exercise you will only focus on image deformations using sets of points.

Write a matlab program which takes as input an image and performs the deformation after the user selects some input and output control points. The algorithm should follow the description from the paper. The three kinds of transformation, i.e. affine, similarity and rigid transformations should be supported by your program. Please note that we do not ask for an interactive application so that you do not need to worry about making the code fast (precomputation of some terms, approximation of the image by a grid), but you can do it if you wish.

Some pointers to help with your implementation:

- Read section 2 of the paper to understand the types of transformations you need to apply.
- Implement backwards warping to avoid artifacts.
- You can use the "meshgrid" and "griddata" functions to implement the warping process.
- See Figure 1 for an example of what your results should look like.

Required output of this section:

- Code that deforms images according to user input using all three methods described above (affine, similarity and rigid).
- Representative image results of the gingerbread man (provided with the exercise) and at least one more image of your choice.

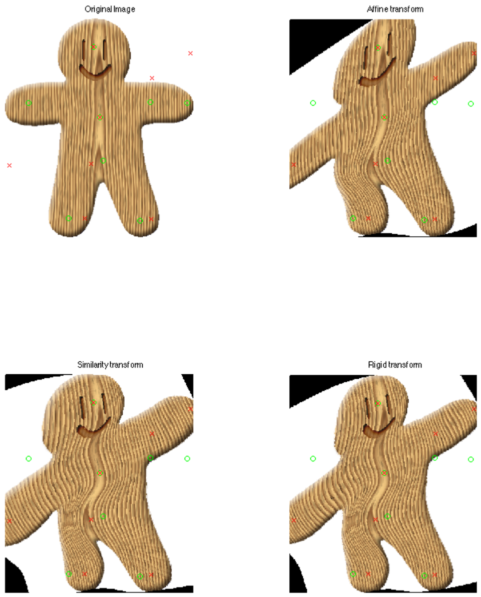

Figure 1. A possible result for exercise part 1.