University of Ljubljana, Faculty of Electrical Engineering
Robot Vision (RV)

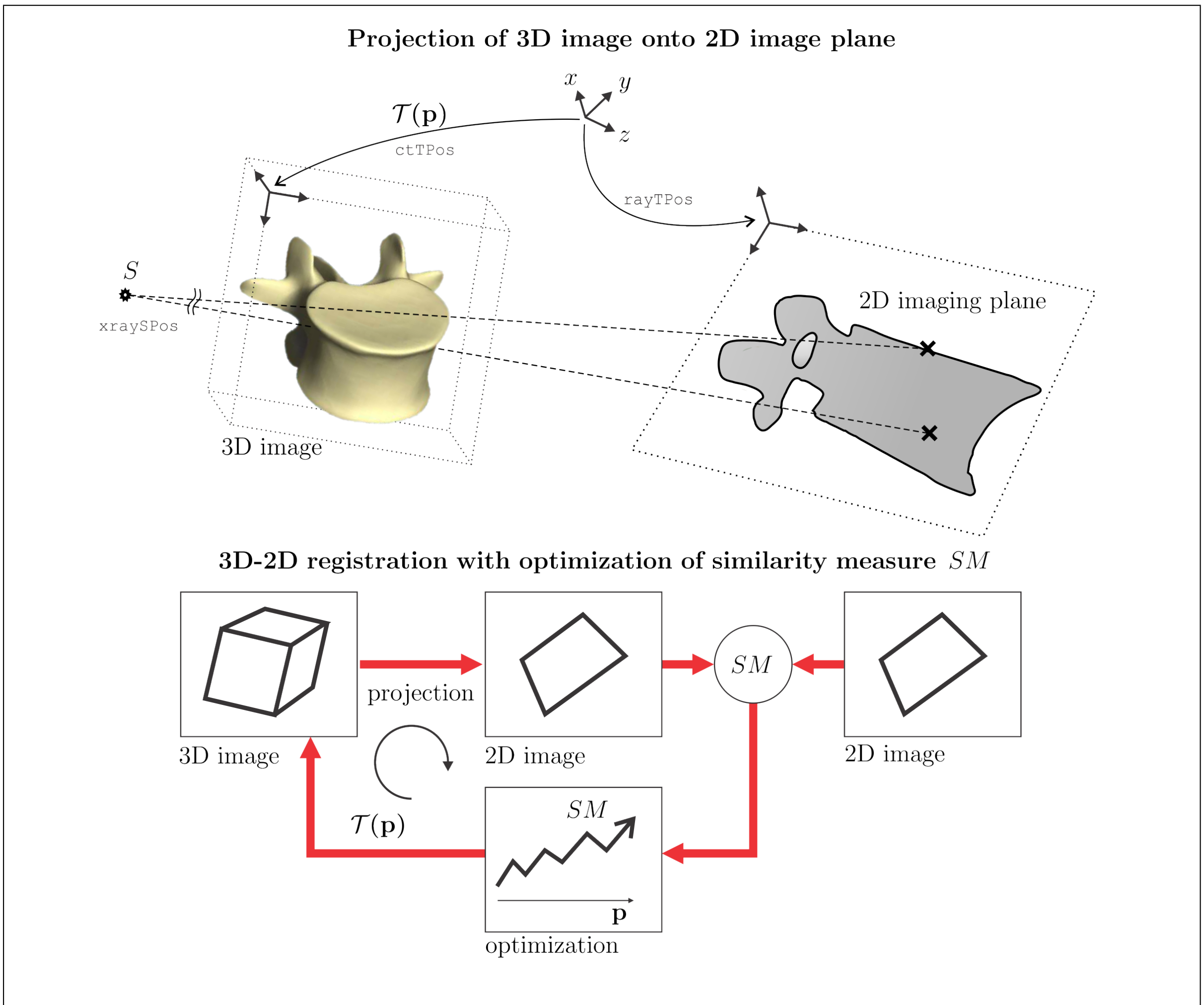
May 10/11, 2016

Exercise 8: Registration of 3D Models to 2D Images

Created by: Žiga Špiclin | http://lit.fe.uni-lj.si/RV | Homework deadline: May 17/18, 2016

Instructions

Registration of a 3D to a 2D image is concerned with finding an optimal geometric transformation $\mathcal{T} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, which transforms the 3D image V into a pose such that its projection is *consistent* with the projection of the same object on the 2D image P. The main challenge of 3D to 2D image registration is to overcome the dimensional correspondence (3D vs. 2D), which can be addressed in two ways: 1) by projection of 3D information to 2D image space or 2) by reconstruction of 3D image from multiple 2D projections or 2D images. During this exercise you will implement a 3D-2D image registration by projecting the 3D image into 2D ($\mathcal{P} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$) and maximizing the similarity measure SM between the projection $\mathcal{P}(V)$ and the 2D image P.



The implementation of projection operator $\mathcal{P}$ depends on the form of the 3D information. In the assignment you will work with the 3D CT image, in which the intensity of each voxel represents the atenuation coefficient $\mu(\mathbf{x})$ of the corresponding structure. For the purpose of 3D to 2D image registration you will use the 3D CT image to simulate the 2D X-ray images, or the so-called digitally reconstructed radiographs (DRRs). The DRRs are obtained by computing the integral of the atenuation coefficient along the line connecting the X-ray source S and the 2D imaging plane, i.e. $\mathcal{P}(V) = \int_l \mu(l)dl$. The projection and symbols are shown in Figure below.

The spatial alignment of structures between the projection and 2D X-ray images $\mathcal{P}(V)$ and P, respectively, can be evaluated by a similarity measure SM; a scalar function whose value is optimal (minimal or maximal) when the positions of the corresponding structures in $\mathcal{P}(V)$ and P are mutually aligned. The similarity measure has to be carefully chosen for a particular application such that it is insensitive to image noise and other image artifacts, but at the same time, highly sensitive to the actual geometric inconsistencies between the two images undergoing registration. In image registration, the **mutual information** MI is often used:

$$MI(I,J) = H(I) + H(J) - H(I,J)\,,$$

where $H(I)$ and $H(J)$ are the marginal entropies of the reference $I(x,y)$ and moving $J(x,y)$ image, respectively, and $H(I,J)$ is their joint entropy:

$$\begin{aligned}
H(I) &= -\sum_{s_I=0}^{L-1} p_I(s_I)\log p_I(s_I)\,,\\
H(J) &= -\sum_{s_J=0}^{L-1} p_J(s_J)\log p_J(s_J)\,,\\
H(I,J) &= -\sum_{s_I=0}^{L-1}\sum_{s_J=0}^{L-1} p_{IJ}(s_I,s_J)\log p_{IJ}(s_I,s_J)\,,
\end{aligned}$$

where s_I in s_J denote the co-occuring discrete intensity values of the reference and moving images $I(x,y)$ and $J(x,y)$, respectively, and L denotes the number of bins. The marginal probability density functions (PDFs) $p_I(s_I)$ in $p_J(s_J)$ and the joint PDF $p_{IJ}(s_I,s_J)$ is obtained from the associated normalized intensity histograms $h_I(s_I)$, $h_J(s_J)$ and $h_{IJ}(s_I,s_J)$ as:

$$p_I(s_I) = \frac{h_I(s_I)}{N\cdot M}\,,\quad p_J(s_J) = \frac{h_J(s_J)}{N\cdot M}\,,\quad p_{IJ}(s_I,s_J) = \frac{h_{IJ}(s_I,s_J)}{N\cdot M}\,,$$

where N and M are the image dimensions along the x and y axes.

During the registration of 3D and 2D images the optimization method is used to iteratively update the parameters $\mathbf{p}$ of the geometric transformation $\mathcal{T} = \mathcal{T}(\mathbf{p})$ such that the similarity measure is maximized:

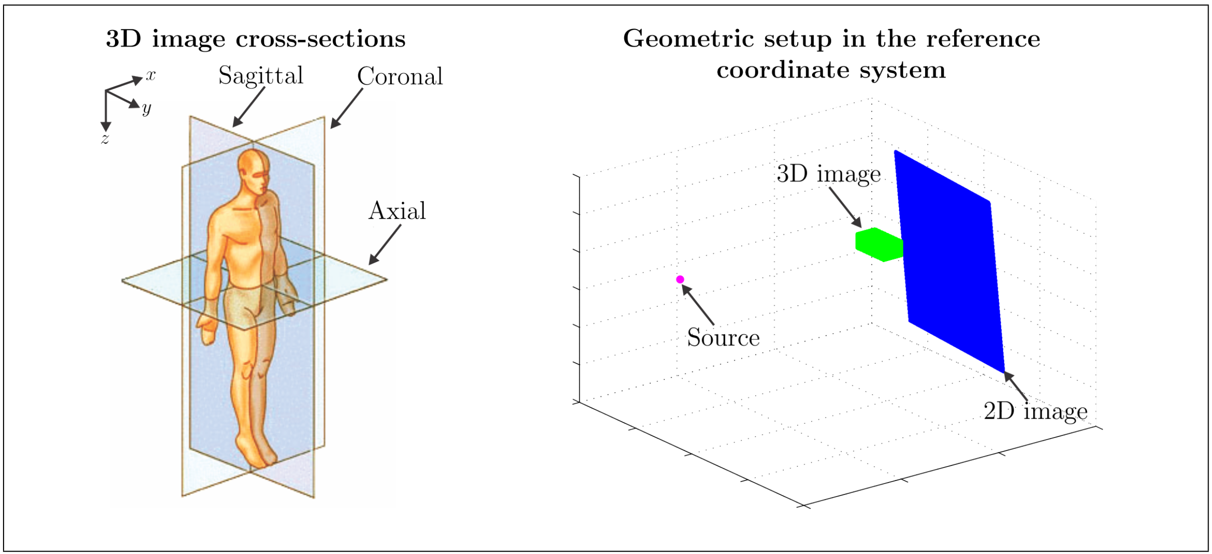
$$\mathbf{p}^* = \operatorname{argmax}_{\mathbf{p}} SM\Big(\mathcal{P}\big(V[\mathcal{T}(\mathbf{p})]\big), P\Big)\,,$$

where $\mathbf{p}^*$ represent the optimal parameters of the geometric transformation $\mathcal{T}(\mathbf{p})$. As the vertebrae are rigid structures you will use the 3D rigid-body transformation, which is defined by six parameters $\mathbf{p} = [t_x, t_y, t_z, \alpha, \beta, \gamma]^{\mathrm{T}}$. The flowchart of 3D to 2D image registration is shown in Figure above.

Materials required for the assignments are given in file `vretenca.npz`, which contains two 3D CT images and a 2D X-ray image. Load the data in the file using function `numpy.load()`, while by reading the properties `files` you can obtain the names of contained variables. All the variables are given as `numpy` arrays. Variables `ctVol`, `ct2Vol` and `xrayImg` are the respective 3D and 2D arrays of grayscale values, while variables `ctTPos`, `ct2TPos` and `xrayTPos` represent the geometric transformations of 3D and 2D images to a reference coordinate system with respect to the first voxel and pixel, respectively (`[0,0,0]` and `[0,0]`). Variable `xraySPos` denotes the position of the X-ray source in the reference coordinate system. The spatial sampling of the 3D and 2D images is 1 milimeter isotropic. The meaning of the geometric quantities is illustrated in the Figure above.

1. For the following assignments you should create a variable of type `dict`, for instance `ct`, `ct2` in `Xray`, and insert the array of grayscale values and corresponding transformation to the reference coordinate system (`...TPos`) and, for the 2D X-ray, also the position of the X-ray source (`xraySPos`).

2. Create sampling grids of the 2D X-ray and 3D CT images by using function `numpy.meshgrid()`. Transform the sampling grids into the reference coordinate system by using the respective transformations `Xray.TPos` and `ct.TPos`. Display the transformed 3D and 2D sampling grids and the position of the X-ray source `Xray.SPos` by using the function `Axes3D.scatter()`. Verify the obtained geometric setup of the X-ray source, 3D and 2D images mapped into the reference coordinate system by comparing the obtained display with the Figure below.



3. Write a function that maps an arbitrary point `iPos` in 3D space on the 2D image plane with respect to the X-ray source S (`xraySPos`):

```
def mapPointToPlane( iPos, Xray ):
    return oPos
```

where `iPos` is a $N \times 3$ array and `Xray` a variable containing information about 2D X-ray image. Function should return 3D points `oPos` in the form of a $N \times 3$ array. The 3D-to-2D mapping of 3D points can be computed as the intersection of 2D image plane and the lines that emanate from the X-ray source and pass through point `iPos`. Verify the function by mapping the coordinates of the corners of the 3D image onto the 2D imaging plane and display the points into the geometric setup, similarly to the previous assignment.

4. Write a function for cone beam projection of 3D CT image into the 2D imaging plane that represents the digitally reconstructed X-ray projection (DRR):

```
def project3DTo2D( ct, Xray, iStep ):
    return oDRR, oMask
```

where `ct` and `Xray` are the variables containing information about the 3D CT and 2D X-ray images and `iStep` is the sampling step in *milimeters* along a line from the X-ray source to the intersection the line with the 2D imaging plane. Function should return a 2D DRR image as `oDRR` and the corresponding 2D DRR mask image as `oMask`, both of which have dimensions 446×446 (equal to the dimensions of `Xray.image`). The 2D DRR mask is 1 in those points, for which the line from the X-ray source to the imaging plane intersects the 3D image, or 0 otherwise. Verify the function using `ct` and `Xray` images and create a projection of the 3D image. Verify the impact of sampling step `iStep` on the quality of the obtained projection image.

5. Write a function that computes the mutual information MI between two images:

```
def mutualInformation( iImageI, iImageJ, iBins ):
    return oMI
```

where `iImageI` and `iImageJ` are 2D grayscale images and `iBins` the number of intensity subintervals used to compute the intensity histograms. Function should return a scalar value in variable `oMI`. Compute the mutual information between the obtained projection image and the 2D X-ray image using only the intensity values that lie in the projection mask. Verify the impact of the number of intensity subintervals `iBins` on the value of mutual information.

Homework Assignments

Homework report in the form of a Python script entitled `NameSurname_Exercise8.py` should execute the requested computations and function calls and display requested figures and/or graphs. It is your responsibility to load library packages and provide supporting scripts such that the script is fully functional and that your results are reproducible. The code should execute in a block-wise manner (e.g. `#%% Assignment 1`), one block per each assignment, while the answers to questions should be written in the corresponding block in the form of a comment (e.g. `# Answer: ...`).

1. The contrast of the vertebrae in the 2D X-ray image can be improved by intensity windowing. Transform the unsigned 16-bit grayscale intensity values of the 2D X-ray image into unsigned 8-bit grayscale intensity values so as to obtain optimal contrast in regions displaying the vertebrae. *The windowed 2D X-ray image should be used in all subsequent assignments!*

2. Compute the median intensity value from each of the two 3D CT images and subtract the obtained value from all the intensity values. Display the medial 2D cross-sections of the 3D image in sagittal, coronal and axial view. *The median-subtracted 3D images should be used in all subsequent assignments!*

3. Write a function for 3D rigid transformation of the sampling grid of 3D CT image about its center:

   ```
   def rigidTrans( ct, iPar ):
       return oX, oY, oZ
   ```

   where the input variable `ct` represents the 3D CT image and `iPar` a 6×1 vector of the 3D rigid-body parameters $\mathbf{p} = [t_x, t_y, t_z, \alpha, \beta, \gamma]^{\mathrm{T}}$, where the rotations are defined with respect to the center of the 3D image. The function outputs `oX`, `oY` and `oZ` that represent the respective transformed x, y and z coordinates of the sampling grid of 3D CT image. Display the geometric setup containing the X-ray source and the sampling grids of 3D and 2D images in the reference coordinate system, where the sampling grid of the 3D image is transformed using `iPar = [0 0 0 90 0 0]`.

4. Extend the function for 3D to 2D projeciton by adding a new input parameter `iPar`, for instance, `project3DTo2D( ..., iPar )`. Let `iPar` be a 6×1 vector of the 3D rigid-body parameters $\mathbf{p} = [t_x, t_y, t_z, \alpha, \beta, \gamma]^{\mathrm{T}}$, where the rotations are defined with respect to the center of the 3D image. Generate the DRR images for the following parameter values `iPar = [0 20 0 0 0 0]`, `iPar = [0 0 0 0 45 0]` and `iPar = [0 0 0 0 0 90]`.

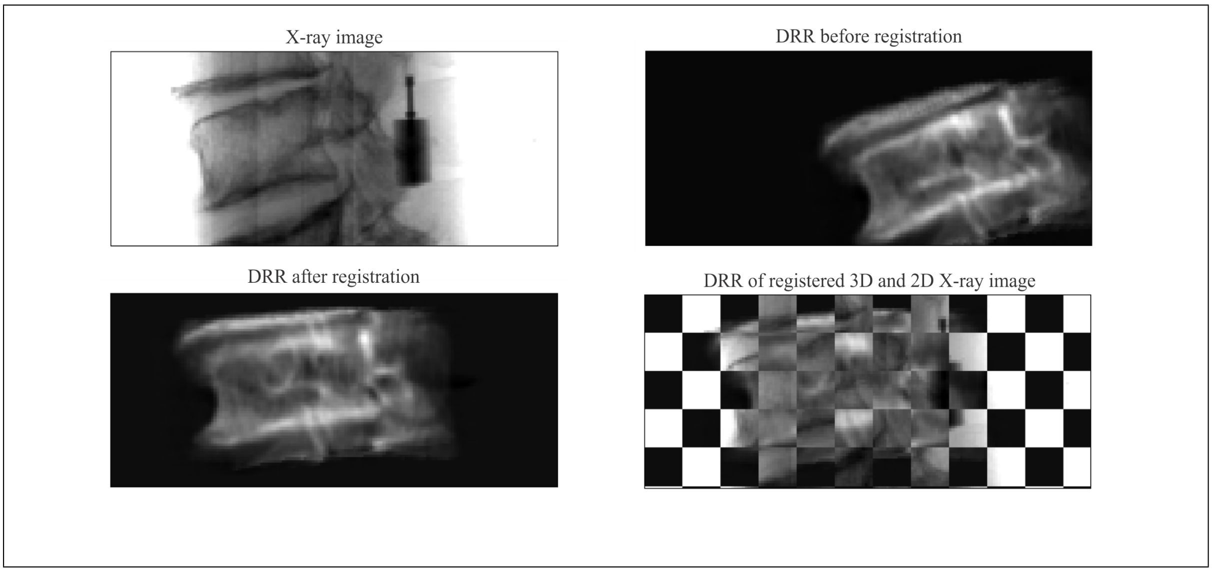
5. Compute the values of mutual information MI between the 2D X-ray image and several DRR images, which are obtained by varying one parameter of the 3D rigid-body transformation at a time. Let translations t_x, t_y and t_z vary in the range from -20 to 20 milimeters with a step of 2 milimeters, and the rotations α, β and γ in the range from -10 to 10° with a step of 1°. Plot the graphs $MI(t_x)$, $MI(t_y)$, $MI(t_z)$, $MI(\alpha)$, $MI(\beta)$ and $MI(\gamma)$.

6. Elaborate on the connection between the value of the mutual information `oMI` and the actual similarity of the 2D X-ray and DRR images. Which values, higher or lower, represent higher similarity between the images? Which are the theoretically minimal and maximal values of mutual information?

7. Devise an automated method for the rigid-body registration of 3D to 2D images based on iterative optimization with a criterion function:

   ```
   def criterionFcn( iPar, ct, Xray ):
       return oMP
   ```

   which takes as input some arbitrary 3D rigid-body parameters `iPar` and computes the corresponding value of mutual information MI between the DRR and 2D X-ray images. To obtain the optimal values of the 3D rigid-body parameters $\mathbf{p}^*$ use the Nelder-Mead simplex optimization (Exercise 7). Verify the optimization method by registering `ct2` and `Xray` images. Report the obtained optimal values of 3D rigid-body parameters $\mathbf{p}^*$, which result in an optimal registration of 3D CT to 2D X-ray image, and show the corresponding 2D DRR and X-ray images as in Figure below.


X-ray image
DRR before registration
DRR after registration
DRR of registered 3D and 2D X-ray image