

# Assignment A1: Top Photograph

Jens E. Wilhjelm

Biomedical instrumentation, Department of Health Technology,  
Ørsteds plads, Building 349  
Technical University of Denmark  
2800 Kgs. Lyngby

(Ver. 1.7 5/9/22) © 20012 - 2022 by JW

## 1 What to do

This assignment is relatively simple and well explained. The next one will be described in a less thorough manner. Therefore, it is important that all assignments are done by each student, in order to be able to scope with the most complicated ones.

### *You will need:*

- The phantom photograph which will be available at the course homepage. This data is typically in TIFF-format.
- Measures of the dimension of the phantom which can be found in the Practical guide.

### *Procedure:*

- First, adjust the rotation angle and cut out the phantom surface itself so that all of the acrylic frame can be seen (not too much, neither too little). There are two ways; use one of them:
  - In a small script, read the TIFF data into MATLAB by using `sis_import` and remember to use the correct representation of the photo by distinguishing between “rgb” and “indexed” as elaborated on in the Appendix of the SIS guide. When this step is finished, you have your data in SIS format. (If the photo happens to be in black-and-white, you do not have to worry about “rgb” and “indexed” since the photo will then just be an intensity image)
  - Use a general image processing program. Save the TIFF data. Read the TIFF data into MATLAB with `sis_import`. Check that your image does not have 4 colour layers; if it has, then you should remove the forth, leaving three layers for subsequent processing. When this step is finished, you have your data in SIS format.
- Create or verify the metric axes and adjust the zero of these two axes to a convenient location on the phantom surface. For many, it would be logical to put the zero point in the middle of the phantom surface.
- Plot the data and verify that the metric axes are correct.
- For this plot, make a small MATLAB *function* (`plot_surface_landmarks`) which draws parallel lines over each set of fiducial markers, much the same way as in Figure 4 in the Practical guide (and if you are enthusiastic, also the binary markers, the filling hole and maybe some of the screws). You will need the created function many times later during the course.

### 1.1 SIS functions that you may want to use:

`sis_welcome` (Run this function as the first thing!)  
`sis_import` (Reading TIFF formats)  
`sis_zoom` (Zoom on a given region in an (3D) image)

```

sis_rotate (Small rotation of images)
sis_rot90 (Rotates images exactly 90 degrees)
sis_rgb_to_icm (Converts rgb images to index images with colormap)
sis_view (Visualization of multiple images in SIS format)
sis_extract_axis_info (Extracts information from a given SIS data structure)

```

## 1.2 About the script and plot\_surface\_landmarks

The main script is short and easy to read and handle since it operates at an overall level. In contrast, the function `plot_surface_landmarks` operates on a detailed level and contains all the graphics that is drawn on the phantom surface. This is a more complicated program so it is nice to have this "put away" when you know that it works.

### 1.2.1 Example of some of the components in the script:

```

% Load photo:
Data = sis_import( 'phantom_3.tif' );

% Possible cut and rotation using sis_zoom and sis_rotate ...

% Adjust zero-point by subtracting values from the axis:
Data.Axes(1).Axis = Data.Axes(1).Axis - 80; % adjust the 80 for your case

% Visualization:
sis_view( Data);

Set hold on and plot all kinds of graphics on the surface:
hold on; plot_surface_landmarks;

% Make an eps-file for your report (check the eps-file with tje Acrobat Viewer):
printfig([cd '\'], 'top_photo');
% eps gives you both vector graphics and bitmaps in one and the same file

```

### 1.2.2 Example of components of plot\_surface\_landmarks

```

function plot_surface_landmarks;
% Plots landmarks on the phantom surface used in the 22481 course
% It assume that the zero point is in the middle of the phantom surface and that
the axis appearing in the MATLAB figure window are in mm

% Plot lines through fiducial markers (put in your own more precise numbers):
Y = about -30;
for y = -Y : 5 : Y
    plot( [-40 40], [Y Y]);
end;

% Plot inner and outer box with rectangle( bla bla )
% Plot circles where appropriate with circle( bla )

```

## 2 Report requirements

Make a small (approximately two) page report with the plot of the phantom surface with all the lines, circles, boxes etc. This small report should also contain:

- Introduction and purpose (the latter both with respect to specific goal and overall goal).
- Be sure to state how the photograph was obtained and how you processed the data.
- Give an explanation of why the zero-point is located as it is.
- Put the MATLAB code in an appendix with comments for each line of code when relevant.

### 2.0.1 Rubric for peer-review

When doing peer-review in the system, the rubric in Table 1 will be used. Consider using this as a check list before hand-in of the assignment.

**Table 1:** Rubric for Assignment 1.

	☺	☺☹	☹	☹☹	☹
Description	Very good. Probably not possible to make better	Quite good and satisfactory	Just acceptable	Improvements are needed	Totally wrong or missing
<b>Purpose:</b> Is the purpose of the assignment stated?					
<b>Method:</b> Is it stated how the image was obtained?					
<b>Processing of image:</b> Does the report contain a clear explanation of how the data was processed?					
<b>Axes:</b> Are the metric axes correct with respect to numbers and units?					
<b>Fiducial markers:</b> Does the top photograph contain lines marking the fiducial markers?					
<b>Additional fiducial markers:</b> Does the top photograph contain any other practical markings (e.g. inner and outer delineation of acrylic box, binary markers, filling hole, screws, other holes)?					
<b>Geometry:</b> Does the phantom surface appear rectangular and with sides parallel to the axes?					
<b>Zero point:</b> Is it stated why the zero-point was placed as it is?					
<b>MATLAB code:</b> Is the MATLAB code commented?					