

## Assignment 3: Multi-Modal Image Registration

In this assignment you will work with two images dataset (CT and MRI) of the same anatomy (knee). The goal is to use data from these two image datasets simultaneously and discover the advantages and difficulties in working with multi-modal images.

Both datasets were obtained by imaging a right knee specimen.

### *CT Dataset*

The CT dataset contains 299 slices (in directory "CT\_Bin"). Each slice has a dimension of **190 x 190** pixels, with a **pixel type 'int16'**. Each CT slice is stored in MATLAB 2d matrix format in the file 'knee\_ct\_nr.bin'. Files are ordered with respect to their position in the volume (feet first). File format is binary. To read the image data you can use fread or similar MATLAB functions.

Each pixel represents a physical size of **1mm x 1mm**, and the slice distance is **1mm** (could I have made it any easier for you ☺ ?)

The origin for the dataset is located at point  $[-83.3998, -209.4639, -803.00]$ , which means the pixel value at the position (1,1) (Matlab format) of the first slice in the dataset represents the location  $(-83.3998 \text{ mm}, -209.4639 \text{ mm}, -803.00 \text{ mm})$  in the CT coordinate system. x,y, and z direction of the CT coordinate system are given in figure 1.

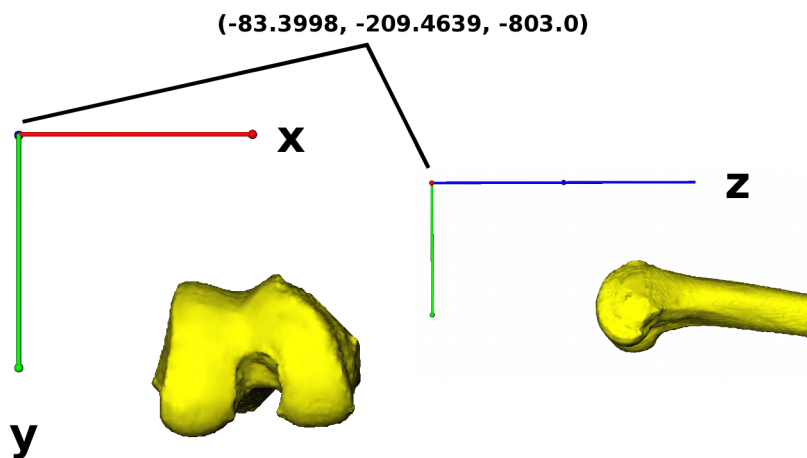


Figure 1: Orientation of axes in CT coordinate system

## ***MRI Dataset***

The MRI dataset contains 154 slices (in directory “MRI\_Bin”). Each image has the dimension **190 x 190** pixels, with a **pixeltype ‘uint16’**. Each MRI slice is stored in MATLAB 2d matrix format in the file ‘knee\_mri\_nr.bin’. Files are ordered with respect to their position in the volume (feet first). File format is binary. To read the image data you can use fread or similar MATLAB functions.

Each pixel represents a physical size of **1mm x 1mm**, and the slice distance is **1mm** (what a coincident ☺).

The origin for the dataset is located at point  $[-89.7749329, -48.4298096, -167.5021362]$ , which means the pixel value at the position (1,1) (Matlab format) of the first slice in the dataset represents the position  $(-89.7749329\text{mm}, -48.4298096\text{mm}, -167.5021362)$  in the MRI coordinate system. x, y, and z direction of the MRI coordinate system are given in figure 2.

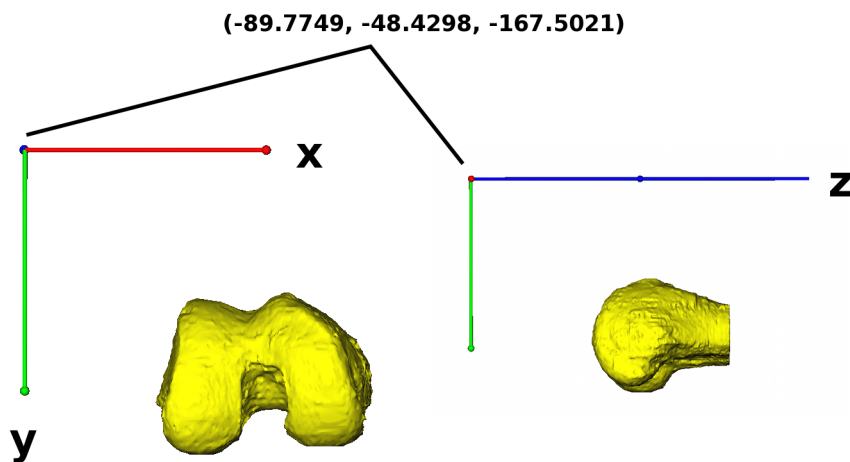
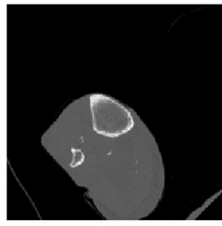


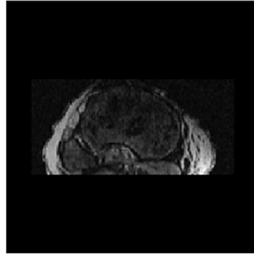
Figure 2: Orientation of axes in MRI coordinate system

### ***Please note:***

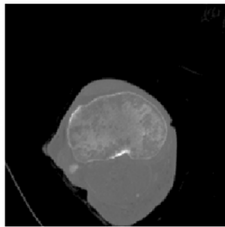
For easier calculation between the two image datasets, I cropped both to the same image dimension (190x190). However, the cropped volume with respect to the knee anatomy is not identical, which means the same pixel position in a CT and MRI slice does not represent the same spot in the anatomy (see figure 3).



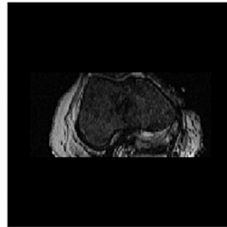
CT Slice nr 50



MRI slice nr 50



CT slice nr 115



MRI slice nr 115

Figure 3: Example slices for MRI and CT dataset

### ***Segmentation Masks:***

In addition to the image datasets, you also receive a segmentation mask for each dataset (MRI\_mask.txt and CT\_mask.txt). Both masks are saved in the following format:

Each file lists points (in image device coordinate system, in mm), which are part of the segmented object and its voxel value. Format for each point is:

x, y, z, d

with x,y, and z describing the position of the pixel (in mm) and d = voxel value.

### ***Anatomical Landmarks:***

For each dataset, coordinates for 5 anatomical landmarks are defined (see figure 4). All coordinates are given in imaging device coordinate system (in mm).

CT Dataset:

Lateral Epicondyle: -26.14317 -93.6106 -643.297

Medial Epicondyle: 60.6069 -112.42 -650.373

Top of the patellar gliding notch: 9.08479 -133.96 -638.603

Medical posterior condyle: 42.0235 -75.6846 -656.996

Lateral posterior condyle: -6.67128 -71.681 -649.853

MRI Dataset:

Lateral Epicondyle: -34.5029 56.4774 -59.9131

Medial Epicondyle: 52.4359 38.6747 -67.2723

Top of the patellar gliding notch: 1.56049 17.2319 -52.8177

Medical posterior condyle: 33.0425 74.4548 -76.3488

Lateral posterior condyle: -15.5747 78.1617 -68.5479

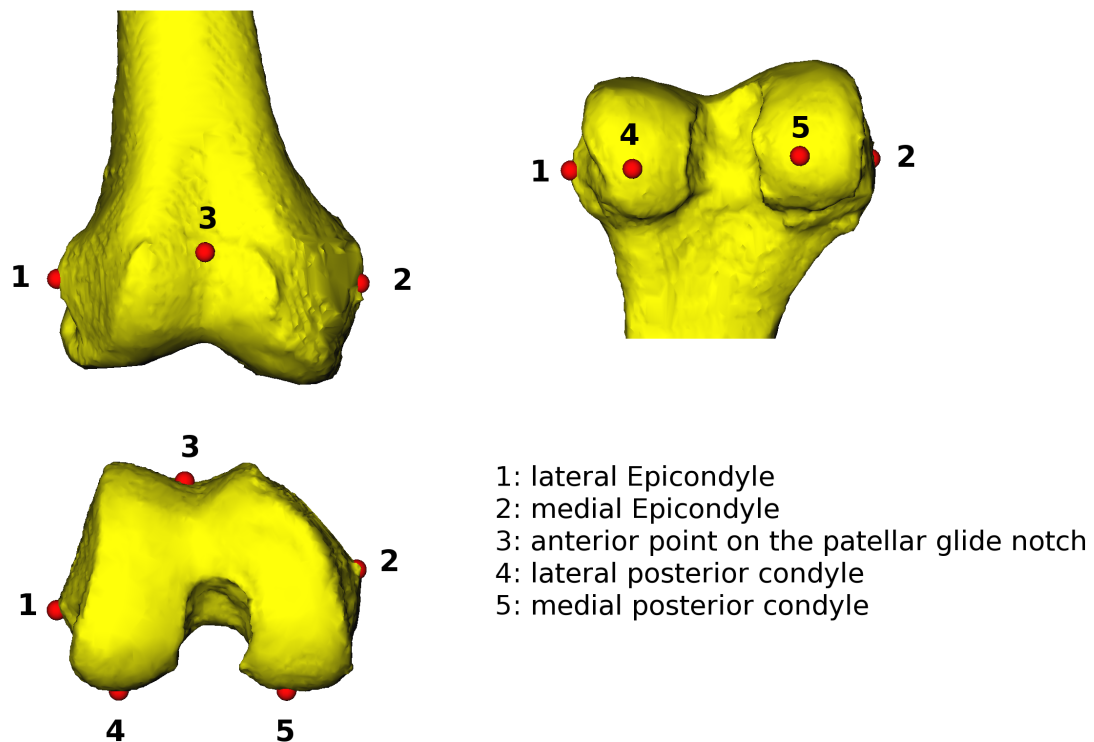


Figure 4: Location of anatomical landmarks

**Task 1: Read data****max. 4 points**

- i) Implement a Matlab function to read the image data and create a 3d voxel volume.

function [volume] = read\_data(prefix,nr,type) with

volume: 3d matrix containing the voxel information for a image dataset

prefix: common prefix for all images in the dataset, for example

'CT\_Bin/knee\_ct\_' or 'MRI\_Bin/knee\_mri\_'

nr: number of slices in the image dataset

type: pixel type of dataset

- ii) To verify your implementation from i) display some of the slices for each dataset (MRI and CT). Compare your axial slices with the ones displayed in figure 3. You may choose to display just some axial slices or also frontal and sagittal slices.

*You do not need to document this subtask for the assignment.* However, since the rest of the assignment relies on you being able to correctly read the data, I trust you will perform this subtask.

**Task 2: Display of landmarks****max. 5 points**

Parts of the following tasks will require you to display landmark positions in the voxel volumes. One easy way of doing this is to actually modify the voxel value at the position of the landmark to create a contrast rich grayvalue at this point. However, one voxel value might be hard to see, therefore it is prudent to extend the single point to a 3d cross.

- i) Implement the following Matlab routine only using elementary Matlab operations (like '+', '-', 'for', 'while', etc)

function [] = display\_landmarks(volume, origin, landmark, value) with

volume: 3d matrix containing the voxel information for a image dataset

origin: 3x1 vector containing the origin of the image dataset (in mm)

landmark: 3x1 vector containing x,y,z information of landmark (in mm)

value: voxelvalue in which landmark will be marked.

Running this Matlab function should create a copy of the volume dataset in which the position of the landmark is marked with a cross, which extents 3 voxels along the x, y, and z direction.

Display three orthogonal slices (axis, frontal and sagittal), which intersect at the landmark position.

Note:

- In this task you have to transform between the voxel volume coordinate system (in which your volume is stored) and the imaging device coordinate system (in mm).
  - Don't forget: The voxel coordinates of the Matlab 3d matrix is (1,1,1). However, for the CT volume the corner has the coordinates (0,0,0).
- ii) Run the matlab routine for each of the above given anatomical landmarks (for both datasets) and confirm the position of the landmark. Choose a contrast rich value based on the image modality. You do not need to document the output of these tests (images). However, document all function calls (with all parameters), either by providing a Matlab script or by documenting the function calls in a text document.

### **Task 3: Multi-modal Registration using a Pair-Point Algorithm *max. 10 points***

The goal of this task is it to establish the relationship between the coordinate systems of each image modalities, which will allow you to use information of both modalities together.

- i) Implement the Horn's algorithm for pair-point registration as discussed in lecture 15. Please DO your own implementation and don't use any downloadable implementation of Horn's. You can use the Matlab functions 'eig' and 'trace' and any elementary mathematical operation (such as '+', '-', ...).

Ensure that your routine has the following format:

function [q, rotation, translation, rms] = horns(P,X) with

q: 4x1 vector containing the quaterion elements of the rotation

rotation: 3x3 rotation matrix

translation: 3x1 translation vector

rms: double value for Root Mean Square Error as a measurement of the final alignment error

P: 3xn matrix containing the coordinate of the n landmark points in the first coordinate system

X: 3xn matrix containing the coordinate of the n landmark points in the second coordinate system

Outputs are the rotation and translation information *to align the point dataset P to the point dataset X*, as well as the Root Mean Square (RMS) error of the final alignment. To test your implementation use a synthetically generated X data set and apply a known transformation T (rotation and/or translation) to the data set to create the P point set. Run Horn's Method using these generated point sets. The result should be the inverse transformation of T. *Document this test.*

- ii) Use your Horn's method from i) to calculate the transformation of the CT coordinate system to the MRI coordinate system using the five anatomical landmarks defined in each dataset (see above). Document the output of the registration.  
Apply the resulting rotation and translation to transform the coordinates of the CT landmarks into the MRI coordinate system. Use your routine 'display\_landmarks' from Task 2 to display the transformed points in the MRI dataset. Document the output for one of the landmarks.
- iii) Use your Horn's method from i) to calculate the transformation of the MRI coordinate system to the CT coordinate system using the five anatomical landmarks defined in each dataset (see above). Document the output of the registration.  
Apply the resulting rotation and translation to transform the coordinates of the MRI landmarks into the CT coordinate system. Use your routine 'display\_landmarks' from Task 2 to display the transformed points in the CT dataset. Document the output for one of the landmarks.

#### **Task 4: Visualization of segmentation**

***max. 10 points***

The goal of this task is to create a function to visualize the segmentations.

- i) Implement the following Matlab routine

```
function [] = segmentation_visualization(volume, origin,
segmentation_file) with
```

volume: 3d matrix for voxel volume

origin: 3x1 vector containing the origin of the image dataset (in mm)

segmentation\_file: name of segmentation file

Functionality of this routine should include:

- Create a copy of the voxel volume
- Modify the voxel value for each position identified in the segmentation file as being part of the object. The new voxel value should be the maximum voxel value +1 in the original volume.

- Display five equally distributed axial slices of the modified voxel volume.
- ii) Run the visualization routine for the CT voxel volume and the CT segmentation 'CT\_mask.txt', which contains the segmentation of the femur bone. Document the output.
  - iii) Run the visualization routine for the MRI voxel volume and the MRI segmentation 'MRI\_mask.txt', which contains the segmentation of the femur bone and cartilage. Document the output.

### **Task 5: Determine Cartilage Volume**

*max. 12 points*

If you compare the outputs for Task 4 ii) and iii) you will see that the segmentation for the CT only contains the femur bone, while the MRI segmentation contains bone and cartilage. The goal of this task is to develop a routine which uses information from both segmentations to calculate the cartilage volume of the femoral bone in the knee for this patient.

Create a Matlab routine, which has the following functionality:

- i) Using the output from Task 3 create a Matlab routine to determine a new segmentation mask 'Bone\_inMRI\_mask.txt' for the femoral bone in MRI coordinate system by transforming the coordinate of the CT\_mask.txt into the MRI coordinate system. Verify the transformation by visualizing this new segmentation mask in the MRI volume dataset (using the routine from Task 4 i)). Document the output of this test.
- ii) Create for both MRI segmentations 'MRI\_mask.txt' and 'Bone\_inMRI\_mask.txt' a binary voxel volume, containing a 1 if the voxel is part of the segmented object, and a 0 if not.
- iii) Create a binary voxel volume only containing cartilage information by subtracting the two binary volumes.  
Note: Verify that your resulting volume is in fact a binary volume.  
Determine and document the volume (in mm<sup>3</sup>) of the cartilage object.
- iv) Create a new segmentation mask 'Cartilage\_inMRI\_mask.txt' from the created binary volume in iii).
- v) To verify the new segmentation mask use your visualization routine from Task 4i) to display the object in the MRI voxel volume. Document the output and save the segmentation mask as an 'ascii' file.

**Note:** Latest in this task you will see the effect of some rounding operations. For example, your cartilage mask might have some noise represented as small objects which are clearly not part of the cartilage. Don't worry about this. Just treat it all as cartilage.



## Task 6: CT of the future

*max. 9 points*

As discussed in class, CT is a modality that can create contrast rich images of bony anatomy, but has a low contrast for soft-tissues. Therefore it is normally not possible to see cartilage information in CT datasets. Imagine that you will discover a contrast material, which will be non-toxic, easy to digest, and is absorbed only by cartilage. The Hounsfield unit for this new contrast material is 2000. If this new contrast material is digested or injected by the patient before the CT scan of the knee was taken and the contrast material was absorbed by the cartilage of the femur, the cartilage would have been imaged with HU of 2000.

The goal of this task is to simulate the slices for this new image modality.

Develop a Matlab routine that has the following functionality:

- i) Based on the saved cartilage segmentation mask 'Cartilage\_inMRI\_mask.txt' from Task 5 iv) create a new segmentation mask 'Cartilage\_inCT\_mask.txt' by transforming the coordinates from the MRI coordinate system into the CT coordinate system. To perform the transformation use the result of the Task 3. Choose the voxel values in the new mask carefully. Verify the transformation by visualizing this new segmentation mask in the CT volume dataset (using the routine from Task 4 i)). Document the output of this test and save the segmentation mask in an 'ascii' file.
- ii) Create a new voxel volume for the CT dataset in which voxels, which are part of the cartilage, have the HU of 2000.
- iii) Create a CT dataset contains 299 slices based on the new voxel volume created in ii). Each slice should have the dimension of **190 x 190** pixels, with a **pixel type 'int16'**. Each CT slice should be stored in MATLAB 2d matrix format in the file 'knee\_ct\_new\_nr.bin'. To verify your dataset use the routine developed in Task 1 and display 3 axial, 3 frontal and 3 sagittal slices which are distributed evenly. Document the output.
- iv) Provided data for this task will also contain the 299 Dicom files for the 'original' CT slices (in "CT\_Dicom" directory). The files are numbered in the same order as the CT binary image files (for example 'CT\_knee\_10.bin' contains the same image information as 'CT\_Knee\_0010.dcm'). Write a routine in which you read the dicom files, replace the pixel information in the file with the pixel information of the corresponding file from iii). We assume that all the attributes from the original dicom file are still valid in this new dicom file. Save the resulting dicom file as 'CT\_Knee\_New\_n.dcm'. To verify this step, you should be able to read the stack of new dicom files in any open source or commercial DICOM reader, such as OsiriX (MacOS), MedCon or xMedCon (Windows and Linux), DICOM WORKS (Windows), etc.  
Submit the new DICOM files with your assignment.

*Note:* Check the Dicom format. Where are the pixel information stored?

**Extra Task:****max. 6 points**

This task will give you the chance to earn an extra 6 points.

In task 5, volume particles which are the result of noise are considered cartilage and part of the volume calculation. For this task, copy and modify your matlab routine from task 5 to exclude these noise voxels from the calculation. The goal is to only consider voxels, which are part of the largest connecting object in the binary volume.

**Further Instruction:**

You can either submit the assignment alone, or as a group of 2.

*Single submission:* I encourage you to discuss the project with your classmates, but you will need to write the programs yourself.

*Group submission:* If you decide to submit as a group, you must send me an email at the latest 1 week before the final submission to inform me about your group. Also, in addition to the assignment, you will need to attend a 10-15 minute long oral defense of the submission in which I will confirm that both members of the group worked on the assignment. This defense will be held within a few days after the final submission date.

**Office Hours:**

March 19<sup>th</sup> at 2-3pm  
March 24<sup>th</sup> @ 2-3pm  
March 26<sup>th</sup> @ 2-3pm  
March 31<sup>th</sup> @ 2-3pm  
April 2 @ 2-3pm  
April 7<sup>th</sup> @ 2-3pm  
April 9<sup>th</sup> @ 2-3pm

All office hours will be held in the KGH in the Human Mobility Research Centre. Take the entrance at the corner of Stuart and George St. The research centre is right opposite the entrance and our receptionist will help you find my office.

**Submission:**

Zip all files (Matlab files, documentation, image files) and submit the zip file to moodle.

Text file can be submitted in word, text or pdf-format.

**Submission deadline: 13<sup>th</sup> April 2015 @  
11:59pm.**