

5 *HipHop*: semiautomatic 2D-3D registration of CT or CAD to fluoroscopic images.

5.1 Introduction

The *HipHop* registration framework performs 2D-3D registration of a CT scan or CAD model of the hip joint to a time sequence of fluoroscopic images acquired during the joint motion. The registration algorithm was adapted for the analysis of both the prosthetic and the native hip joint. The direct application in the context of this thesis was the retrieval of the kinematics of the prosthetic hip joint during different activities of daily living, for them to be provided as input to wear simulations or to musculoskeletal models, or as a tool for post-operative functional evaluation of THA.

The main challenges presented by image registration of the hip joint during motion were (Figure 5.1):

- *The image quality.* Due to the large amount of soft tissue and bone structure surrounding the hip joint, the contrast-to-noise ratio of fluoroscopic images of the hip is considerably reduced compared to other joints such as the knee and the ankle. Moreover, the image contrast can vary significantly within the same motion

sequence depending on the amount of overlap of the contralateral limb, thus making difficult to find a consistent set of parameters for optimal registration.

- *Overlap.* The constant overlap between the pelvic and the femoral segments reduces the amount of features (i.e. contours) available for registration, hence may decrease the achievable accuracy in pose retrieval. In addition, frequent false contours generated by the surrounding soft tissue may lead to misregistration.
- *Missing parts.* Due to the small field of view of the clinical fluoroscopic C-arms (30 cm), a relatively small portion of the hip joint is typically visible in the images compared to the size of the pelvis and of the femur. Missing parts may reduce the registration accuracy, in particular for the out-of-plane degrees of freedom.
- *Symmetry.* The quasi-symmetrical shape of the femoral segment makes hard the retrieval of the rotational degree around its main axis (as in internal-external rotation).
- *The motion blur.* Motion related blur is caused by the relative movement between the target and the moving fluoroscopic machine, and it is dependent on the specific activity as well as on the fitness of the patient. Generally, blur represents a less serious issue for the hip joint compared to the knee joint, due to the lower involved accelerations.

Furthermore, a common issue to tackle for the fluoroscopic analysis of the human joints during motion is the required *computational time*. The amount of fluoroscopic images acquired during our measurements was considerably large. The measurement of five motion cycles for four activities of daily living, with an average duration of 1.4 seconds per cycle and acquisition frame rate of 25 to 30 Hz led to about 12.000 fluoroscopic frames

for all 15 THA patients included in the study at ETH Zurich. Registration of such a large amount of images is extremely time demanding.

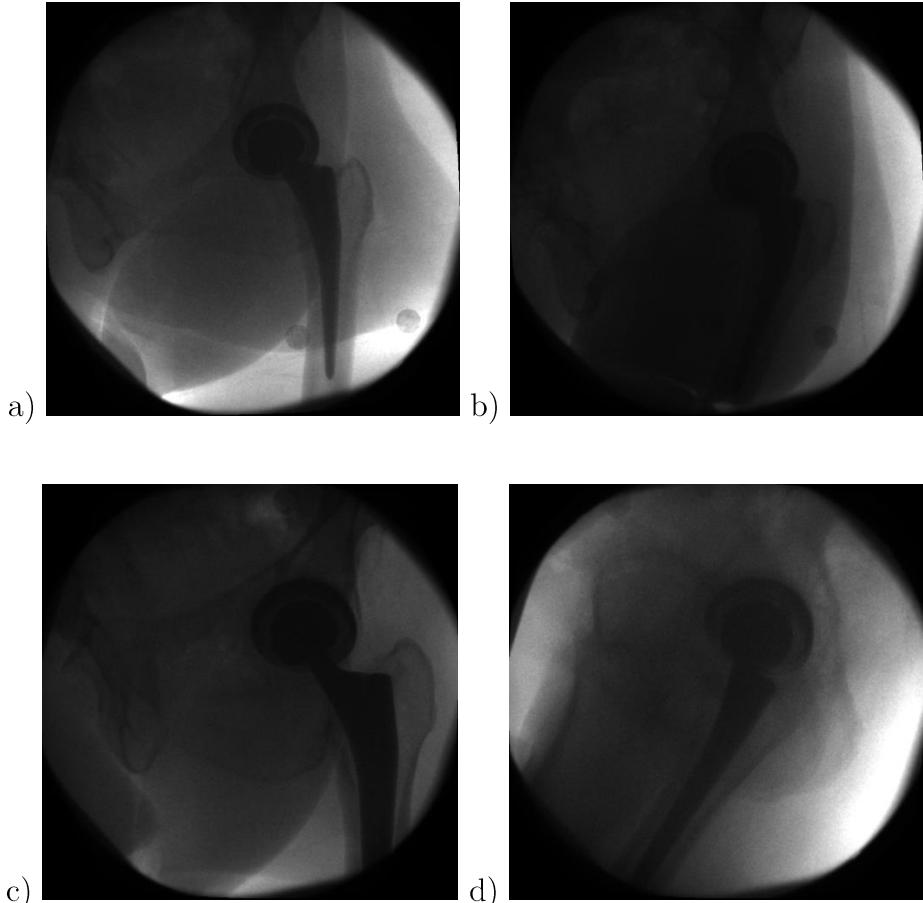


Figure 5.1. Challenges in X-ray fluoroscopy of the hip joint during motion. Images a) and b) belong to the same gait cycle and their comparison shows the varying image contrast depending on the amount of overlap with the contralateral limb. Image c) shows a typical case of missing part, in which the femoral component is partly outside of the field of view. Image d) is acquired during the higher acceleration phase of the chair rise, and shows motion blur.

An overview of the *HipHop* registration framework is provided in Figure 5.2.

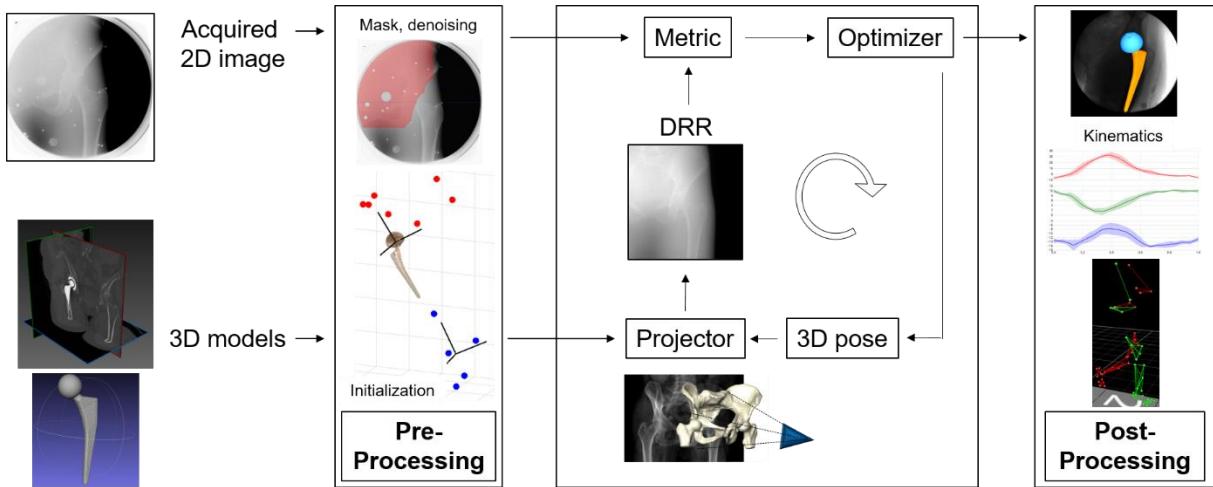


Figure 5.2. Overview of the HipHop 2D-3D registration framework. Input are a sequence of acquired fluoroscopic images, and a CT scan or a CAD model of the hip joint. Prior to registration, input images are de-noised and masks are generated to identify the region of interest in which registration shall be performed; also, the pose of the 3D model is initialized to a close guess. The core of the registration framework is the optimization routine: digitally reconstructed radiographs (DRRs) are iteratively generated for each guessed pose of the 3D model; the optimal pose is found that maximizes the similarity between the DRR and the acquired image, according to a predefined similarity metric. At last, the optimal pose is given as input to the post-processing step, which produces the kinematic information required for the motion analysis of the hip joint.

5.2 Methods

The *HipHop* registration framework was implemented with the Anaconda Python distribution [173] on Visual Studio Community 2015 (Microsoft Corp, Redmond, WA, USA). It runs on a regular PC for registration between CAD models and fluoroscopic images. Registration between a CT scan and fluoroscopic images was run on HP PC with 16 GB RAM memory and with a graphic card NVidia Quadro M5000 with 8 GB memory.

HipHop makes partly use of the open source Insight Segmentation and Registration Toolkit (ITK) for medical image analysis and registration [174]. The ITK toolkit stresses out the fact that medical images are more than just N-dimensional rectilinear grids of regularly sampled pixel values: “*The information associated with the physical spacing between pixels and the position of the image in space with respect to some world coordinate system are extremely important. Registration, for example, is performed in*

physical coordinates. Improperly defined spacing, direction, and origins will result in inconsistent results in such processes. Medical images with no spatial information should not be used for medical diagnosis, image analysis, feature extraction, assisted radiation therapy or image guided surgery. In other words, medical images lacking spatial information are not only useless but also hazardous.” [175]. For the above reasons, ITK images include useful features for image registration, such as spacing, origin, size, orientation and spatial coordinates for each pixel, as well as regions of interest (Figure 5.3). ITK also provides a built-in pluggable registration framework. However, this framework was not used in *HipHop* due to the need of implementing custom modules, such as fast parallelized generation of Digitally Reconstructed Radiographs (DRR), and metrics and optimizers other than the ones included in the ITK package.

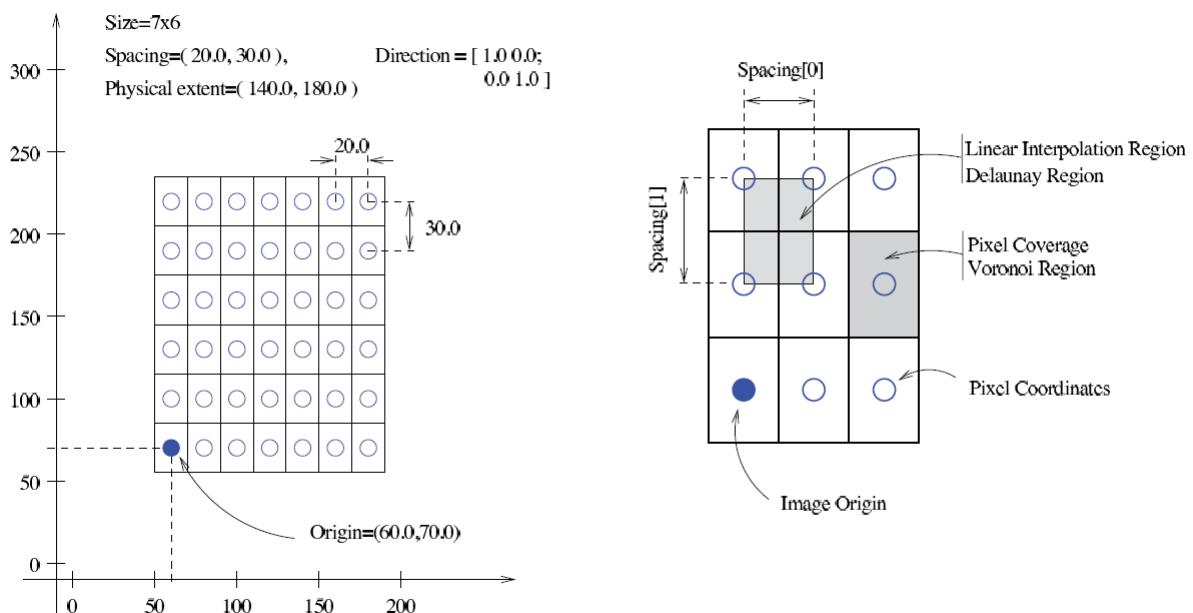


Figure 5.3 Geometrical concepts associated with the ITK image (from the ITK Software Guide Book [175]).

HipHop is structured in Python modules for each of the building blocks of the registration pipeline (Figure 5.2), which are described in more details in the following sections. The

modules “Projector”, “Metric” and “Optimizer” are implemented with a factory mechanism that allows easy plug-in of possible new future projectors, similarity metrics, and optimizers, respectively.

5.2.1 Pre-processing

The “Pre-processing” module prepares the input fluoroscopic image and the input CT scan or CAD model for the 2D-3D registration. It includes three main functions:

- *Denoising.* A required step for any image-based analysis is the denoising of the raw images. In *HipHop* denoising of the acquired raw fluoroscopic images was performed using the “Curvature Flow Image Filter” provided by the ITK library and adapted from [176]. The advantage of this approach is that sharp boundaries are preserved with smoothing occurring only within a region. This aspect is particularly important for similarity metrics that rely on edges, as the ones used in our application. The parameters chosen for our application were: number of update iterations = 30, and time step between each update = 0.5.
- *Masking.* Masking consists in the selection of the region of interest (ROI) of the fluoroscopic image that shall be compared to each of the generated DRRs (Figure 5.4a). Masking facilitates the convergence of the optimization to a correct solution by avoiding false registrations. These may occur due to false edge detection, and were frequent during registration of the hip implant for two reasons: 1) the silhouette of the acetabular cup was mistaken for the one of the femoral head, due to overlap of the components (Figure 5.4b), 2) the bone edge of the thigh was mistaken for the edge of the femoral shaft. Masking was also used when evaluating the accuracy of *HipHop* with the validation dataset described in Chapter 5: the

metal beads of the validation images were masked out, since they were used to obtain the ground truth but shall not be exploited by the tested registration algorithm. Masking was performed manually. The gain in robustness of the registration outcomes compensated for the additional user interaction that was necessary for masking.

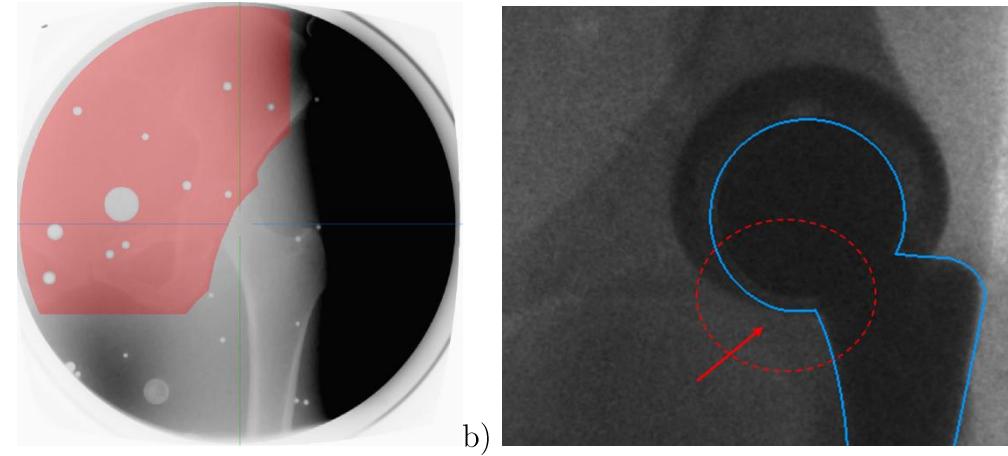


Figure 5.4. a) Example of masked image, where the pelvis only is selected for registration and the metal beads used for retrieval of the ground truth position are excluded. *b)* Example of false registration of the femoral head to the silhouette of the acetabular cup. Masking out the edge of the acetabular cup prevents the occurrence of this type of false registration.

1. *Initialization.* A good initialization of the searched pose is of paramount importance for convergence of any global or local optimization to the correct solution. However, a close initial guess manually provided by the user for each one of the thousands acquired images would prohibitively increase the required registration time. Since skin marker-based motion capture was performed simultaneously to video-fluoroscopy, we were able to perform a close initialization in an automatic fashion based on the motion capture positions (Figure 5.5). Reference skin marker-based segments of the hip joint were defined at the standing position, and their relative position with respect to the corresponding prosthetic

component was calculated. This relative position at the standing position was used to estimate the 3D pose of the prosthetic components for each acquired fluoroscopic image from the 3D pose of the relative skin marker-based segment. The 3D pose for each acquired frame of the skin marker-based segments was computed by a least-square fit with the markers cloud at the reference standing position.

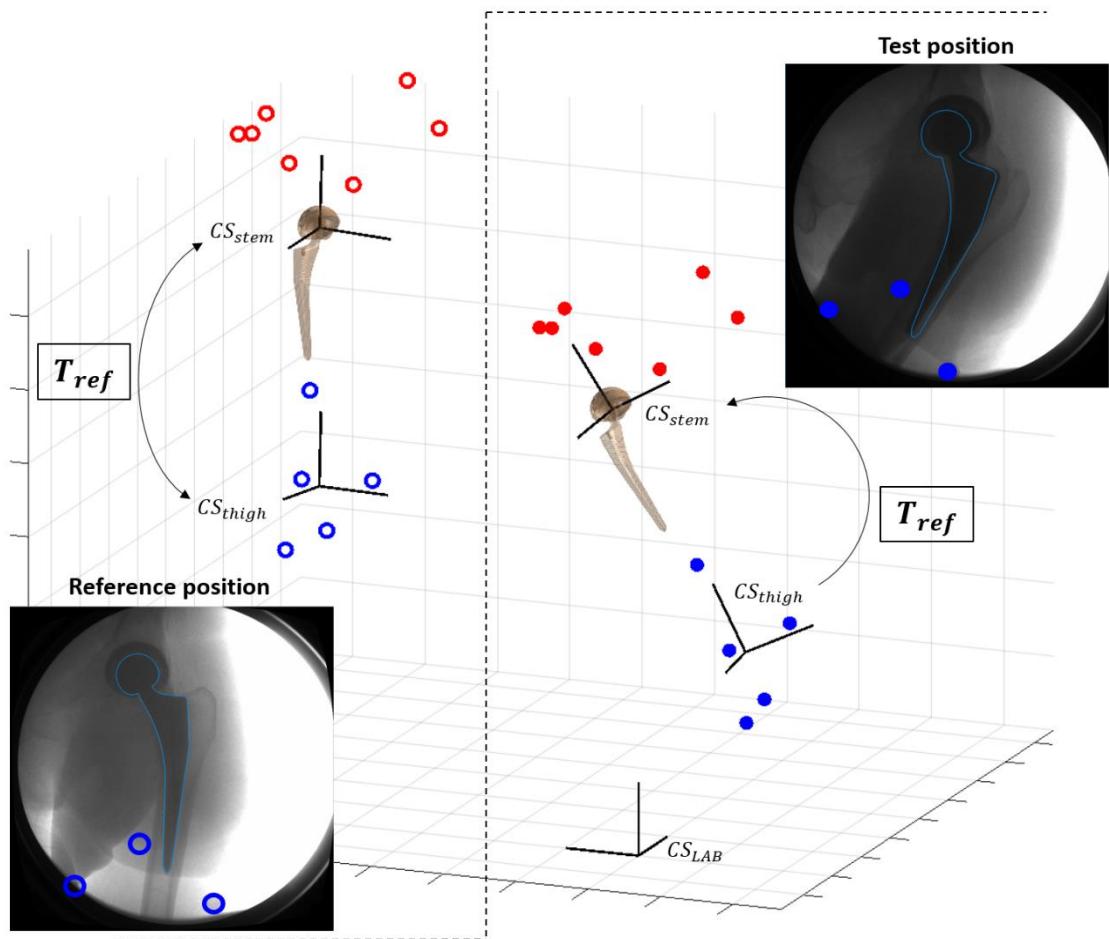


Figure 5.5. Scheme of pose initialization with skin markers (red for pelvis, blue for thigh): at reference standing position, the transformation T_{ref} between skin-marker based segment (CS_{thigh}) and prosthetic component (CS_{stem}) is computed. T_{ref} is used to initialize the pose of the prosthetic component for each new test position.

The pose initialized with motion capture is not optimal due to the soft tissue artefact. For each test position, the error of the initialized pose is directly proportional to the deviation of the skin markers from their configuration at the reference standing position. Thus, a closer initialization can be achieved by choosing a reference position for which the skin marker configuration deviates less compared to the standing position. Therefore, we optimized the initialization procedure by picking three possible reference positions chosen at characteristic phases of each motion trial to be registered. For each test position, the residual of the least-square fit between the skin marker clouds was computed relative to each possible reference position. The final initialization pose was obtained from the reference position providing the minimum residual (Figure 5.6).

One additional type of initialization for the in-plane translations was implemented specifically for the prosthetic hip implant, by exploiting the spherical geometry of the femoral head. The Hough transform was used to fit a circle to the silhouette of the femoral head (Figure 5.7), and the center of the fitted circle indicated the correct translation of the implant along both of the in-plane horizontal and vertical directions. However, this method was not used for registration of the fluoroscopic sequences acquired in our in-vivo study due to the required manual user interaction and consequent increase of the registration time, and due to the difficulty of Hough circle detection for images with poor image contrast.

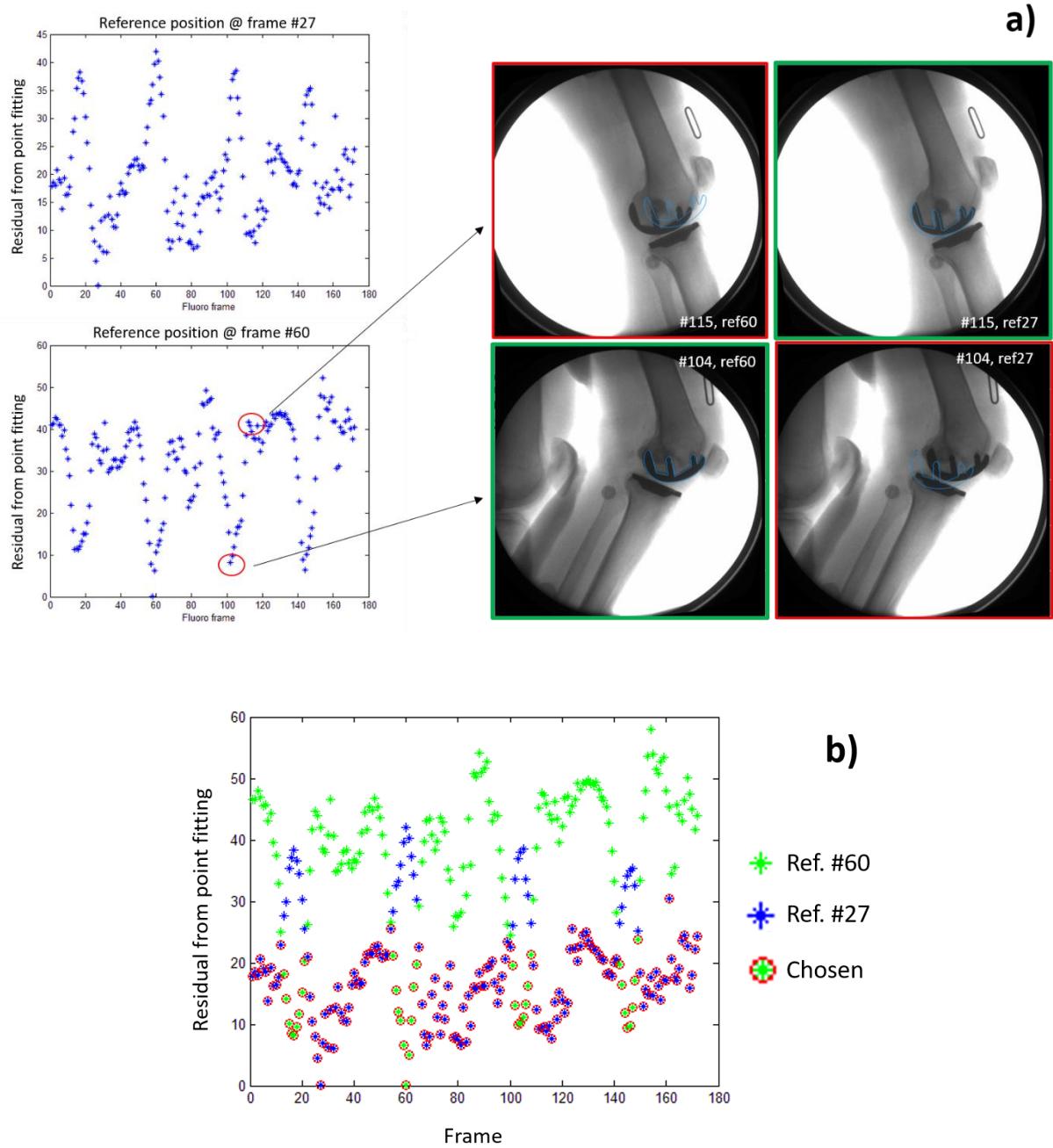


Figure 5.6. Improved initialization from skin marker-based motion capture (example for the prosthetic knee joint during one gait trial and for two reference positions). **a)** Two reference positions are chosen at frames #60 and #27, at the maximum and the minimum knee flexion during the gait trial respectively. It is evident that the high flexion test pose at frame #104 is better initialized by using the reference position at frame #60, and the low flexion test pose at frame #115 is better initialized by using the reference position at frame #27. **b)** The improved initialization procedure calculates the residual point fitting error for both reference positions throughout the whole trial (green dots for reference #60, blues dots for reference #27) and picks for each test frame the pose initialized from the reference position that provides the lowest residual error (red circles).

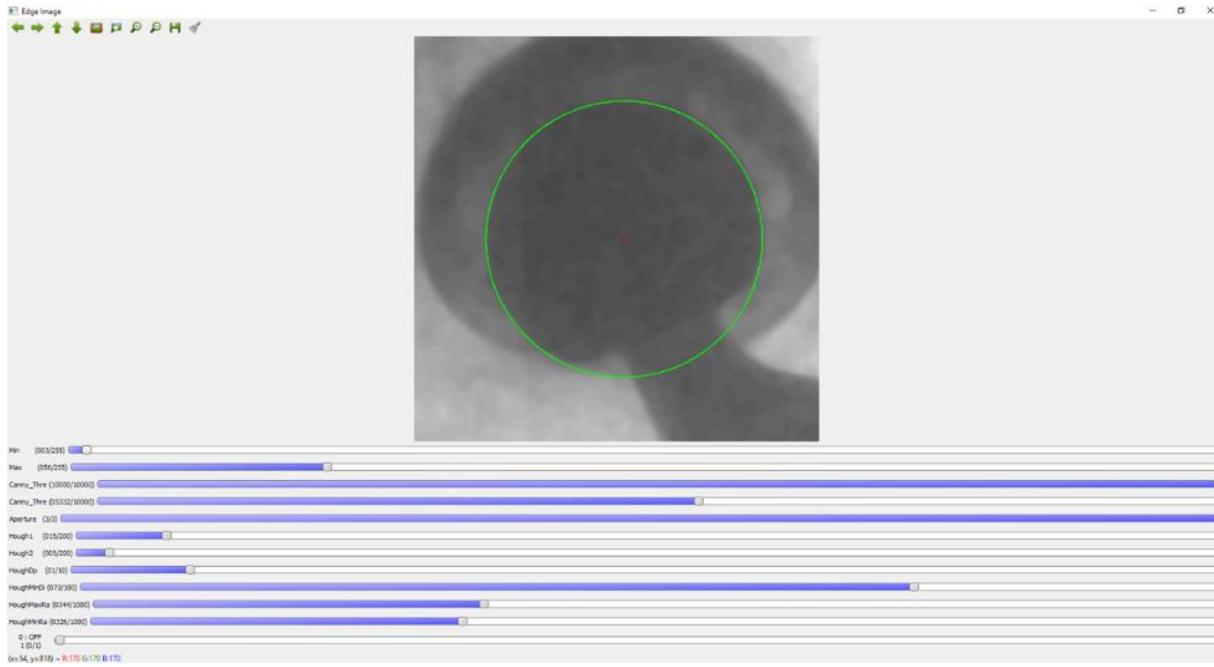


Figure 5.7. Graphical User Interface (GUI) developed to initialize the in-plane translations of the hip implant prior to registration. In-plane translations were provided by the location of the center of the fitted circle to the femoral head. The Hough transform was used to fit a circle.

5.2.2 Projector

The “projector” module implements the function that is used at each iteration of the optimization procedure in order to produce a DRR of the 3D model at a given pose. The DRR is then compared with the real fluoroscopic image to compute the similarity metric. Two type of projector functions were implemented in *HipHop*, depending on whether the 3D model is a tomographic CT scan or a CAD surface model.

DRR from CT scan

The DRR generation from a CT scan is described in detail in Annex 5.5. It is based on a combination of the serial algorithm proposed by Jacob [128] to improve the original

seminal Siddon's algorithm [127], and of the fast parallel implementation proposed by Greef et al. [126]. Shorty, a ray is casted from the X-ray source to the center of each one of the pixels of the image that is to be simulated. Each pixel value (called "radiological path length" RPL) of the simulated DRR is calculated as the weighted sum of the CT voxel values traversed by the relative casted ray:

$$RPL = \sum_i \sum_j \sum_k l(i,j,k) * \rho(i,j,k)$$

Where ρ is the CT value for the voxel with indices i, j, k , and the weight $l(i, j, k)$ is the normalized length travelled by the ray within the voxel i, j, k . In the parallelized version implemented in this thesis, each ray was computed independently by one thread of the parallel structure.

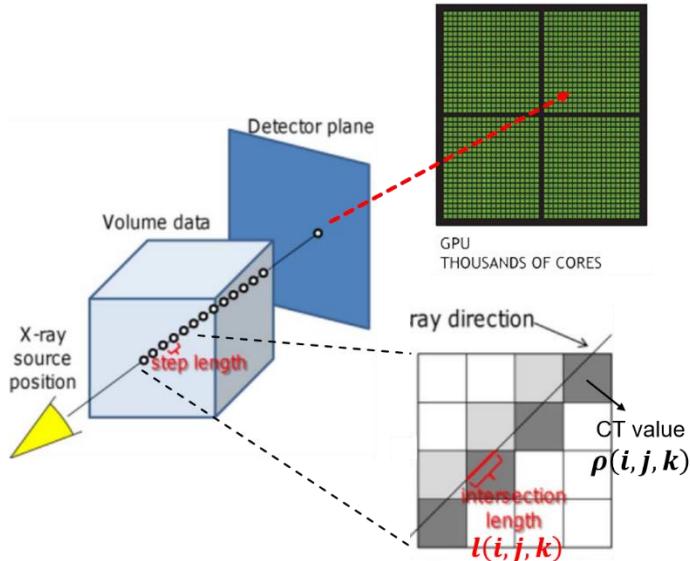


Figure 5.8. Overview of the process for DRR generation from a CT scan and of the approach for parallelization. Readjusted from a Figure from Otake et al. [125].

Parallelization was achieved with CUDA API [133] on Microsoft Visual Studio Community 2015 in a Windows 10 Enterprise environment and a graphic card NVIDIA Quadro M5000 with 8 GB memory. CUDA was introduced by NVidia in 2006, as a general purpose parallel computing architecture. Because CUDA can be programmed with the C programming language, it is easier to use than other GPU program languages, particularly for non-computer-graphics specialists.

The CUDA library for parallel DRR generation was written in C and used in the definition of a C++ class. The C++ class was then wrapped in Python using Cython (<http://cython.org>) in order to be used in the Python module “Projector”.

The programming structure was generated as follows, with terms in italics representing the relevant CUDA application program interface reference [10]:

1. On the host (CPU), linearize CT scan into a 1-dim standard C array (Vectors from the Standard Template Library of C++ are not supported by CUDA).
2. Allocate memory on the device (*cudaMalloc*) for linearized CT scan and for output DRR, both as 1-dim array).
3. Copy host (CPU) memory array of the CT scan to device (GPU) memory (*cudaMemcpy*)
4. Execute code on device (GPU).
5. Copy device (GPU) memory array for the complete output DRR to host (CPU) memory (*cudaMemcpy*).
6. Free memory space (*cudaFree*) and exit device execute.

In the current implementation, a number of 16 threads along each dimension perform the computation of one DRR in parallel. An example of a DRR is shown in Figure 5.14 in the Results section.

DRR from CAD models

For DRR generation from CAD models in stereolithographic format (STL), the rendering tools from the open source software *VTK* [177] and the open source computer vision library *opencv* [178] were used. Simulated images of the prosthetic implant are fast generated as binary images with white background by means of no-shading rendering from VTK Python graphics library. Subsequently they are blurred with Gaussian filter from *opencv* to simulate a real imaging system. The combination of *VTK* and *opencv* allowed fast generation of DRRs in about 15 msec. An example of generated DRR is shown in Figure 5.14b.

The produced images are binary, except for the edges, thus do not reproduce real X-ray casting where the attenuation of each X-ray is proportional to the length travelled within an object. However, in the real fluoroscopic images the pixel values within the area of the implant silhouette are rather homogenous: this is because metal implants have typically high linear attenuation coefficient, resulting in high absorption of X-ray photons at clinical energies. Therefore, quasi-binary images are deemed sufficiently accurate for registration of a metal implant. Furthermore, registration of CAD models was performed in the context of this thesis using the edges of the images, therefore the pixel values within the silhouette area were not relevant.

A proper X-ray casting algorithm for DRR generation from surface models was tested by using the *IntersectWithLine* method of the *vtkOBSTree* *VTK* class, which calculates the entry and exit points of intersection of a ray with the object, thus allowing computation of the exact radiological path length. However, this method resulted too slow for the purpose of image registration, and a parallel version of it was not implemented.

For the analysis of registration accuracy reported in Chapter 3, an alternative method that does not use graphics rendering was implemented (Figure 5.9). Here, the 3D points

corresponding to the vertices of the STL files of the CAD models are projected onto the imaging plane using the camera projection Equation 2.1d and the known intrinsic imaging parameters (focal length, pixel size and coordinates of the principal point). The produced binary image of scattered cloud of projected points undergoes a filling operation so that a plain binary mask is obtained. Finally, Gaussian blur is applied to smooth the edges. This method was not used in the *HipHop* framework, because it was slower than the method using *VTK* and *opencv*. Another reason was that the parameters of the filling operation have to be adjusted depending on the resolution of each individual STL file: in some cases holes due to not perfect filling may occur.

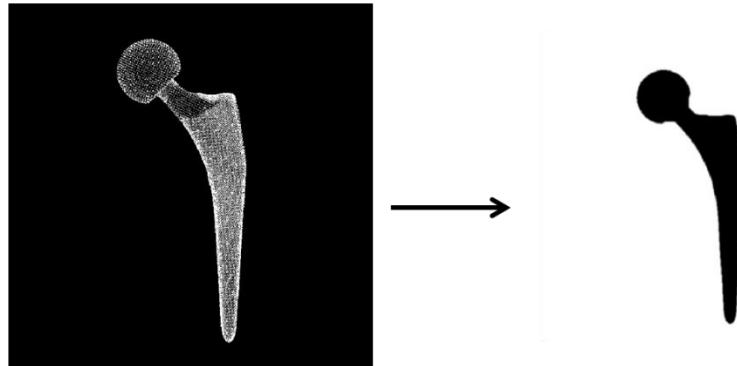


Figure 5.9. The method used in Chapter 3 for generation of a binary digitally reconstructed radiograph (DRR) form a stereolithography (STL) 3D model. The vertices of the STL file are projected onto the image plane, and a filling operation is performed in order to retrieve a binary plain projection. This method was not used for the 2D-3D registration carried out for our in-vivo studies.

5.2.3 Metrics

The “Metrics” module implements different types of similarity measures between the real fluoroscopic image and a DRR. One chosen metric represents the cost function that is maximized by the optimizer in order to retrieve the optimal 3D pose that best aligns the 3D model to the given 2D image.

Several similarity measures were implemented using mostly the functions provided by the ITK library. The implemented similarity measures are listed below. I_1 and I_2 are the two images to be registered (the real fluoroscopic image and a DRR), T is the overlap domain (combined with a region of interest) between I_1 and I_2 , and N is the number of pixels in T .

- *Mean squares difference:*

$$\frac{1}{N} \sum_{(x,y \in T)} (I_1(x,y) - I_2(x,y))^2$$

This similarity metric is mostly used in registration applications where two images form the same modalities (for example X-ray) are used.

- *Mean Reciprocal Squares Difference:*

$$\frac{1}{N} \sum_{(x,y \in T)} \frac{(I_1(x,y) - I_2(x,y))^2}{1 + (I_1(x,y) - I_2(x,y))^2}$$

This metric is a more robust version of the mean squares difference metric. It normalizes the similarity measure to the interval [0,1] and limits the influence of occasional huge differences between the intensities.

- *Normalized cross correlation (NCC):*

$$\begin{aligned} NCC(I_1, I_2) &= \frac{1}{\sigma_1 \sigma_2} \frac{1}{N} \sum_{(x,y \in T)} (I_1(x,y) - \bar{I}_1) \cdot (I_2(x,y) - \bar{I}_2) = \\ &= \frac{\sum_{(x,y \in T)} (I_1(x,y) - \bar{I}_1) \cdot (I_2(x,y) - \bar{I}_2)}{\sqrt{\sum_{(x,y \in T)} (I_1(x,y) - \bar{I}_1)^2} \cdot \sqrt{\sum_{(x,y \in T)} (I_2(x,y) - \bar{I}_2)^2}} \end{aligned}$$

Being \bar{I}_1 and \bar{I}_2 the mean intensity values in the images I_1 and I_2 , respectively. This metric computes the correlation between pixels in the two images. If one

image is linearly dependent on the other one, e.g. $\forall x,y \in T: I_2(x,y) = aI_1(x,y) + b$, a value of 1 would be achieved (and -1 accordingly, if a is negative). In practice, that means that both different contrast and brightness values in the images should not affect the similarity measure (except for differences due to rounded intensity values), which is desirable as it increases metric robustness [179]. Like “mean squares difference”, this similarity metric is mainly limited to images of the same modality.

- *Mutual Information:*

$$H(I_1) + H(I_2) - H(I_1, I_2) = \sum_{a,b} p(a,b) \cdot \log \frac{p(a,b)}{p(a)p(b)}$$

Where $H(I_1, I_2)$ is the joint entropy between the two images, and $p(a,b)$ is the joint probability distribution function of the pixel intensities a and b of the two images, respectively. The joint probability distribution function can be visualized as a two dimensional joint histogram: every value of $p(a,b)$ tells the occurrence of the intensity a in the first and intensity b in the second image at the same positions, respectively. For the theoretical details we refer to the works by Mattes et al. [180], [181] and by Viola et al. [182], which were used for two different implementations in the ITK library. This metric quantifies in a probabilistic way the amount of information carried by one image about the other image. It assumes no functional dependency between the two images, only statistical dependence between the intensities. It has become very popular in registration problems that involve different modalities. One drawback is that it does not take into account of any spatial information (e.g. image location of the intensity values).

- *Gradient Correlation:*

By using horizontal and vertical Sobel templates, four gradient images $\frac{dI_1}{dx}, \frac{dI_2}{dx}, \frac{dI_1}{dy}, \frac{dI_2}{dy}$, are created (Figure 5.10). Then normalized cross correlation NCC (see above) is calculated of these horizontal and vertical gradient images, respectively. The final value of the measure is the average of the two normalized cross correlation values:

$$\frac{NCC\left(\frac{dI_1}{dx}, \frac{dI_2}{dx}\right) + NCC\left(\frac{dI_1}{dy}, \frac{dI_2}{dy}\right)}{2}$$

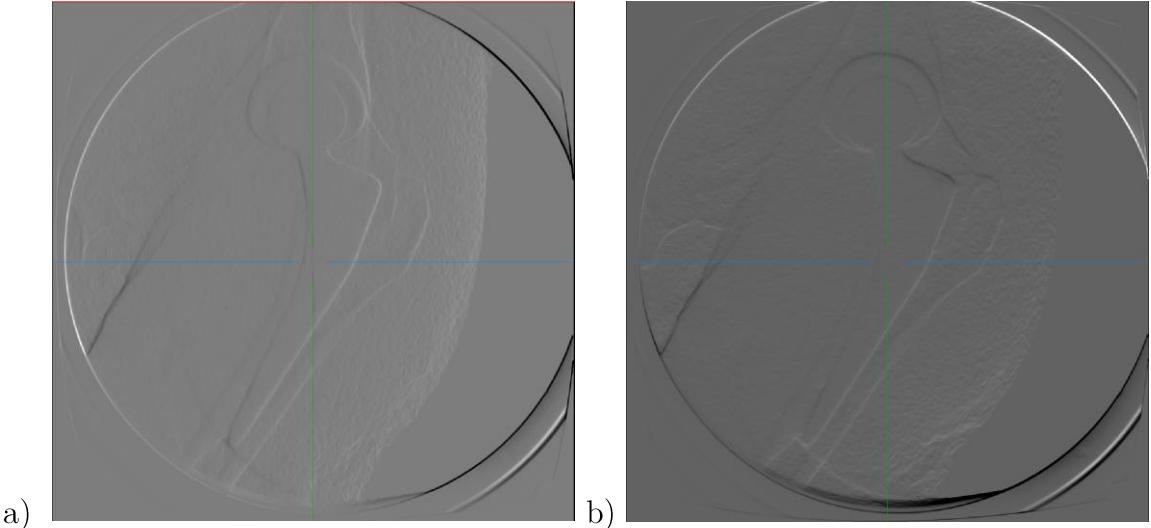


Figure 5.10. Example of horizontal (a) and vertical (b) derivative images, obtained with Sobel filters for computation of the Gradient Correlation similarity metric.

- *Mahfouz*:

A metric similar to the one proposed by Mahfouz et al. [86] and described in the Introduction Chapter was also implemented. This metric consists in a weighted combination between an intensity-based metric IM and a contour-based metric EM . IM is similar to cross correlation of the simulated and the real X-ray image (Figure 5.11 a and b), and is a measure of the amount of overlap for both images

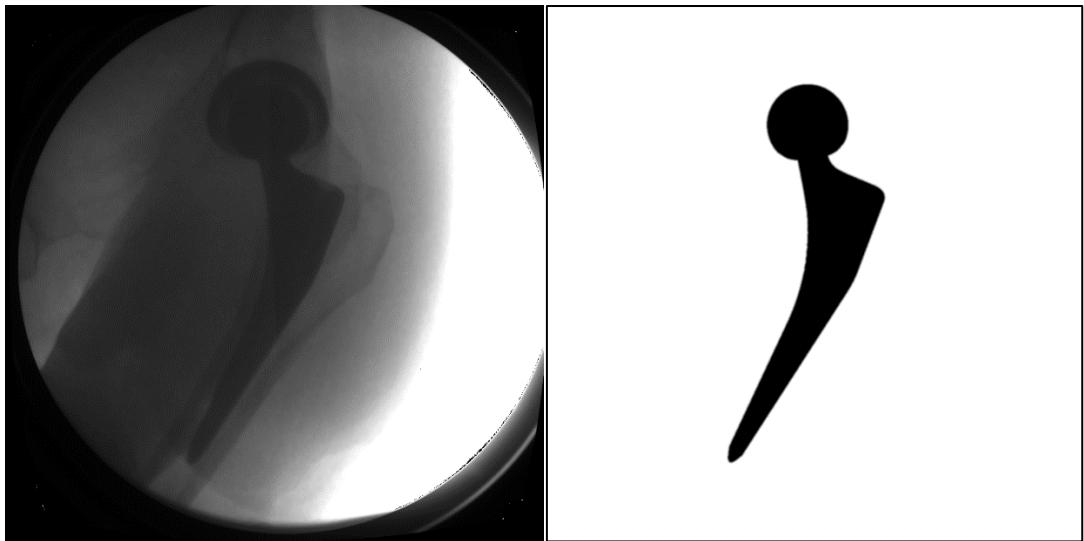
between the darker areas relative to the projected implant. EM is similar to cross correlation of the corresponding edge images dI_1, dI_2 (Figure 5.11 c and d) obtained by Canny edge detection [183] followed by Gaussian blur.

$$w_{int} * IM + w_{edge} * EM$$

$$IM = \frac{\sum_{(x,y \in T)} I_1(x,y) \cdot I_2(x,y)}{\sum_{(x,y \in T)} I_1(x,y)}$$

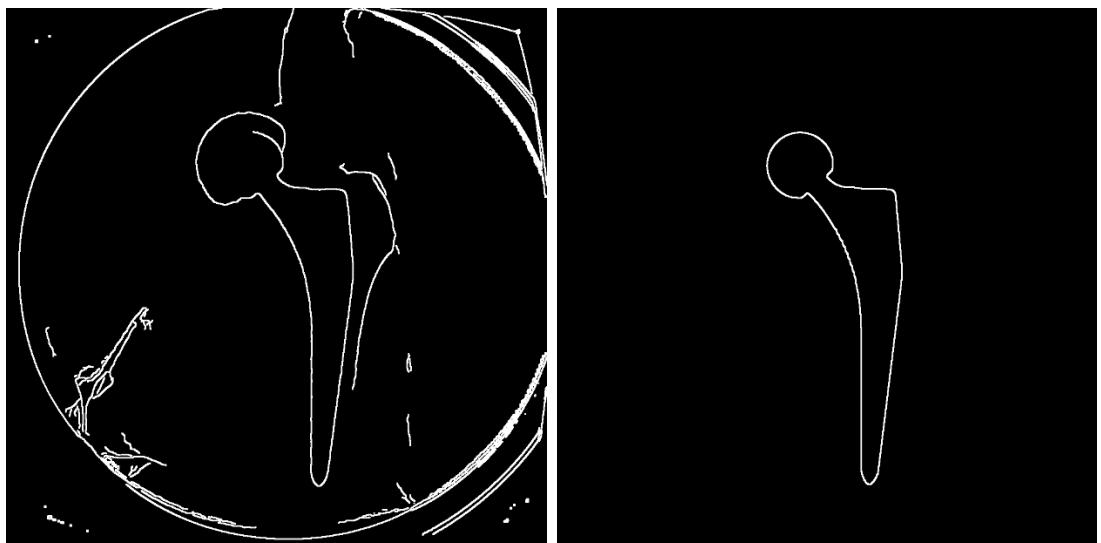
$$EM = \frac{\sum_{(x,y \in T)} dI_1(x,y) \cdot dI_2(x,y)}{\sum_{(x,y \in T)} dI_1(x,y)}$$

The method relies on the combined effect of the two metrics on the similarity cost function: the contour-based metric provides accurate minimum due to its specificity, while the intensity-based metric provides a large basin of attraction around the minimum to facilitate the optimizer to find the correct solution. The weights w_{int} and w_{edge} are determined experimentally to achieve good results for typical images. The weights used for our registration were typically set to the recommended ones by Mahfouz et al., that is $w_{int} = 1, w_{edge} = 2.67$. The main drawback of this similarity metric is that the parameters for Canny edge detection shall be determined for each image, due to the large difference in contrast between images. Consequently, more user interaction and thus more computational time is needed.



a) Real intensity image

b) Simulated intensity image



c) Real edge image

d) Simulated edge image

Figure 5.11. The similarity score from the Mahfouz metric is obtained from comparison of both intensity images (a and b) and edge images (c and d).

5.2.4 Optimizer

The “Optimizer” module includes one local and one global optimization methods to find the optimal 3D pose, both obtained from the free and open source “NLOpt” library [184]. Both methods are heuristic and do not compute the derivatives of the cost function at each optimization step, as gradient-descent approaches do. Derivative-free methods were chosen since computation of the derivatives with finite differences requires at least the generation of a double amount of DRRs for a forward approach, thus the computational time is doubled.

The local optimization method is the popular Nelder-Mead simplex algorithm, based on the original version proposed by Nelder and Mead [185].

The global optimization method is an Evolutionary Strategy algorithm based on the work of Silva Santos et al. [186] and adapted from ideas described in [187], [188].

The global optimizer was the choice for our study, since previous studies have pointed out the inefficiency of local optimization methods for 2D-3D registration. Intensity-based similarity measures are not convex and not monotone, and present many local minima besides the global optimum. Local optimizers get stuck to these numerous local minima, and their convergence to the global optimum is more dependent on the initial guess compared to global optimizers.

5.3 Results

5.3.1 Pre-processing

The accuracy of our initialization procedure was evaluated by comparing the ground truth registered poses obtained using *HipHop* with the motion capture initialized poses,

for 4 patients during gait, stair descent and putting on socks (Figure 5.12). Mean absolute errors for the rotations were below 3° , and for translations were below 7 mm.

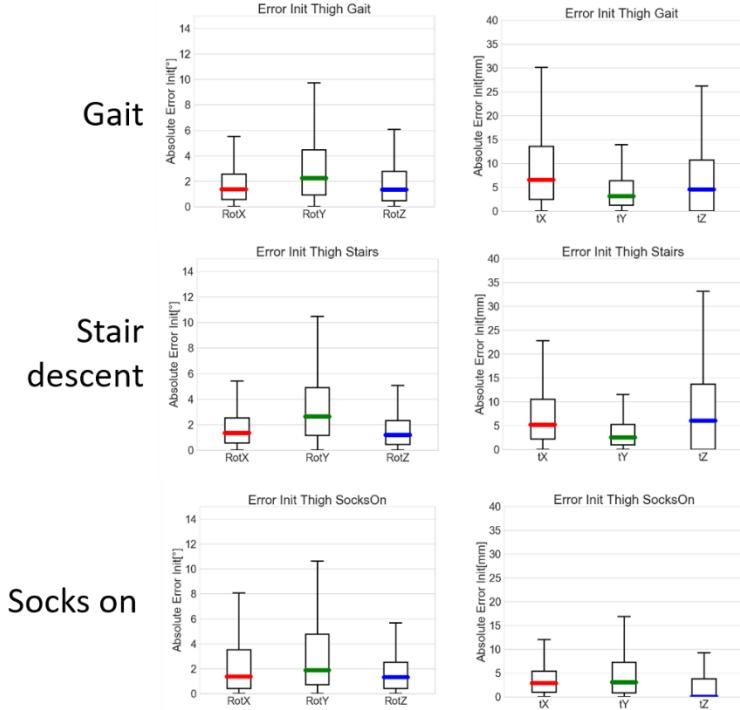


Figure 5.12. Accuracy of the skin marker-based pose initialization of the femoral stem, for 4 THA patients and three motion activities. Error values were computed as the difference between the ground truth poses (registered with “HipHop” and subsequently manually refined) and the initialized poses. Each pose in the lab coordinate system is described by three rotations (RotX , RotY , RotZ) and three translations (tX , tY , tZ).

5.3.2 Projector

An overview example of generation of a digitally reconstructed radiograph from a CT scan of a patient with a hip implant is shown in Figure 5.13.

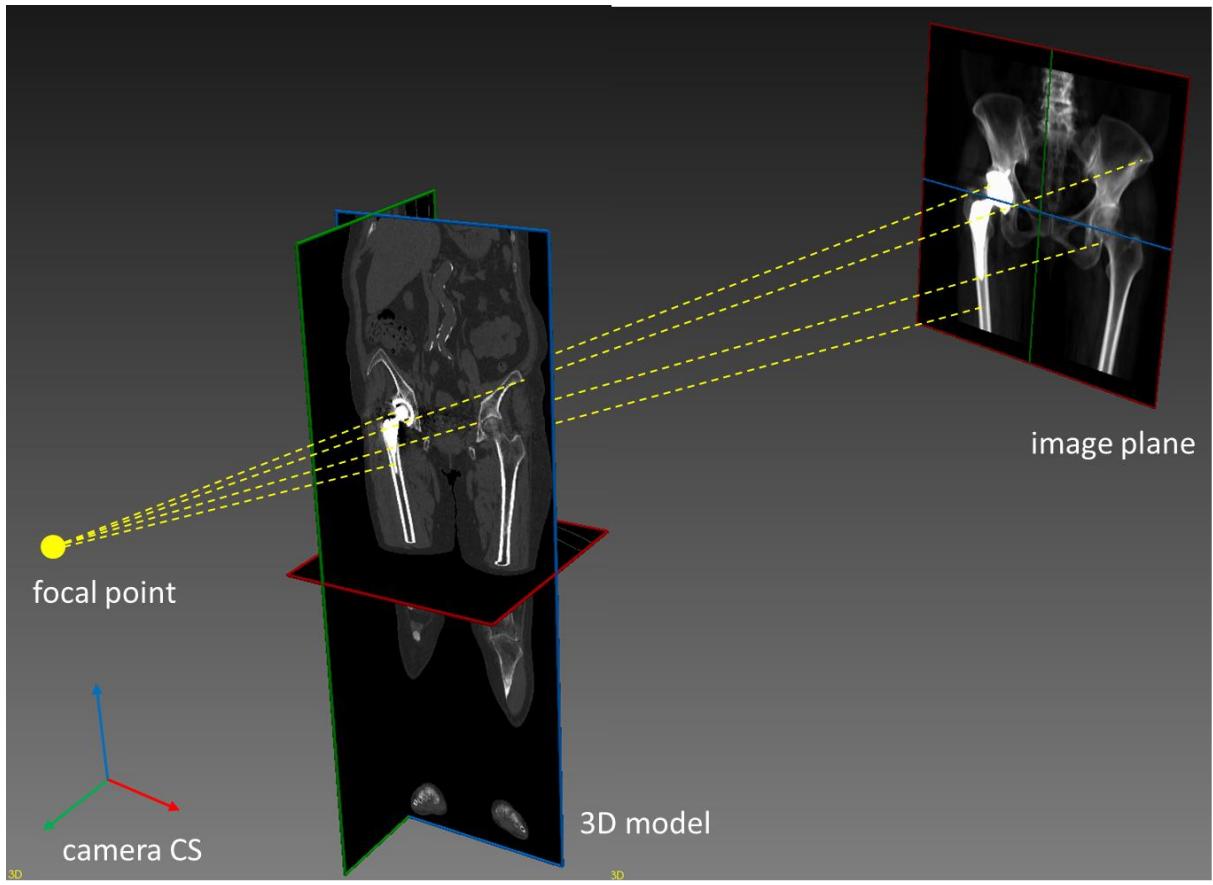
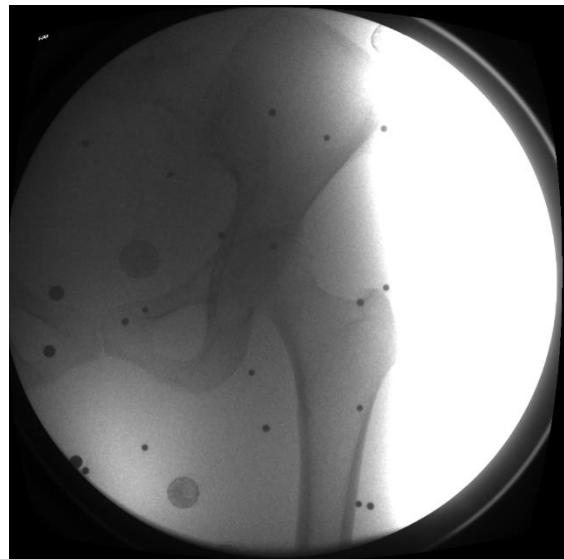


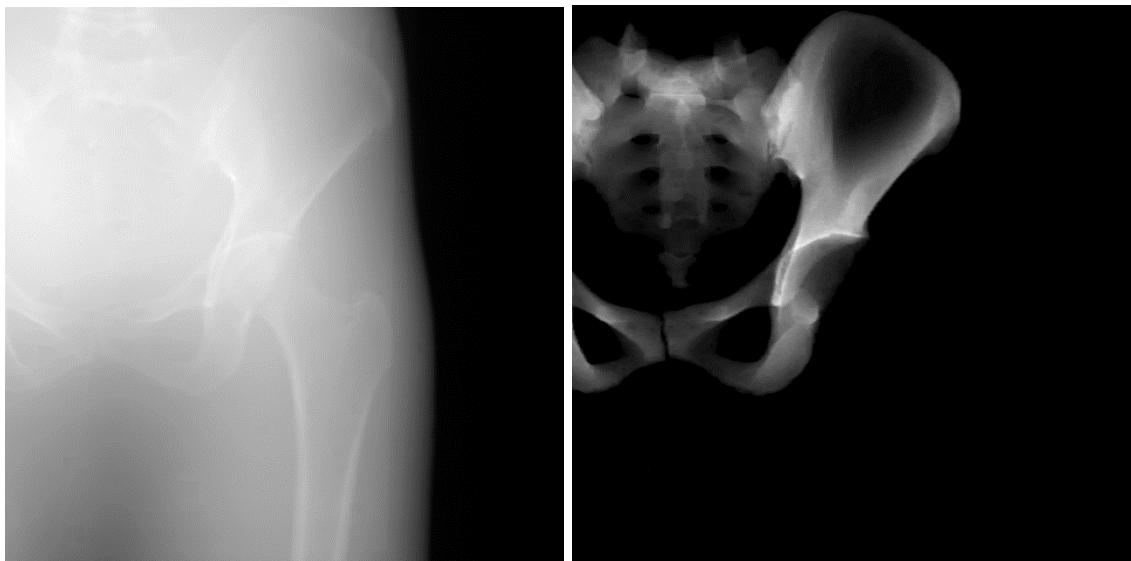
Figure 5.13. Overview example of generation of a digitally reconstructed radiograph (DRR) from a CT scan of a patient with a hip prosthetic implant.

A detailed example of a digitally reconstructed radiograph (DRR) of a hip phantom is shown in Figure 5.14 b and c. The DRR of size 1000x1000 was generated from a CT scan of size 415x200x150 with GPU in 80 msec, compared to 7 sec required when using the Ray Cast Interpolator function of the ITK library.
[\(https://itk.org/Doxygen46/html/Filtering_2DigitallyReconstructedRadiograph1_8cxx-example.html\)](https://itk.org/Doxygen46/html/Filtering_2DigitallyReconstructedRadiograph1_8cxx-example.html)

An example of binary DRR from a CAD model produced using OpenCV is shown in Figure 5.15 b. The required computational time was 15 msec.



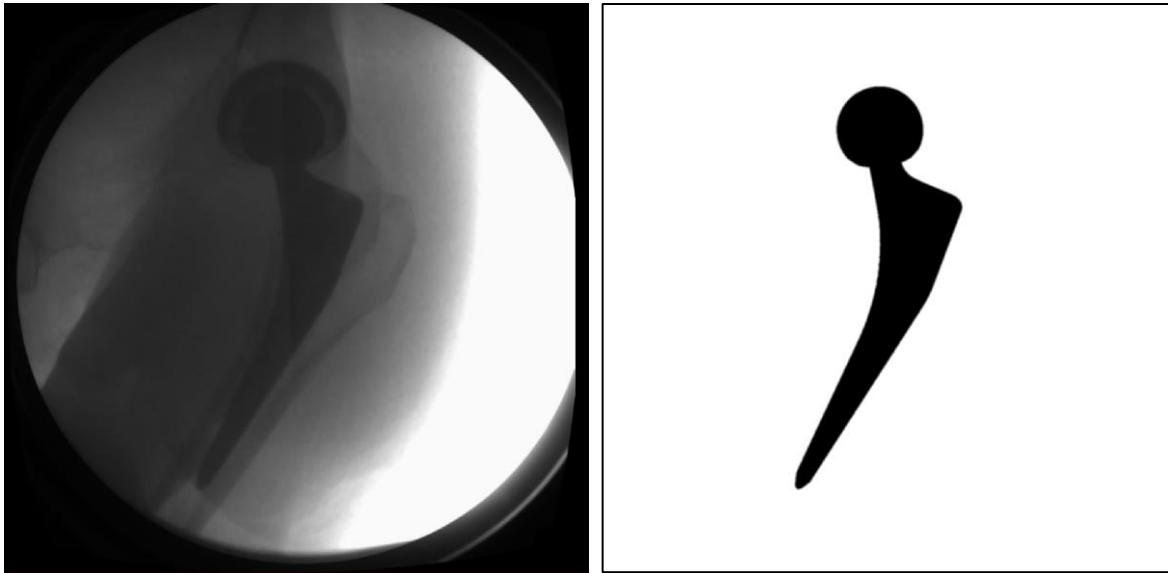
a) Real image of a hip phantom



b) DRR from CT scan of the hip phantom

c) DRR from segmented bone only

Figure 5.14. a) A real fluoroscopic image of a hip phantom b) A 1000x1000 digitally reconstructed radiograph (DRR) of the hip phantom generated with GPU in 80 msec from a CT scan of size 415x200x150. c) A DRR generated from the pelvic bone model only segmented from the CT scan.



a) Real in-vivo image of a hip implant **b)** DRR from CAD model of the hip implant

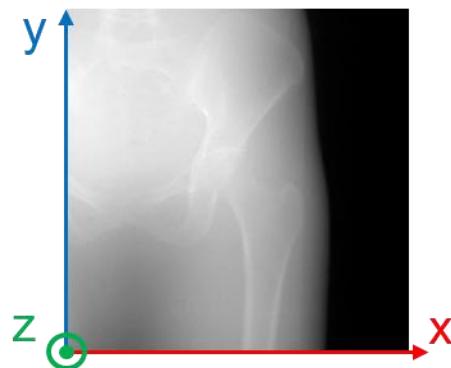
Figure 5.15. a) A real in-vivo fluoroscopic image of a hip implant. b) Corresponding binary DRR of the femoral stem generated from a CAD model with OpenCV/OpenGL in 15 msec.

5.3.3 Metrics

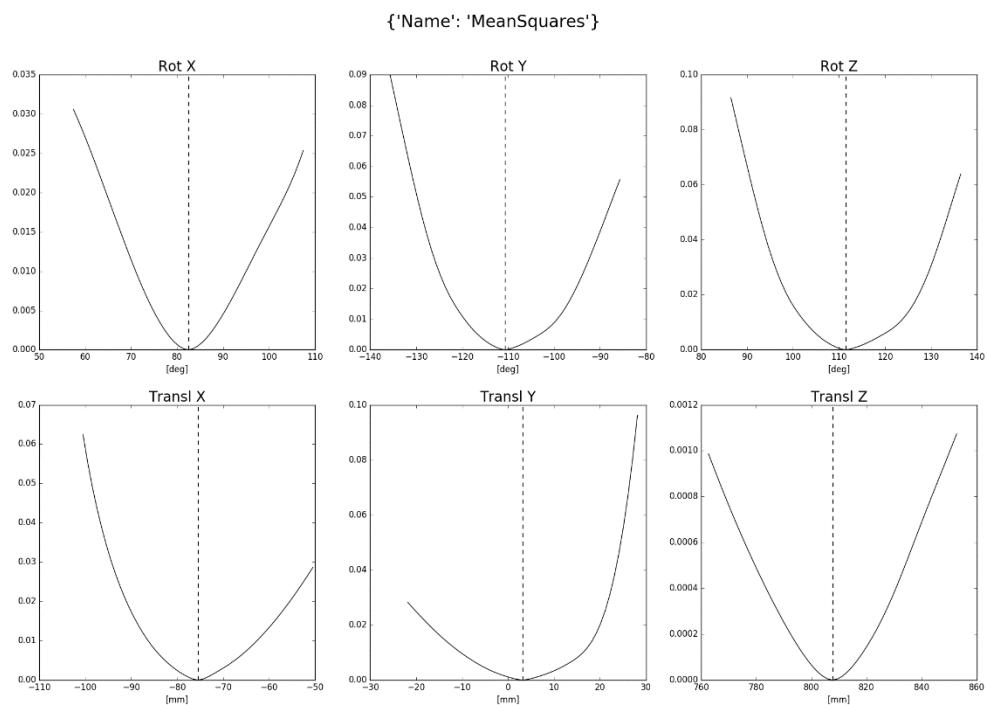
In order to check the correctness of each similarity metric, to compare the performance between different metrics and to identify the best one for our specific application, a preliminary study was carried out using simulated images as ground truth. A DRR at known position was generated and used as if it were a real fluoroscopic image. Each similarity measure was evaluated between this ground truth DRR and DRRs generated at 3D poses within a certain range from the ground truth pose. The tested DRRs were generated by modifying one pose parameter at time within a $\pm 25^\circ$ range from the ground truth pose for rotations, and within a ± 25 mm from the ground truth pose for translations. For each similarity metric, 6 one-dimensional landscapes of the similarity function were finally produced around the ground truth pose (Figure 5.16).

The results showed in Figure 5.16 for a pelvis phantom proved that all similarity metrics were implemented correctly, since the minimum/maximum along each dimension

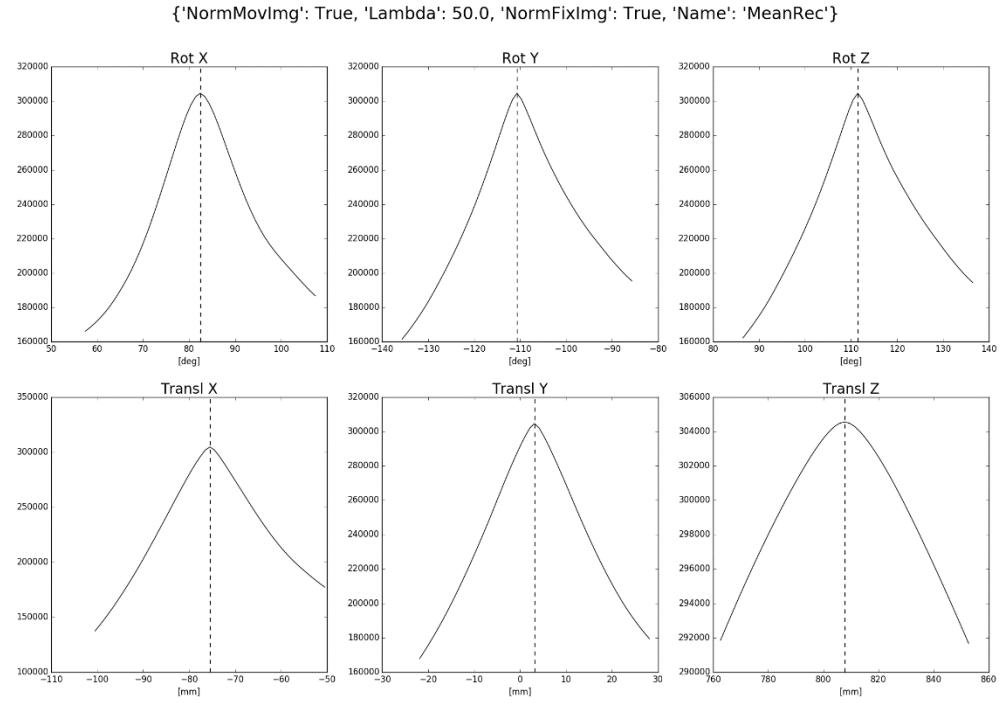
occurred exactly at the ground truth position, for which the DRR is pixel wise identical to the ground truth DRR. The attraction basin around the global minimum was wider for the Mean Squares Difference, the Mean Reciprocal Squares Difference and the Normalized Cross Correlation metrics, and narrower for the Mutual Information and Gradient Correlation metrics.



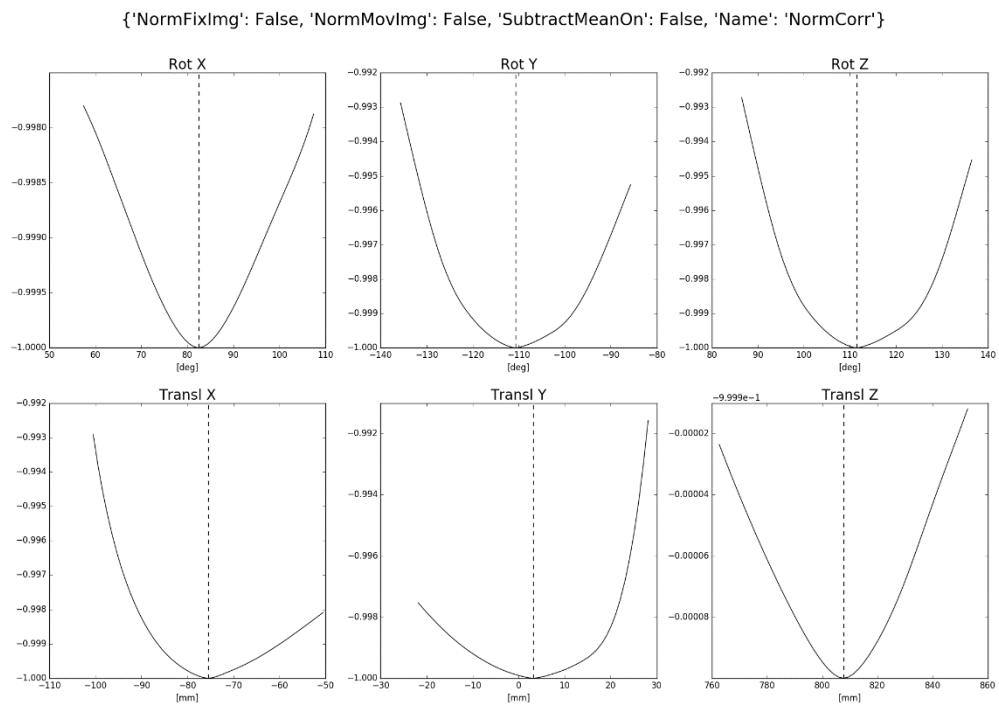
a) Mean squares difference



b) Mean Reciprocal Squares Difference

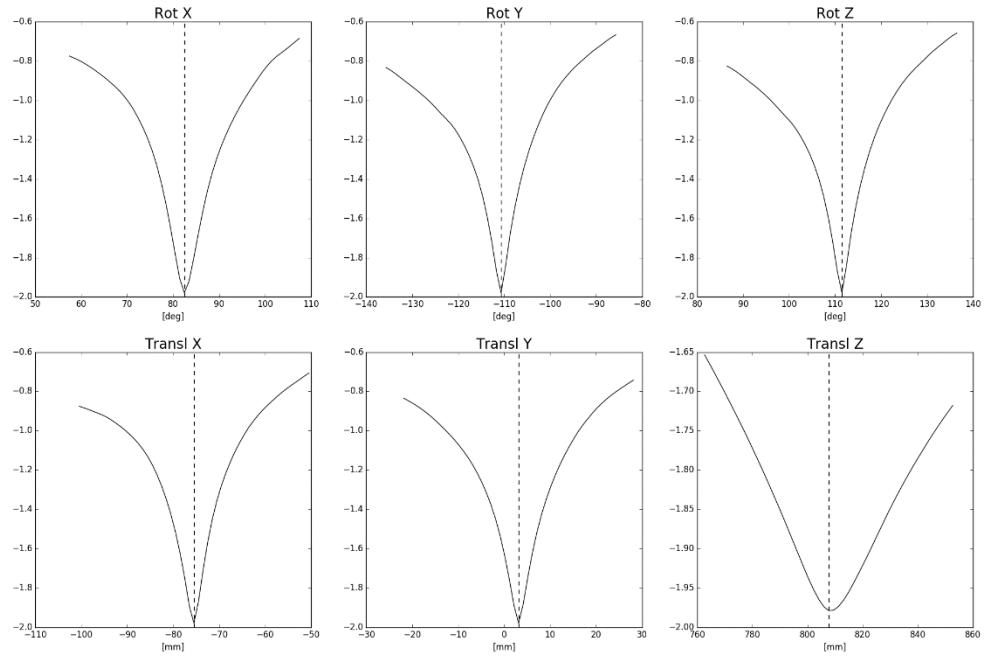


c) Normalized cross correlation



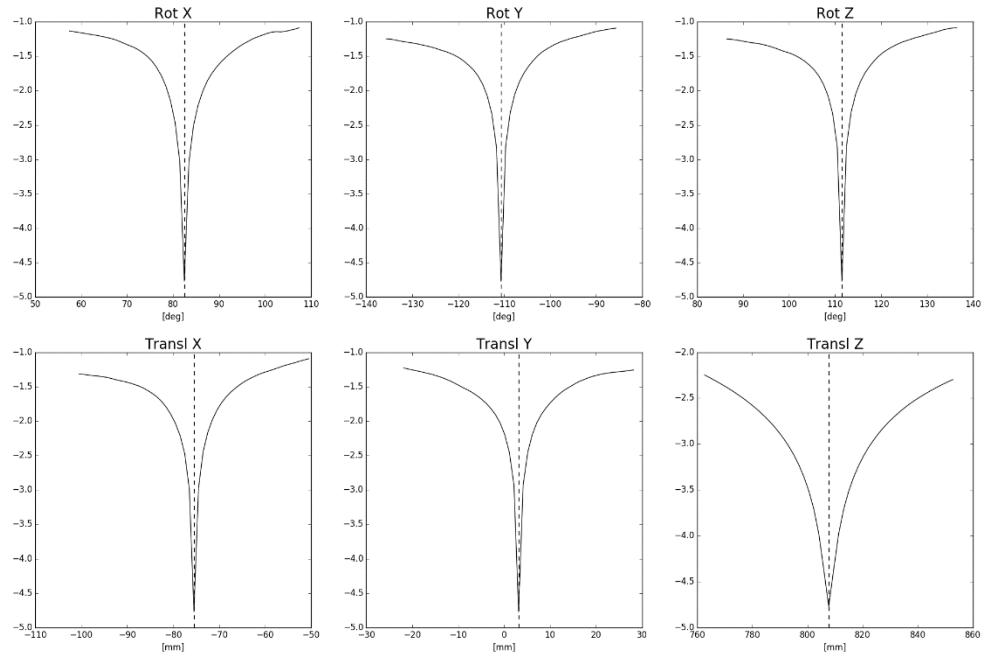
d) Mutual Information (64 histogram bins)

{'NumSamples': [], 'NormMovImg': False, 'AllPixels': True, 'Name': 'MattesMutualv3', 'NumHistBins': 64, 'NormFixImg': False}



e) Mutual Information (1024 histogram bins)

{'NumSamples': [], 'NormMovImg': False, 'AllPixels': True, 'Name': 'MattesMutualv3', 'NumHistBins': 1024, 'NormFixImg': False}



f) Gradient Correlation

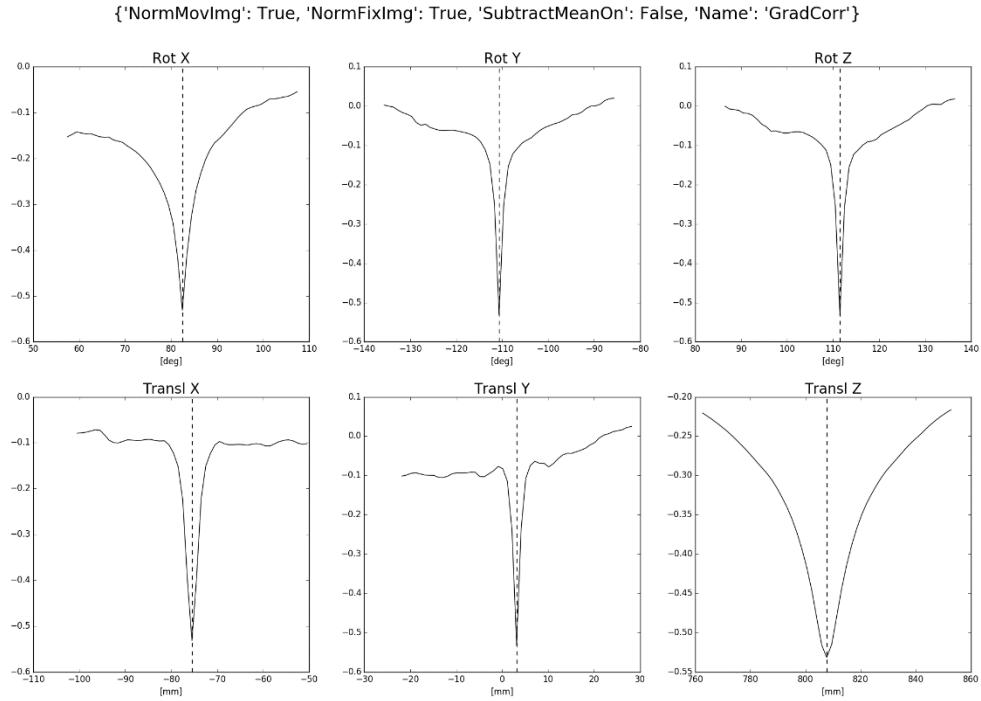


Figure 5.16. One-dimensional landscapes of the similarity functions between a ground truth DRR and DRRs generated within a range from the ground truth pose. This analysis was conducted for a CT scan of a pelvis phantom. A vertical dotted line indicates the ground truth pose. All similarity metrics showed a global minimum at the ground truth pose, for which the DRR is pixel wise identical to the ground truth DRR. This confirmed the correctness of the implemented similarity metrics. The Mutual Information metric was tested for two different values for the numbers of bins of the image histograms computed to obtain the joint probability distribution function: the larger the number of histogram bins, the narrower the attraction basin around the minimum.

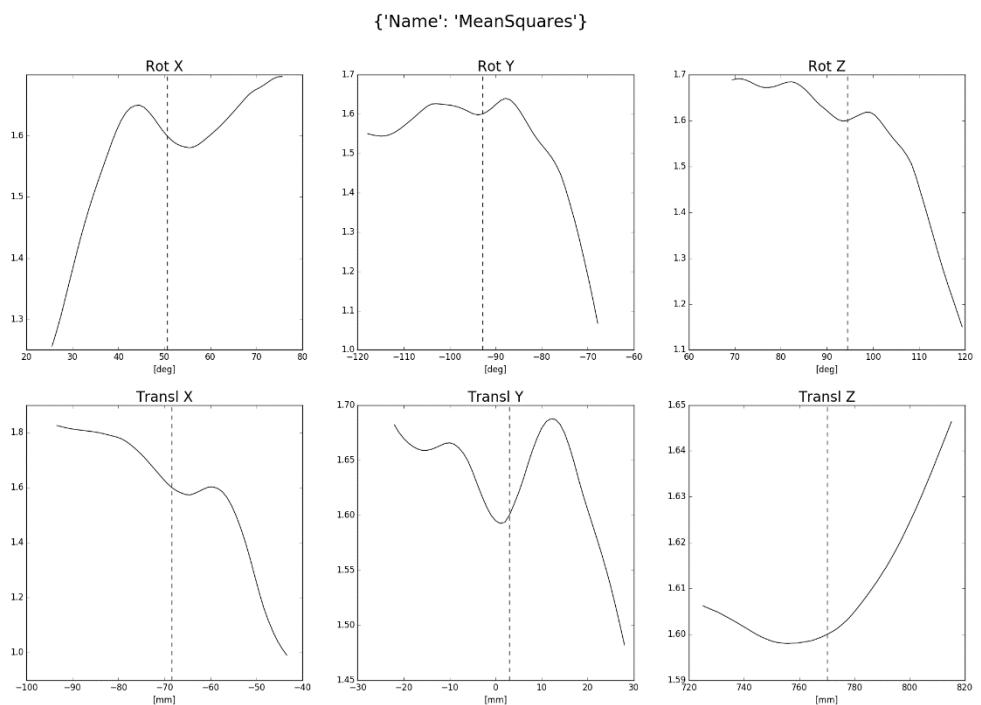
After the preliminary study on simulated DRRs, the performance of the similarity metrics was assessed on real fluoroscopic images of both a hip phantom (Figure 5.17) and an implanted hip prosthesis (Figure 5.18). For a fluoroscopic image of a hip phantom, only Gradient Correlation presented a global minimum at the ground truth pose for all rotational and translational pose parameters (Figure 5.17 f). The other metrics failed at showing the highest similarity between real image and DRR at the ground truth pose (Figure 5.17 a - e). For the in-vivo image of a hip prosthesis, similar good performance for all pose parameters was observed for Mutual Information, Gradient Correlation and the Mahfouz metric with $w_{edge} > 0.01$ (Figure 5.18 c, d, h,i). Normalized Cross

Correlation presented a global minimum at the ground truth pose only for the x - and y -translations (Figure 5.18 b).

Regarding the Mahfouz metric, it is interesting to notice the effect of an increasing edge-based matching score w_{edge} to the overall similarity measure: when w_{edge} is null the global minimum at the ground truth pose is not captured for most of the pose parameters (Figure 5.18 e); an increase of w_{edge} from 0.005 to 2.67 produces a gradually narrower and steeper attraction basin around the global minimum at the ground truth pose (Figure 5.18 f - i). The shape of the attraction basin significantly affects the performance of the optimizer in converging to the global minimum.

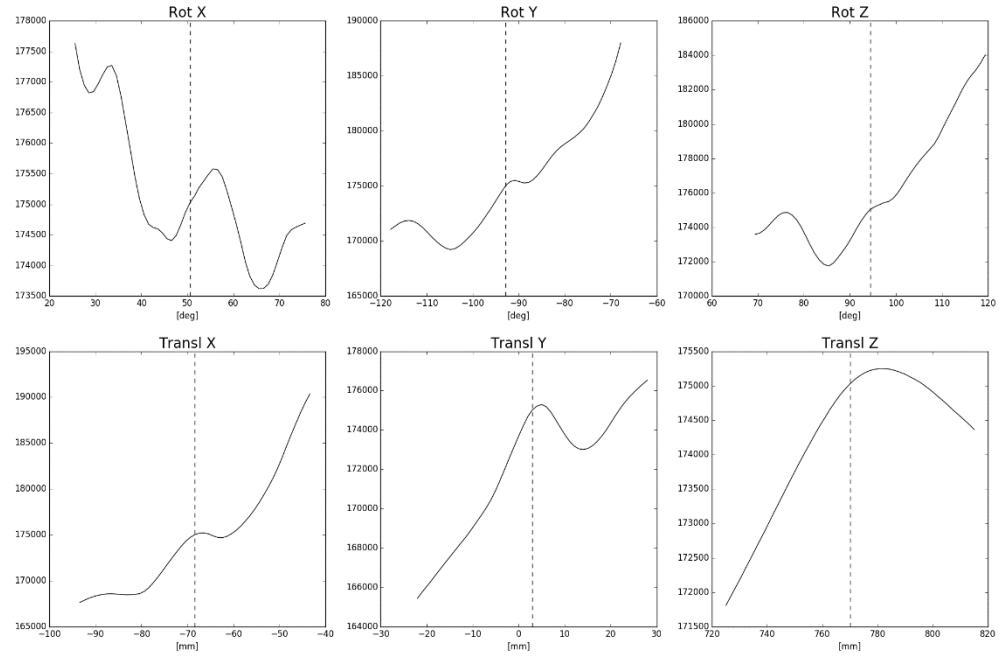


a) Mean squares difference



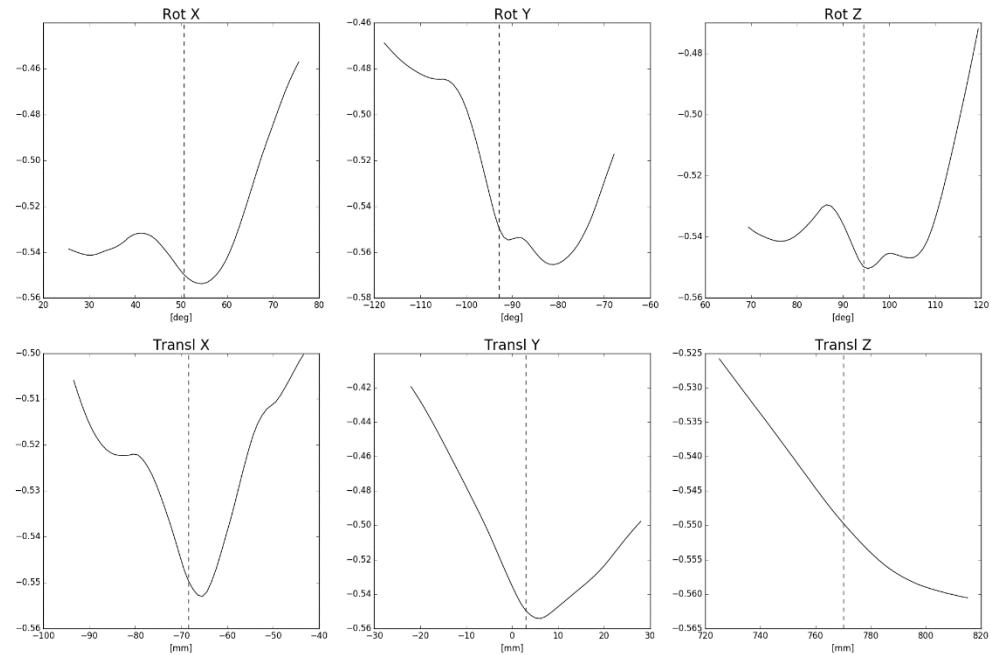
b) Mean Reciprocal Squares Difference

{'Name': 'MeanRec', 'NormFixImg': True, 'Lambda': 1.0, 'NormMovImg': True}



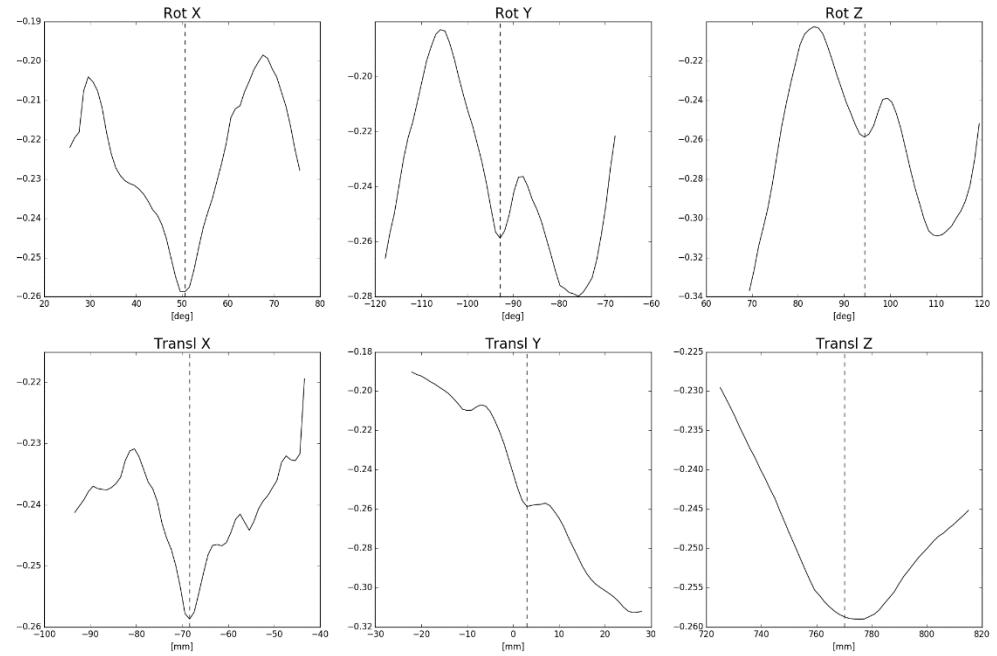
c) Normalized cross correlation

{'NormFixImg': True, 'SubtractMeanOn': False, 'NormMovImg': True, 'Name': 'NormCorr'}



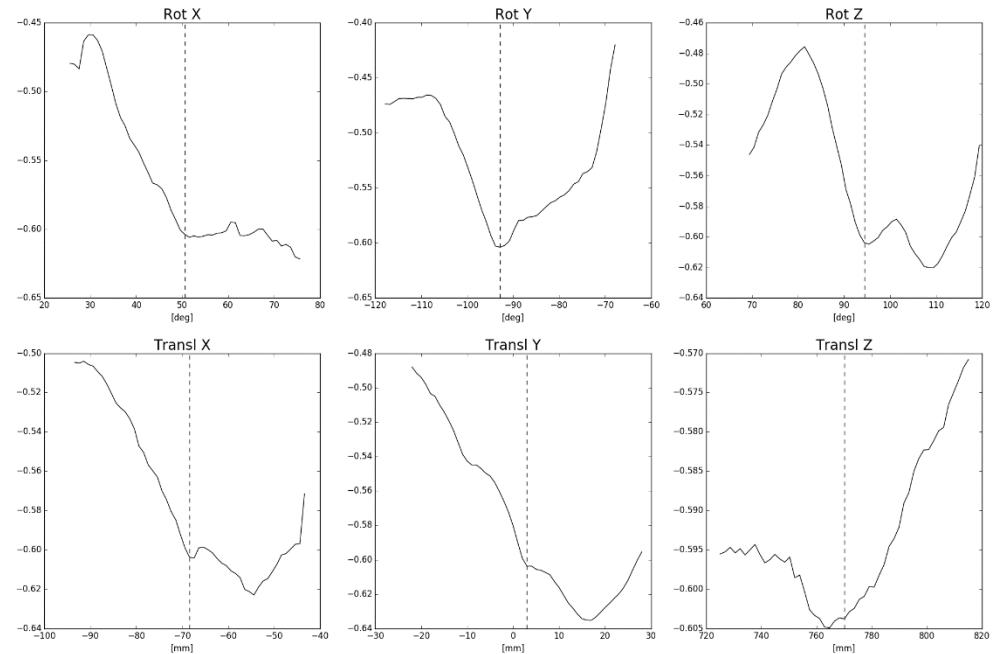
d) Mutual Information (64 histogram bins)

```
{'AllPixels': True, 'NumHistBins': 64, 'NormFixImg': False, 'NumSamples': [], 'NormMovImg': False, 'Name': 'MattesMutualv3'}
```



e) Mutual Information (1024 histogram bins)

```
{'AllPixels': True, 'NumHistBins': 1024, 'NormFixImg': False, 'NumSamples': [], 'NormMovImg': False, 'Name': 'MattesMutualv3'}
```



f) Gradient Correlation

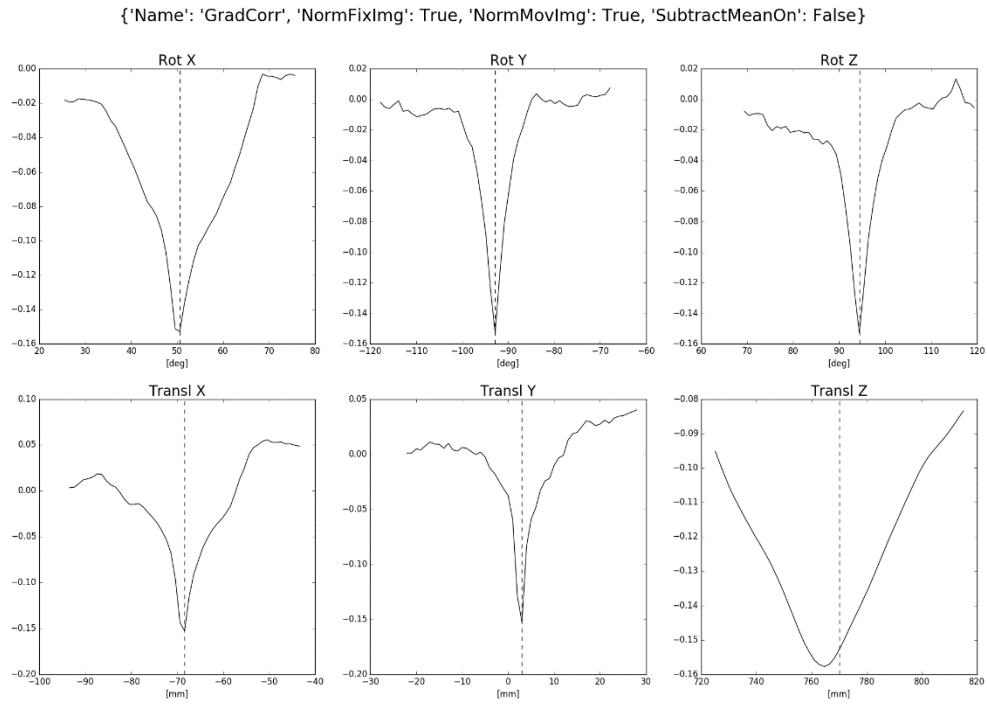
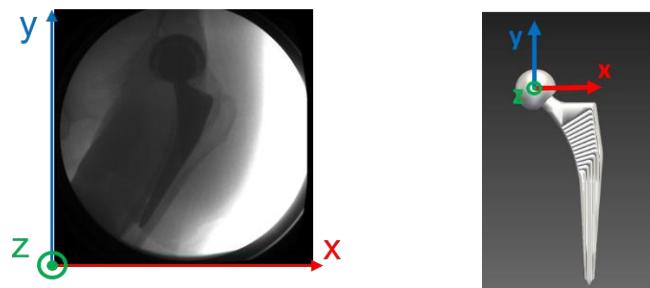
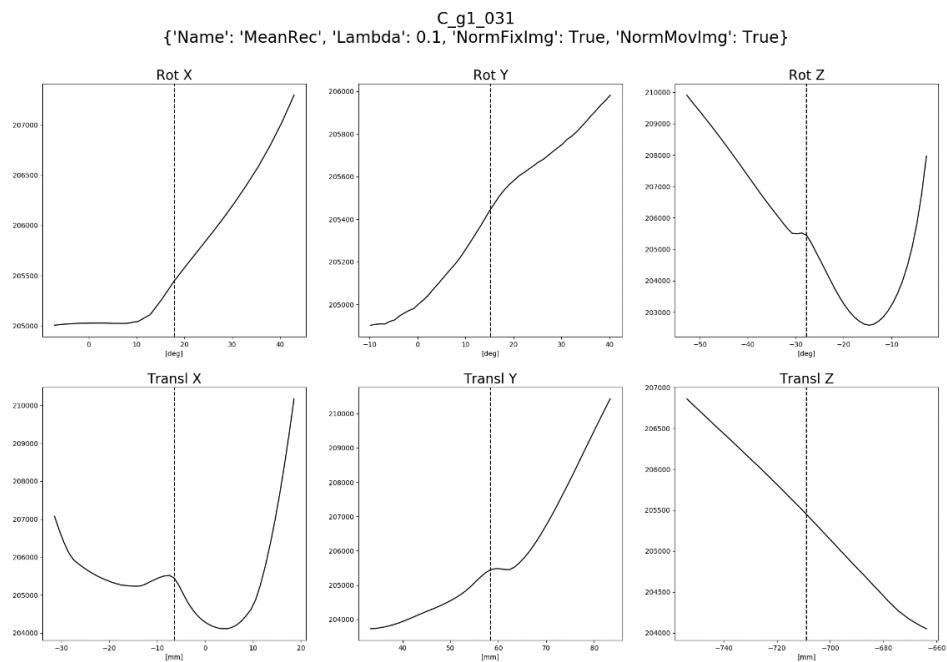


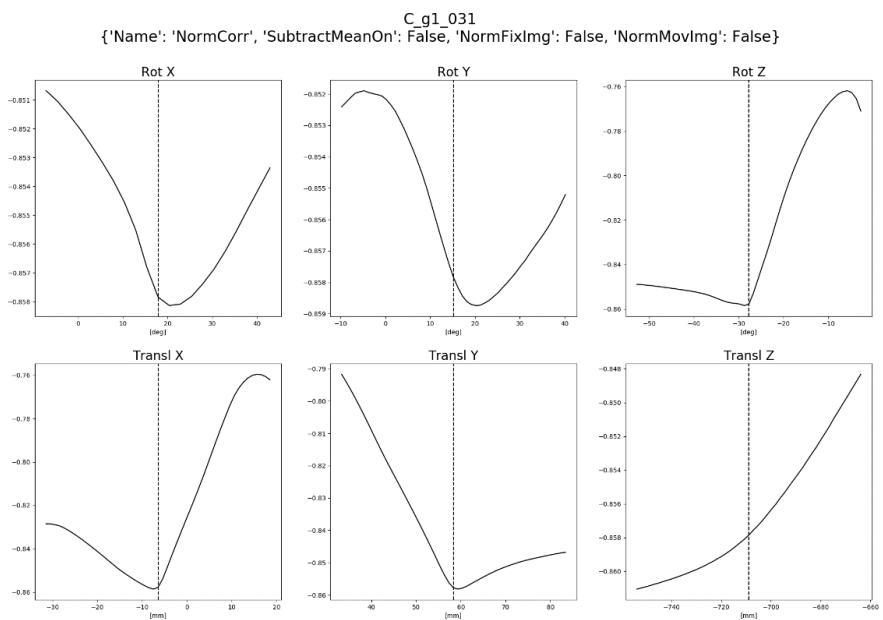
Figure 5.17. Assessment of the performance of the implemented similarity metrics on one real fluoroscopic image of a hip phantom and relative ground truth extracted from the gold standard dataset presented in Chapter 5. A vertical dotted line indicates the ground truth pose. While all similarity metrics showed good performance for the simulated image (Figure 5.16), only Gradient Correlation presented the global minimum at the ground truth pose for all rotational and translational parameters.



a) Mean Reciprocal Squares Difference

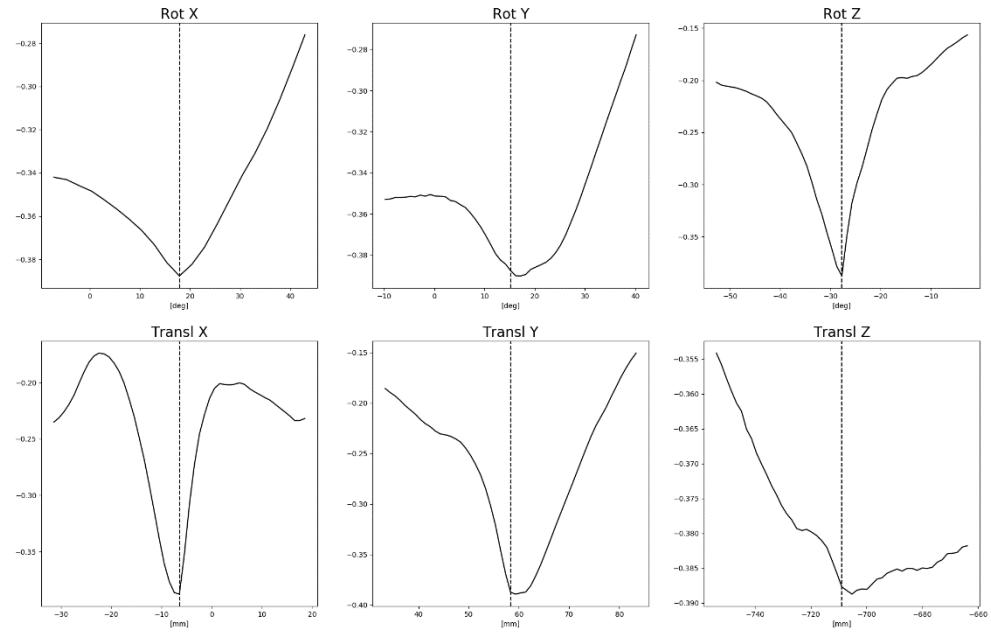


b) Normalized cross correlation



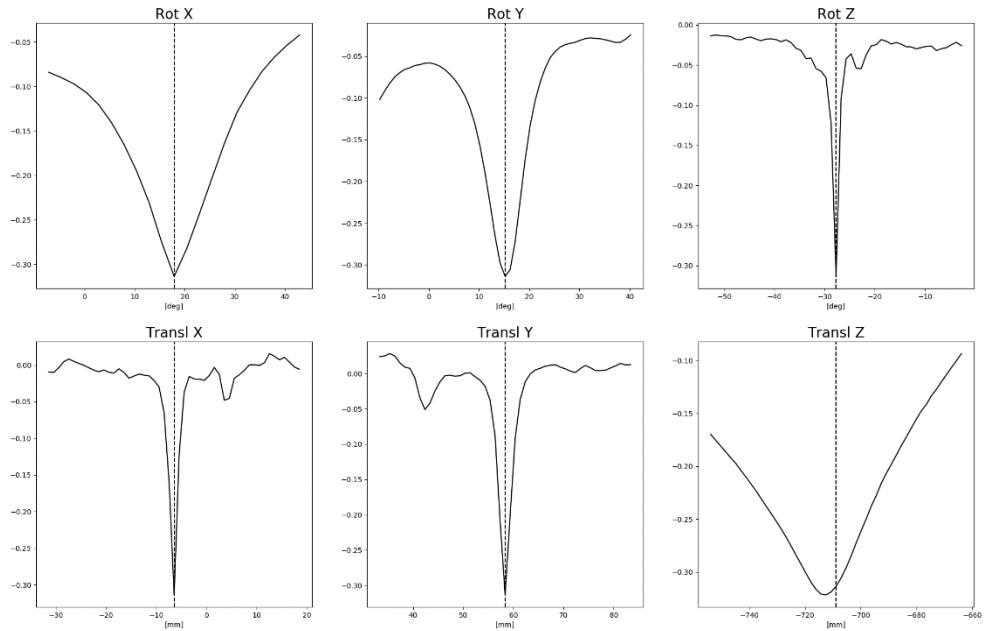
c) Mutual Information (256 histogram bins)

C_g1_031
 {'Name': 'MattesMutualv4', 'NumHistBins': 256, 'AllPixels': False, 'NormMovImg': False, 'NormFixImg': False}



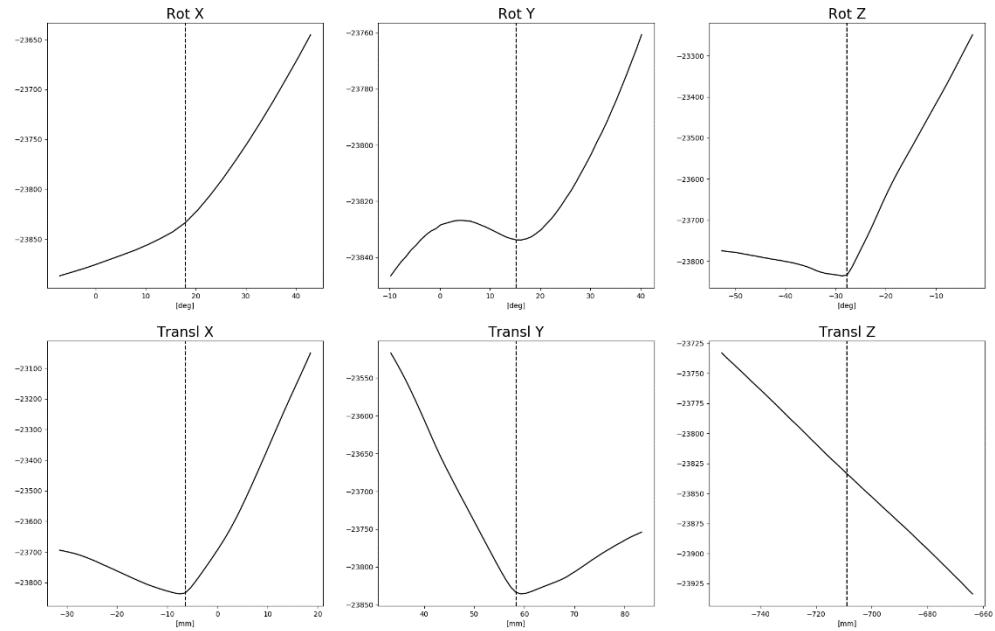
d) Gradient Correlation

C_g1_031
 {'Name': 'GradCorr', 'NormFixImg': False, 'NormMovImg': False, 'SubtractMeanOn': False}



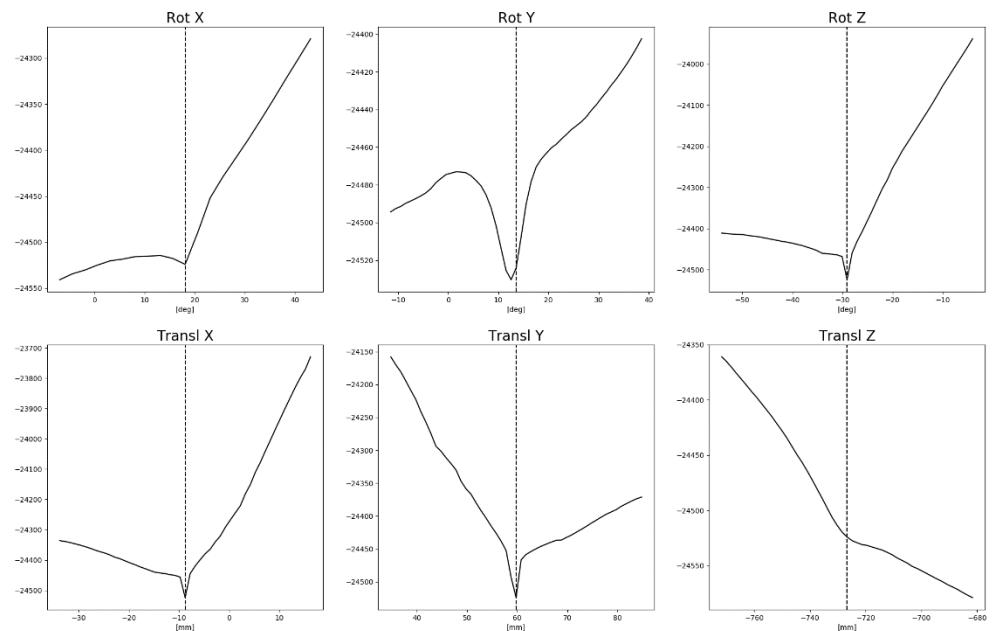
e) Mahfouz ($w_{int} = 1, w_{edge} = 0$)

C_g1_031
 {'edgeGsigma': 1, 'edgeGsize': [3, 3], 'MovCannyThr1': 100, 'MovCannyThr2': 200, 'FixCannyThr1': 1299, 'FixCannyThr2': 1775, 'FixCannyAperture': 7, 'intWeight': 1.0, 'edgeWeight': 0.0, 'saveFixedImages': False, 'useMask': False, 'saveMovImages': True}



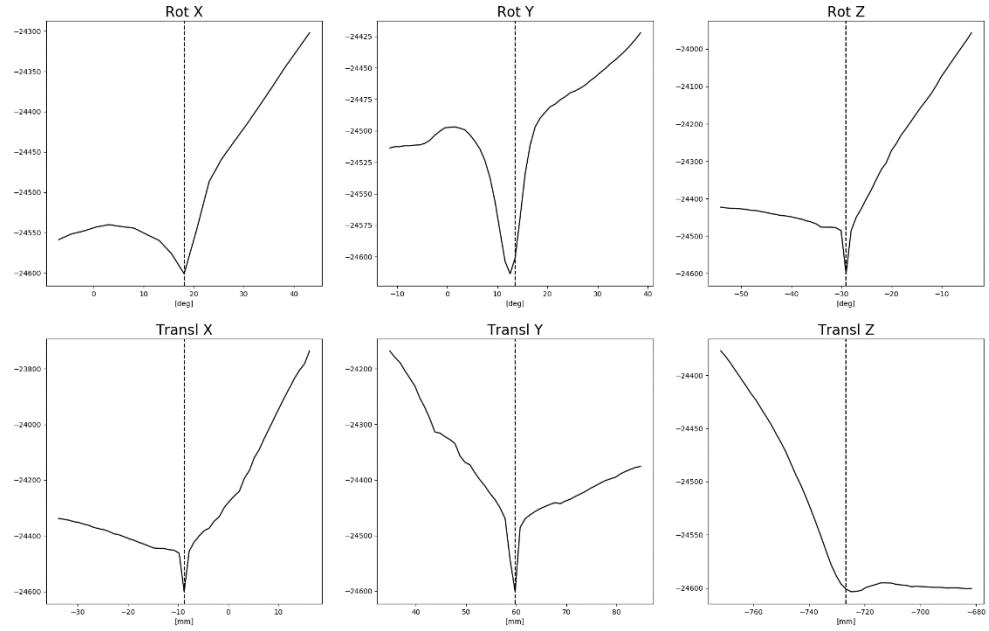
f) Mahfouz ($w_{int} = 1, w_{edge} = 0.005$)

C_g1_032
 {'Name': 'MahfouzHipHop', 'edgeGsigma': 1, 'edgeGsize': [3, 3], 'MovCannyThr1': 100, 'MovCannyThr2': 200, 'intWeight': 1.0, 'edgeWeight': 0.005, 'useMask': False, 'saveFixedImages': True, 'saveMovImages': False}



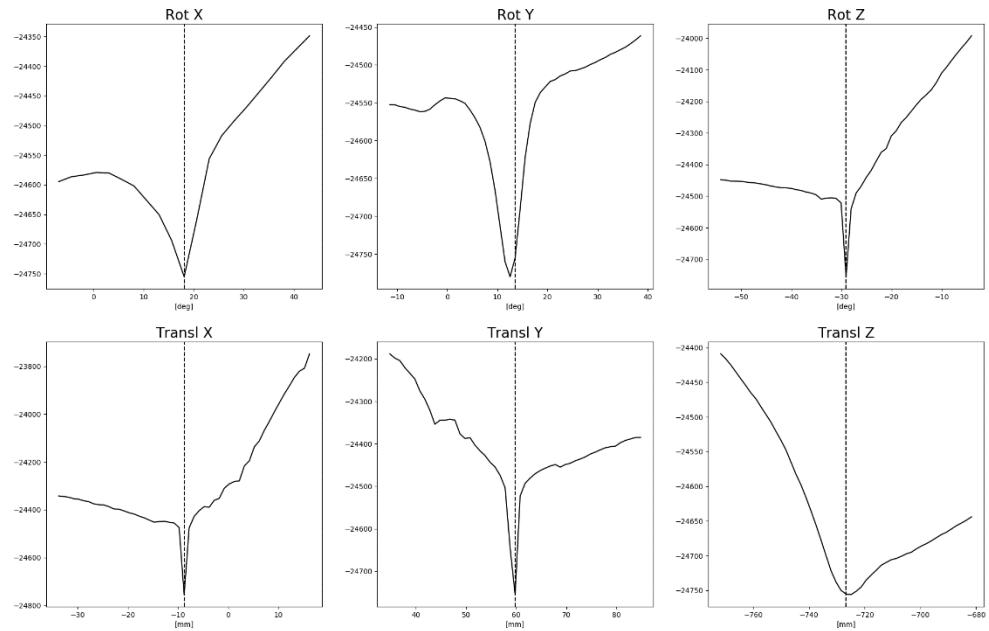
g) Mahfouz ($w_{int} = 1, w_{edge} = 0.01$)

```
C_g1_032
{'Name': 'MahfouzHipHop', 'edgeGsigma': 1, 'edgeGsize': [3, 3], 'MovCannyThr1': 100, 'MovCannyThr2': 200, 'intWeight': 1.0, 'edgeWeight': 0.01, 'useMask': False, 'saveFixedImages': True, 'saveMovImages': False}
```



h) Mahfouz ($w_{int} = 1, w_{edge} = 0.02$)

```
C_g1_032
{'Name': 'MahfouzHipHop', 'edgeGsigma': 1, 'edgeGsize': [3, 3], 'MovCannyThr1': 100, 'MovCannyThr2': 200, 'intWeight': 1.0, 'edgeWeight': 0.02, 'useMask': False, 'saveFixedImages': True, 'saveMovImages': False}
```



i) *Mahfouz* ($w_{int} = 1, w_{edge} = 2.67$)

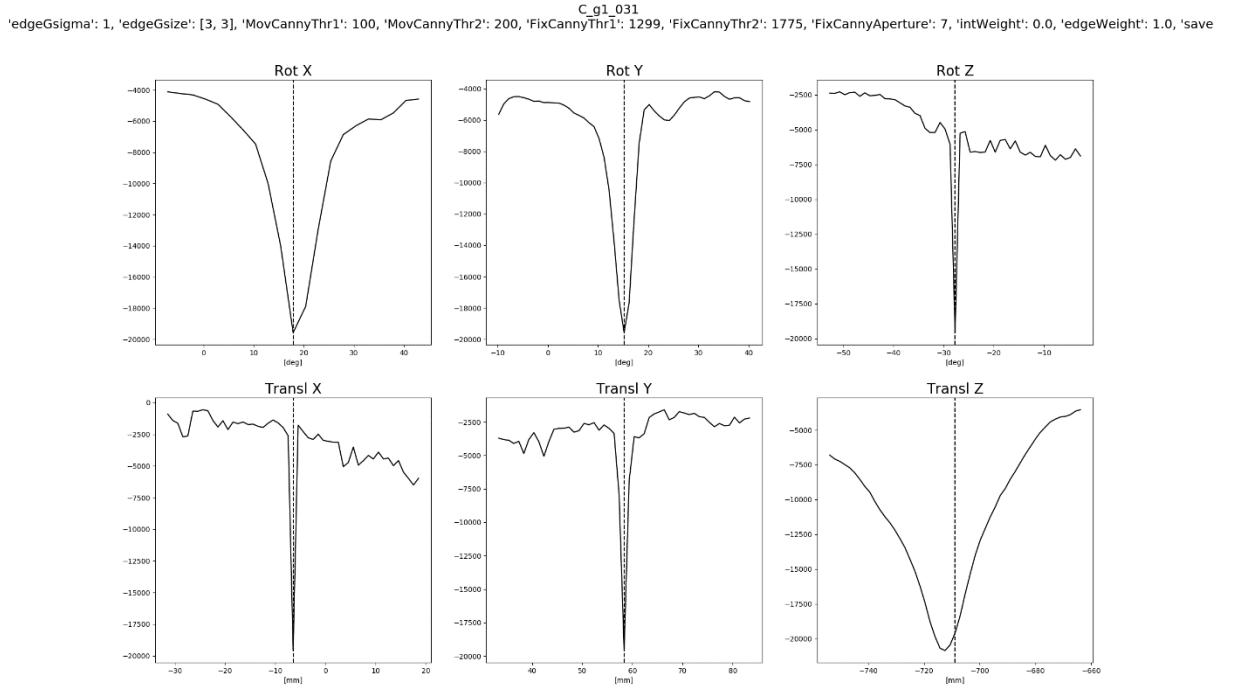


Figure 5.18. Assessment of the performance of the implemented similarity metrics on one in-vivo fluoroscopic image of a hip prosthesis and relative ground truth obtained from manual registration. A vertical dotted line indicates the ground truth pose. Mean Squares Difference and Mean Reciprocal Squares Difference failed at presenting the minimum at the ground truth pose (a). Normalized Cross Correlation (b) worked well for the y- and the z- translations only. Good performance for all pose parameters was observed for Mutual Information (c), Gradient Correlation (d) and the Mahfouz metric with $w_{edge} > 0.01$.

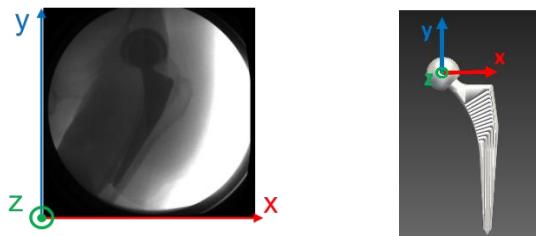
Based on the performance assessment of the similarity metrics for one-dimensional landscapes (Figure 5.16, Figure 5.17, Figure 5.18), Gradient Correlation and the Mahfouz metrics were the most promising ones. In fact, the functions associated to the similarity metrics are six dimensional (three dimensions for the rotations and three dimensions for the translations). Hence, the occurrence of a global minimum along one-dimensional slices of the six-dimensional function does not guarantee that the found minimum is global in the six dimensional space. Therefore, six-dimensional landscapes metrics were produced for the Gradient Correlation, the Mahfouz and the Mutual Information metrics. Evaluation of the shape of these six-dimensional landscapes is also important for the

choice of the optimizer and for the tuning of its parameters. The amount of false local minima indicates how robust shall the optimizer be and which strategy shall be adopted to avoid them. The smoothness and width of the attraction basin around the global optimum indicates how close the starting position should be to the minimum in order to achieve convergence.

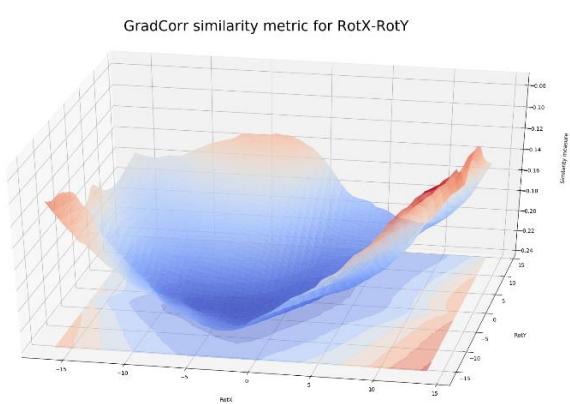
The six-dimensional landscapes relative to the femoral stem are visualized as three-dimensional slices in Figure 5.19 for both the Gradient Correlation and for the Mahfouz similarity metrics. For both metrics the global minimum appeared to be at the ground truth pose (the center of the domain), except for Mahfouz landscape relative to the out-of-plane rotations (RotX and RotY). Both metrics showed a well pronounced global minimum along the dimension relative to the in-plane rotational parameter (RotZ): in fact, an in-plane rotation of the femoral stem produces a distinct peak in the similarity measures at the ground truth pose. On the other hand, wide and more flat profiles around the global minimum were observed for the two out-of-plane rotations (RotX and RotY), due to their smaller effect on the similarity measure. For example, a rotation of the femoral stem around its main symmetry axis (out-of-plane rotation) produces smaller changes in the projected image compared to an in-plane rotation like flexion. The more flat attraction basin for the out-of-plane rotations slows down the convergence of the optimizer, due to the lack of sufficient information about the location of the minimum. The landscape obtained for the Mahfouz metric presented more noise and more small local minima compared to the one obtained for the Gradient Correlation. Also, it presented two major minima for the view relative to the out-of-plane rotations (RotX-RotY): this indicates that two different combinations of the in-plane rotations of the femoral stem produce two images that are both most similar to the real image.

The six-dimensional landscapes relative to the native hip joint are visualized as three-dimensional slices in Figure 5.20, for both the Gradient Correlation and the Mutual

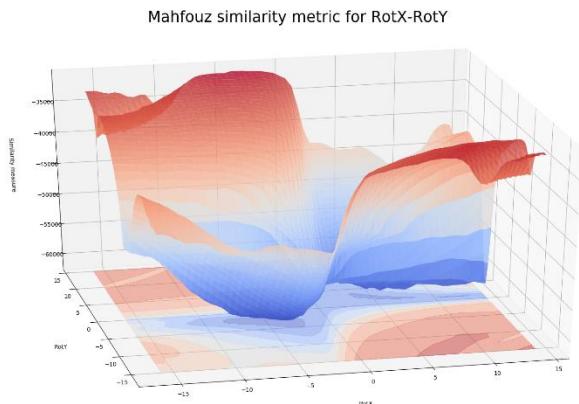
Information similarity metrics. The Gradient Correlation provided a more pronounced global minimum for the out-of-plane rotations (RotX-RotY) compared to the Mutual Information metric. For both metrics, a narrow valley was observed along a direction produced by a 1:1 combination of the RotY and the RotZ: this means that multiple combinations of RotY and RotZ exist such that the similarity metric is similarly high. We point out also the flat profile observed along the out-of-plane translation (Tz), which is shown for the Mutual information metric only: this shows the little effect on the image of such a translation of the target object. Additionally, small local minima were evident from the landscape relative to RotY and RotZ for the Gradient Correlation metric. It should be however better investigated, whether those local minima are true or are rather artefacts generated by the sampling of the search domain.



Gradient Correlation



Mahfouz



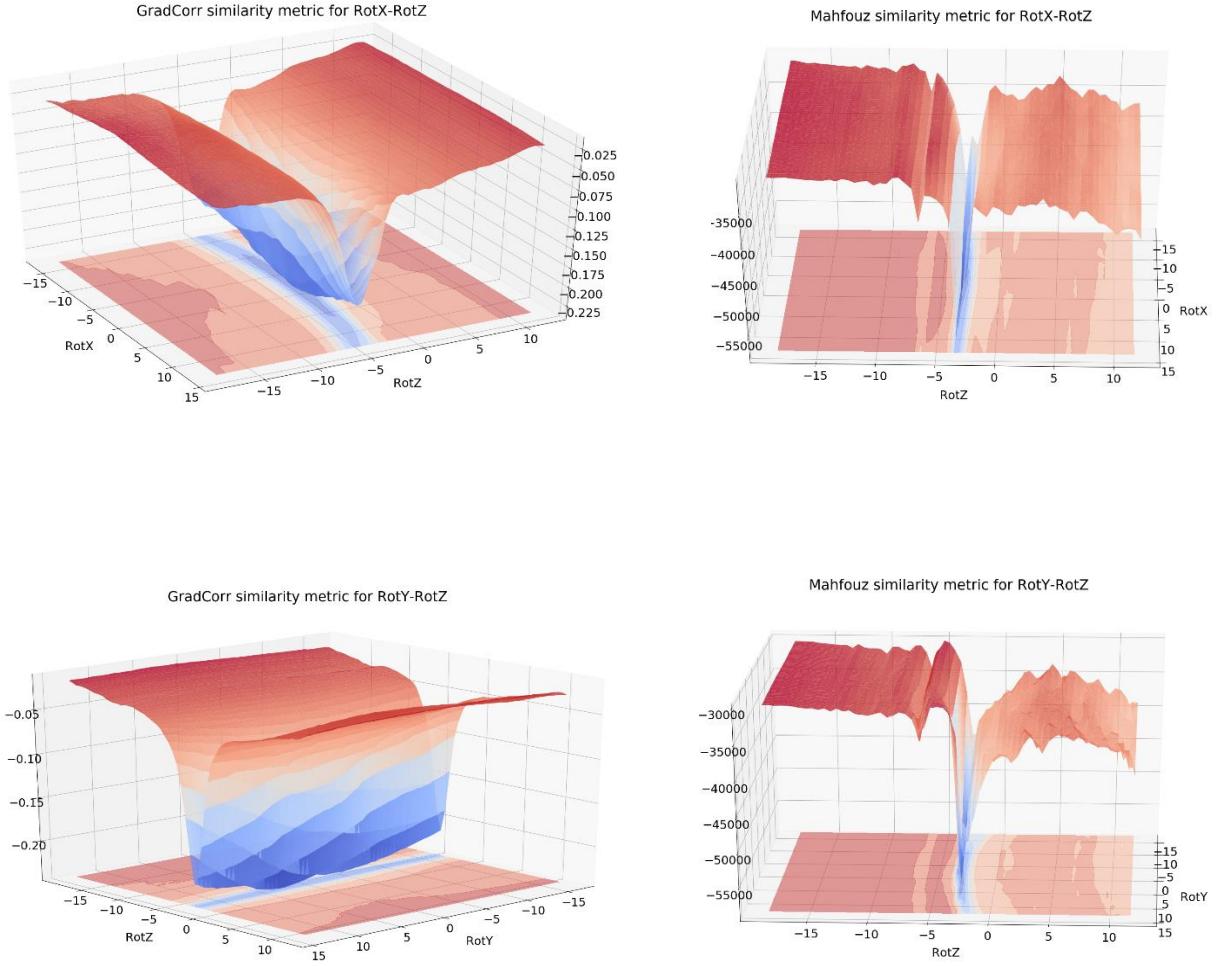
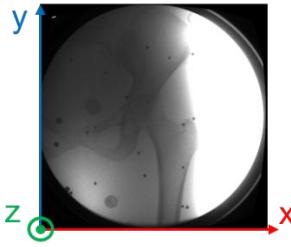
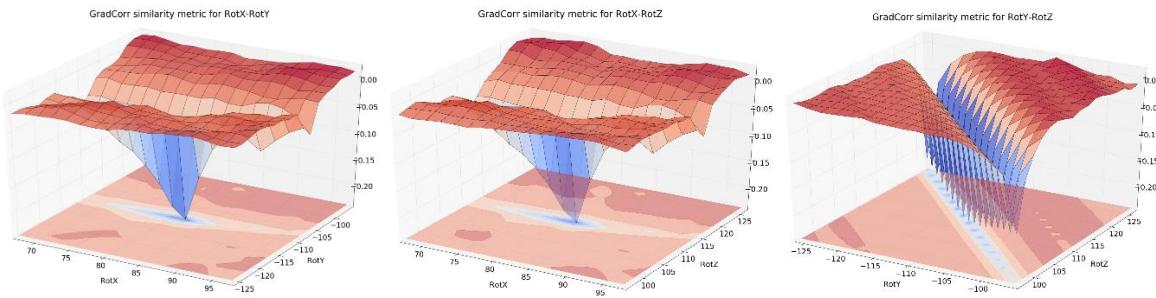


Figure 5.19. Three-dimensional slices of the six-dimensional function relative to the Gradient Correlation (left column) and to the Mahfouz (right column) similarity metrics, for the X-Y, the X-Z and the Y-Z rotations. The landscapes were computed for the in-vivo image of a hip prosthesis shown on top of the figure, and by modifying the rotations of the femoral stem within the range $\pm 15^\circ$ around the ground truth rotations (located in the center of the domain).



Gradient Correlation (rotations only)



Mutual Information (rotations and translations)

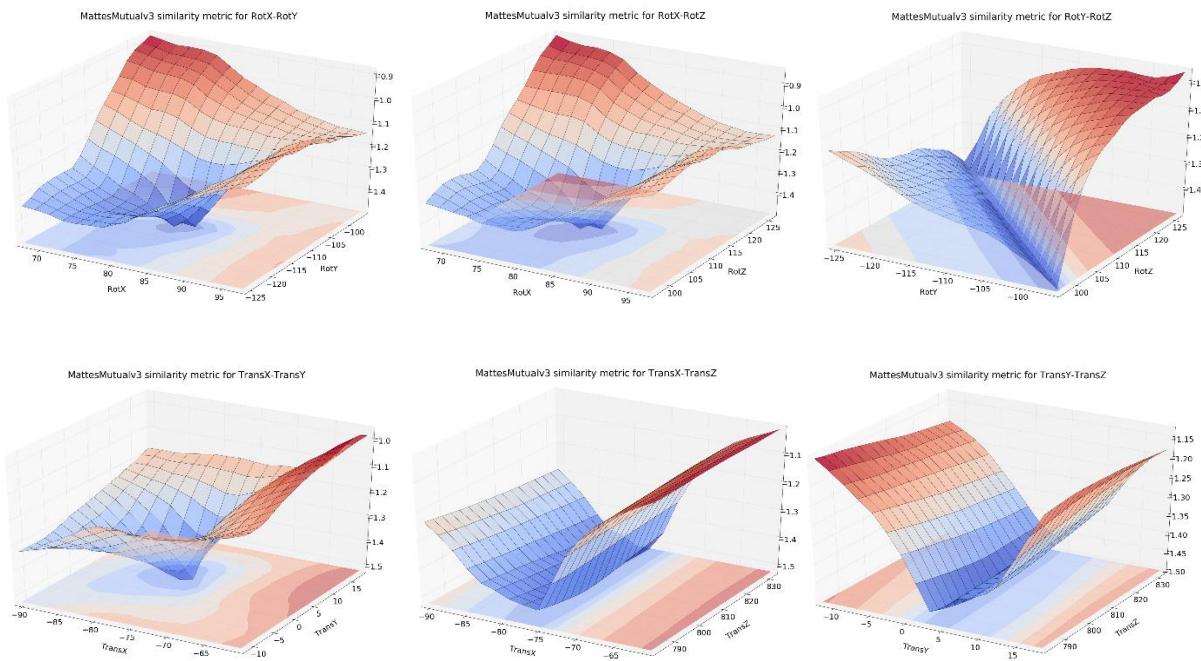


Figure 5.20. Three-dimensional slices of the six-dimensional function relative to the Gradient Correlation and to the Mutual Information similarity metrics, for the a real fluoroscopic image of a native hip phantom shown on top of the figure. The landscapes were computed by modifying the rotations and translations of the hip within the range $\pm 15^\circ$ and ± 15 mm, respectively, around the corresponding ground truth values (located in the center of the domain).

In conclusion, the Gradient Correlation metric appeared to be the most suited six-dimensional function for a more robust convergence of the optimizer to the global minimum. Computation of one DRR and of the relative similarity with the real image took 0.17 seconds for the Gradient Correlation metric, and only 0.06 seconds for the Mahfouz metric. However, while the Mahfouz metric required lower computational time, manual user interaction is needed to identify the image-specific parameters for the Canny edge extraction. In contrast, Gradient Correlation does not need tuning of any parameter. For the above reasons, Gradient Correlation was the preferred metric used in our study for registration of the hip implant from the in-vivo fluoroscopic images. Moreover, its computation can be sped up by parallelization in future implementations.

5.3.4 Optimizer

The performance of the Evolutionary Strategy optimizer was evaluated for registration of sample fluoroscopic images of the hip prosthesis. Convergence was monitored by means of customized functions belonging to the “Optimizer” Python module. Monitoring of the convergence was useful to determine the required maximum number of iterations and the proper width of the search range for each pose parameter.

Examples of the optimization process for registration of the femoral stem with the Gradient Correlation metric are showed in Figure 5.21 and Figure 5.22. In general, convergence to the solution was reached after 2000 iterations, unless the optimizer got stuck into false local minima (bottom plot of Figure 5.21).

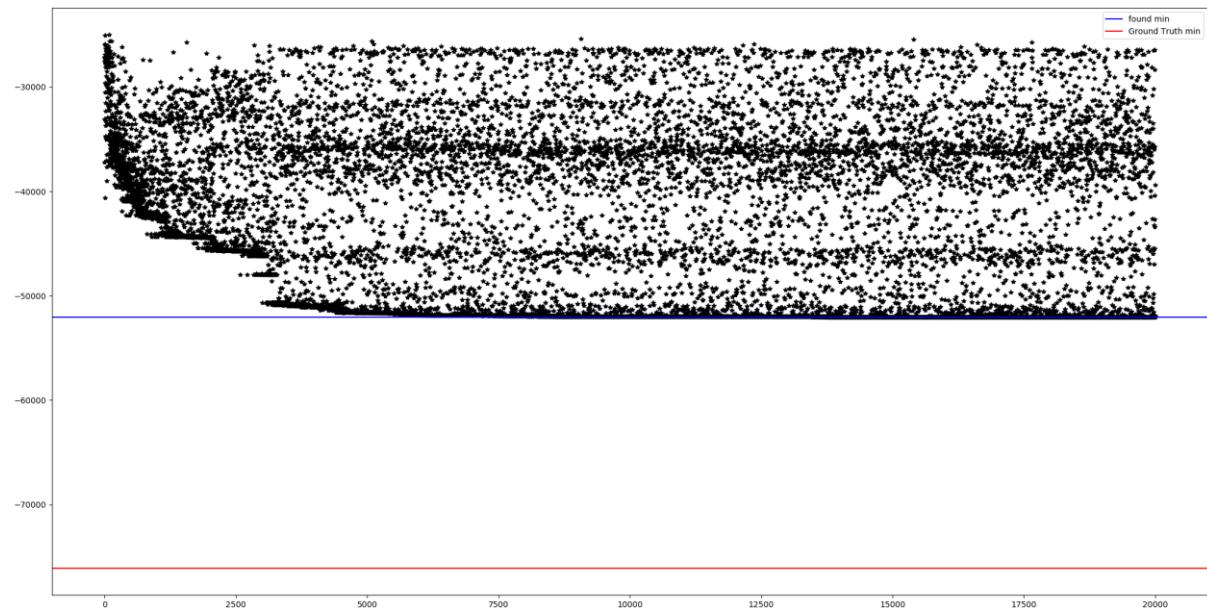
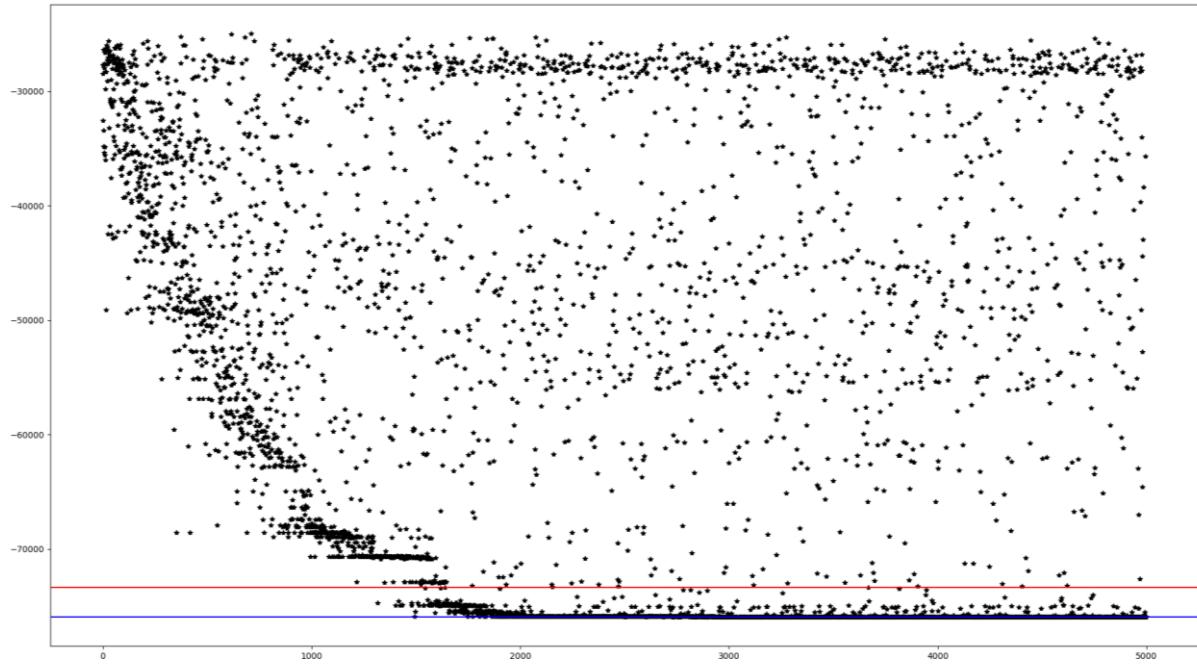


Figure 5.21. Examples of the evolution of a typical optimization of the pose of the femoral stem in order to match a real fluoroscopic image. The Gradient Correlation values computed at each iteration are plotted on the y-axis as a function of the iteration step. The ground truth solution is indicated by a red line, while the final solution at convergence is indicated by a blue line. In the first example on top, the solution found already around iteration #1500 provided a similarity score even higher compared to the ground truth pose. This was because the ground truth pose was retrieved by manual registration, which is less accurate than a validation method based on metal beads as the one described in Chapter 5. In the second example at the bottom, the optimizer got stuck into a local minimum and did not converge to the global minimum.

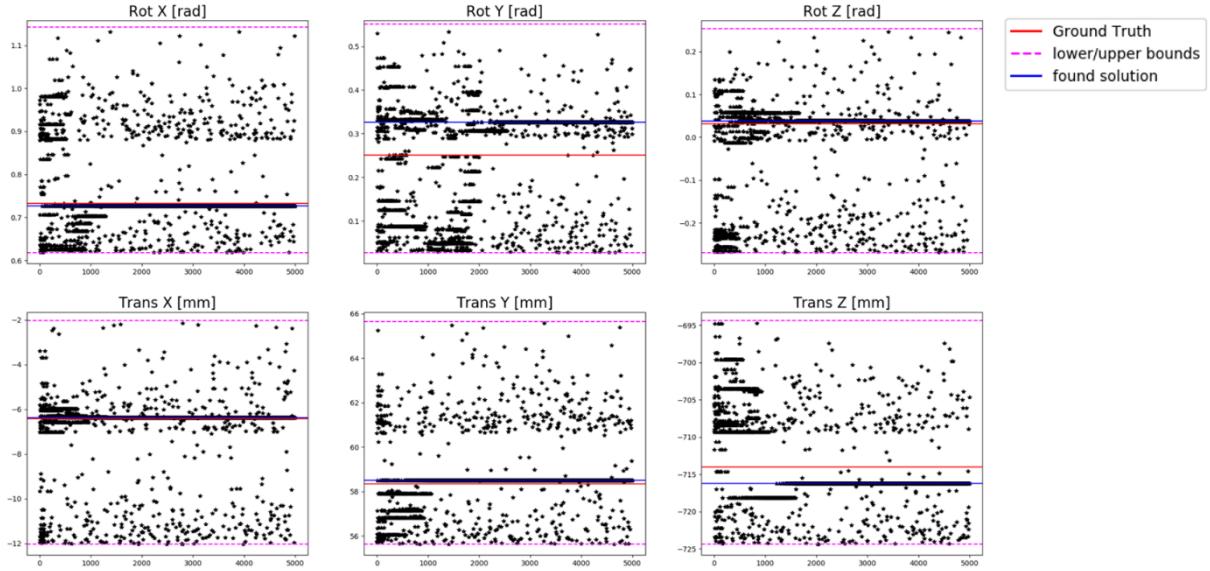


Figure 5.22. Evolution of a six-dimensional optimization for 5000 iterations. Each plot is relative to one of the six searched parameters, and shows the guessed parameter value as a function of the iteration step. The red line indicates the ground truth, the blue line indicates the final solution, and the pink lines indicate the upper and lower bounds for the search range.

5.3.5 Registration of THA to fluoroscopic images

The error in registration of THA to in-vivo fluoroscopic images using the Evolutionary Strategy optimizer was evaluated for the Gradient Correlation (GC), the Mahfouz (MF) and the Mutual Information (MI) metrics (Table 5.1). For each metric, optimization was run 10 times starting from different random initial guesses within a specified distance interval from the ground truth, and the mean absolute registration error was computed as the mean difference between the registered poses and the ground truth pose. The ground truth was obtained from manual registration, as no more accurate method could be used to determine the ground truth for in-vivo images. Optimizations were performed over the full set of pose parameters (6 degrees of freedom d.o.f. for the femoral stem, 5 d.o.f. for the acetabular cup excluding the rotation around the cup's symmetry axis). Additionally, optimizations were performed over a subset only of the pose parameters (3 d.o.f. relative to the three rotations Rx, Ry,Rz for the femoral stem, and 3 d.o.f. relative

to the rotations Rx,Rz and to the out-of-plane translation Tz for the acetabular cup), while leaving the remaining pose parameters at the ground truth values.

The in-plane rotation (Rz) was registered with the best accuracy for all cases (average absolute error was between 0.3° and 1.8°). The reason for this high accuracy is the steep and narrow attraction basin observed along the dimension relative to the in-plane rotation from the function landscapes plotted in Figure 5.19: once identified the domain area of steep descent, the optimizer easily finds the global minimum located at the bottom of the steep valley. On the other hand, the lowest registration accuracy was achieved for the two out-of-plane rotations (Rx and Ry), with mean absolute registration errors between 0.8° and 5.7°. The reason for this poor accuracy is that the attraction basin around the global minimum is rather flat for both the Gradient Correlation and the Mahfouz metrics (Figure 5.19), and thus the optimizer finds it hard to identify such global minimum. The computed registration error for the in-plane translations (Tx and Ty) was between 0.2 mm and 1.4 mm.

As expected, the registration accuracy improved when the optimization was performed over only 3 pose parameters rather than the full set of 6 pose parameters. The overall best performance was achieved with the Gradient Correlation similarity metric, followed by the Mahfouz and at last the Mutual Information metrics.

Registration of the femoral stem using the Gradient Correlation metric was also tested for a poor quality image, in order to test the robustness of the registration in case of low contrast and hardly detectable edges. Compared to the accuracy achieved for a good quality image, the observed registration error decreased by 0.6°, 1.3°, 1.3° for Rx,Ry,Rz and by 0.7 mm and 0.1 mm for Tx,Ty, respectively.

THA part	Metric	Optimizer			Initial Guess			Image quality	Mean absolute registration error (N = 10)									
		# d.o.f.	Search range		Rot [°]	In-plane Transl [mm]	Out-of-plane Transl [mm]		Rot [°]	In-plane Transl [mm]	Out-of-plane Transl [mm]	Rx [°]	Ry [°]	Rz [°]	Tx [mm]	Ty [mm]	Tz [mm]	
			Max # iterations	Rot [°]														
Stem	GC	6	5000	± 15	± 5	± 15	± 10	± 3	± 10	Good	3.4	4.0	0.5	0.8	1.1	-		
	MF										4.2	5.3	0.7	0.4	0.9			
	MI										5.7	13	0.7	1.4	2.0			
Stem	GC	3	5000	± 15	-	-	± 10	-	-	Good	0.8	2.9	0.3	-	-	-		
	MF										1.4	2.5	0.9	-	-			
	MI										1.5	2.8	0.4	-	-			
Cup	GC	5	2000	± 15	± 5	± 15	± 10	± 3	± 10	Good	2.1	-	1.6	0.2	0.8	-		
	MF										2.4	-	1.8	0.2	0.5			
Cup	GC	3	1500	-	-	-	± 10	-	-	Good	1.4	-	0.3	-	-	-		
	MF										0.9	-	0.5	-	-			
Stem	GC	6	2000	± 15	± 5	± 15	± 10	± 3	± 10	Bad	4.0	5.3	1.8	1.5	1.2			

Table 5.1. Performance of the HipHop framework evaluated for the registration of THA to in-vivo fluoroscopic images. The average absolute registration error for the 6 pose parameters (three rotations Rx, Ry, Rz, and three translations Tx, Ty, Tz) of the acetabular cup and of the femoral stem was computed using different metrics (GC = Gradient Correlation, MF = Mahfouz, MI = Mutual Information) and different settings for the Evolutionary Strategy optimizer. For registration of the stem with GC, the registration error was evaluated also for an in-vivo image with poor image contrast. The ground truth was obtained by manual registration. The initial guess was chosen as a random offset from the ground truth pose within the specified interval. 6-dimensional optimizations were run over the full set of parameters, and 3-dimensional optimizations were run over a subset of the pose parameters, while leaving the remaining ones at the ground truth values. The registration error was not reported for the out-of-plane translations (Tz) due to the expected low accuracy of the ground truth, and for the rotation of the cup around its symmetry axis (Ry).

In conclusion, our analysis of the registration accuracy of THA to in-vivo fluoroscopic images showed that Gradient Correlation was the best performing similarity metric, as expected from the analysis of the function landscapes in Section 5.3.3 (Figure 5.19, Figure 5.20). Moreover, the improvement observed for the optimization of the implant pose over 3 pose parameters only instead of 6 suggested to perform registration on a subset of the pose parameters, whenever the remaining ones can be accurately identified in advance. Based on these guidelines, the final strategy adopted in our study for registration of THA to each of the acquired fluoroscopic images consisted in the following steps:

Pre-processing:

- 1) The pose of the femoral stem and of the acetabular cup were initialized based on motion capture (Section 5.2.1).

- 2) A mask including the femoral stem only was created in order to avoid false registrations due to the edges of the bony femoral shaft being mistaken for those of the femoral stem.

Registration of the femoral stem:

- 3) The two in-plane translations of the femoral stem were manually refined from the initialized values in order to best fit the femoral head with its projection in the image.
- 4) Starting from the initialized pose, the femoral stem was automatically registered using the Gradient Correlation metric by running the optimization over four dimensions: all of the three rotational parameters and the out-of-plane translation. The search range was $\pm 20^\circ$ from the initial pose for the rotations, and ± 20 mm from the initial pose for the out-of-plane translation. These bound constraints were chosen also based on the accuracy of the motion-capture based initialization, which is discussed in Section 5.3.1. The optimizer was stopped after 3000 iterations.

Registration of the acetabular cup:

- 5) Once completed the registration of the femoral stem, the translations of the acetabular cup were set to the registered translations of the femoral stem, under the assumption of no occurrence of hip separation. This assumption was verified to comply with the achievable accuracy of our method, since no hip separation could be visible from our single-plane fluoroscopic images.
- 6) Once fixed the translations for the acetabular cup, the two rotations other than the one around its symmetry axis were determined by running a two-dimensional optimization using the Gradient Correlation metric. The search range was $\pm 15^\circ$ from the initialized rotations. The optimizer was stopped after 800 iterations. Having to determine two rotational parameters only, the optimizer provided solutions with a high level of accuracy.
- 7) Finally, the rotation of the acetabular cup about its symmetry axis was determined from the skin markers, as described in Chapter 6 - Section 6.2.

5.4 Discussion and conclusions

Application of the *HipHop* algorithm for registration of THA to the in-vivo fluoroscopic images acquired in our study sped up considerably the analysis of the large amount of collected data. Yet, manual user interaction was still needed to set the in-plane translations of the femoral stem (step 3 above), and in several cases to refine the solution provided by the automatic algorithm. In order to reduce further the required user interaction and thus to decrease further the computational time necessary for registration, the in-plane translations shall be automatically set from detection of the center of the femoral head in the image, as mentioned in Section 5.2.1 and in Figure 5.7. For this purpose, a technique should be implemented in order to automatically identify the parameters required by the Hough transform to detect the head center.

Moreover, accuracy and robustness of the registration should be improved in order to avoid the necessary manual intervention for refinement of the solutions. One direction of possible future research could be a better implementation of the optimization. For example, convergence of the optimizer can be improved in terms of speed and accuracy by adapting the width of the effective searched domain to the landscape of the cost function. The analysis of the six-dimensional landscapes presented in Section 5.3.3 showed that there are preferential dimensions with a pronounced global minimum and dimensions with a more flat profile around the minimum. For the latter ones, search of the global minimum might be slow and difficult for a standard optimizer, as it would not have enough information to identify the direction of descent leading to the minimum. A self-adaptive version of the Evolutionary Strategy optimizer may be adopted, in which the width of the search range is adapted on the run for each specific dimension based on the landscape. In such way, the width of the search range for the dimension associated to a steep and narrow profile around the minimum will become smaller and smaller with the increasing number of iterations, while the search range along the dimension associated

to a flat valley will be kept wide in order to increase the chance of spotting the minimum (Figure 5.23). Furthermore, improved robustness to local minima could be achieved by adopting a multi-restart approach (Figure 5.24), in which the optimization starts from a new random initial guess when the maximum number of iterations is reached or when the computed similarity values does not change of more than a certain threshold (indicating that the optimizer is stuck to a minimum). At last, registration accuracy could be improved in the future by simultaneously registering both segments of the hip joint under an optimization constraint that avoids physical collisions between the segments.

A limitation of the present analysis was that the performance of the *HipHop* algorithm for registration of THA to in-vivo fluoroscopic images was based on a manually determined ground truth. This ground truth suffers from the registration errors intrinsic to single-plane fluoroscopy, as reported in Chapter 3. However, a more accurate ground truth could not be obtained for those in-vivo images. Another limitation was the limited amount of repetitions per registration scenario ($N=10$) that were used to calculate the mean registration errors in Table 5.1. Due to the project time constraints, the registration performance was not evaluated for the native hip joint with the gold standard validation database presented in Chapter 5.

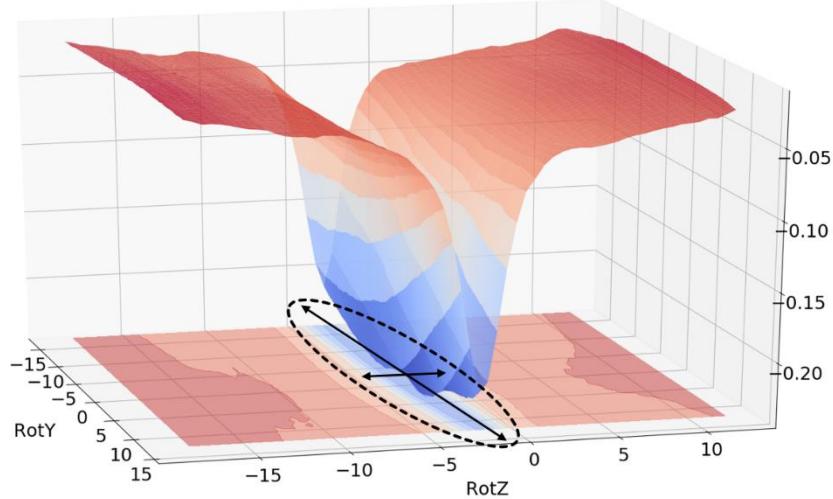


Figure 5.23. A self-adaptive Evolutionary Strategy optimizer will keep a wide search range along the dimension showing a flat profile around the global minimum (RotY) and will narrow the search range along the dimension showing a steep and narrow valley around the global minimum.

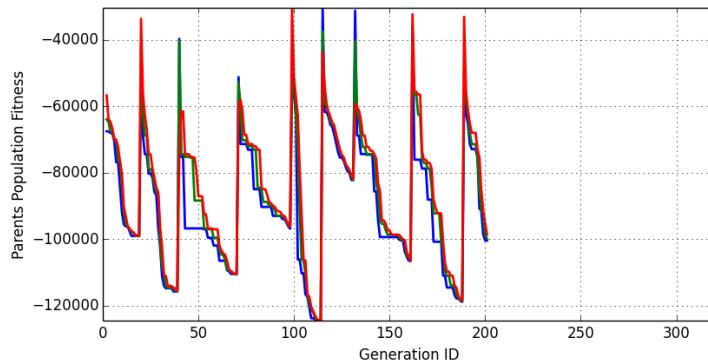


Figure 5.24. Multiple-restart of the optimizer at different initial guesses can help avoiding to get stuck into local minimum and finding the global one.

Acknowledgments

We would like to thank sincerely Simon Andermatt for his support on the GPU-based parallelization of the projector function, and Filippo Arcadu for his support on the use of Cython for wrapping the C++ codes.

Very much appreciated is also the support of Dmitriy Alexeev for the design of the optimization procedure, and of Jerome Schmid for sharing his theoretical and practical knowledge about image registration.

5.5 Annex: Parallelized generation of Digitally Reconstructed Radiographs.

5.5.1 Siddon implementation

The CT scan is expressed as three sets of orthogonal planes with coordinates X_i, Y_i, Z_i :

$$\begin{aligned} X_i &= X_0 + i\Delta X \\ Y_i &= Y_0 + j\Delta Y \\ Z_i &= Z_0 + k\Delta Z, \end{aligned} \quad (5.1)$$

where $\Delta X, \Delta Y, \Delta Z$ are the voxel dimensions, i, j, k are plane indices in the three directions, and X_0, Y_0, Z_0 are the three coordinates of the side planes that are defined by i, j, k equal to zero.

The path of a ray from the source $p_1 = (p_{1x}, p_{1y}, p_{1z})$ to the center of a single pixel detector $p_2 = (p_{2x}, p_{2y}, p_{2z})$ is represented in the parametrical form as:

$$\begin{aligned} x &= p_{1x} + \alpha(p_{2x} - p_{1x}) \\ y &= p_{1y} + \alpha(p_{2y} - p_{1y}) \\ z &= p_{1z} + \alpha(p_{2z} - p_{1z}), \end{aligned} \quad (5.2)$$

where α is in the range [0,1] for points located between source and detector.

From Equations 5.1, 5.2 the values of α at an intersection of the ray with planes of indices i, j, k can be calculated by:

$$\begin{aligned}\alpha_x(i) &= \frac{x_0 + i\Delta X - p_{1x}}{p_{2x} - p_{1x}} \\ \alpha_y(j) &= \frac{y_0 + j\Delta Y - p_{1y}}{p_{2y} - p_{1y}} \\ \alpha_z(k) &= \frac{z_0 + k\Delta Z - p_{1z}}{p_{2z} - p_{1z}},\end{aligned}\quad (5.3)$$

Equations 5.3 can be written in recursive form as:

$$\begin{aligned}\alpha_x(i) &= \alpha_x(i-1) + \frac{\Delta X}{p_{2x} - p_{1x}} = \alpha_x(i-1) + \delta\alpha_x \\ \alpha_y(j) &= \alpha_y(j-1) + \frac{\Delta Y}{p_{2y} - p_{1y}} = \alpha_y(j-1) + \delta\alpha_y \\ \alpha_z(k) &= \alpha_z(k-1) + \frac{\Delta Z}{p_{2z} - p_{1z}} = \alpha_z(k-1) + \delta\alpha_z\end{aligned}\quad (5.4)$$

Using Equations 5.4, the values of α for all the intersections between the source point and the voxel the ray travels to can be calculated.

The original Siddon's algorithm [127] calculates a list of α values for all intersected planes and for all directions:

$$\begin{aligned}\overline{\alpha_x} &= \{\alpha_x(i_{min}), \dots, \alpha_x(i_{max})\} \\ \overline{\alpha_y} &= \{\alpha_y(j_{min}), \dots, \alpha_y(j_{max})\} \\ \overline{\alpha_z} &= \{\alpha_z(k_{min}), \dots, \alpha_z(k_{max})\},\end{aligned}\quad (5.5)$$

where subscripts "*min*" and "*max*" denote the first and last intersections of the ray with the orthogonal planes when traveling from the source to the target pixel. The three lists are merged together and sorted in ascending order to produce the list of N values:

$$\bar{\alpha} = \{\alpha_{\min}, \text{merge}(\bar{\alpha}_x, \bar{\alpha}_y, \bar{\alpha}_z), \alpha_{\max}\}$$

where two consecutive elements of $\bar{\alpha}$ are associated with entering and leaving a voxel. The indices $i(m), j(m), k(m)$ of the voxel corresponding to each value $\bar{\alpha}(m)$ can be calculated from:

$$\begin{aligned} i(m) &= \lfloor \varphi_x(\alpha_{mid}) \rfloor = \left\lfloor \frac{p_{1x} + \alpha_{mid}(p_{2x} - p_{1x}) - X_0}{\Delta x} \right\rfloor \\ j(m) &= \lfloor \varphi_y(\alpha_{mid}) \rfloor = \left\lfloor \frac{p_{1y} + \alpha_{mid}(p_{2y} - p_{1y}) - Y_0}{\Delta y} \right\rfloor \\ k(m) &= \lfloor \varphi_z(\alpha_{mid}) \rfloor = \left\lfloor \frac{p_{1z} + \alpha_{mid}(p_{2z} - p_{1z}) - Z_0}{\Delta z} \right\rfloor, \end{aligned} \quad (5.6)$$

Where $\lfloor \cdot \rfloor$ denotes the truncation operator, and α_{mid} is given by:

$$\alpha_{mid} = \frac{\bar{\alpha}(m+1) + \bar{\alpha}(m)}{2}.$$

The radiological path length RPL is eventually calculated as the weighted sum of CT values $\rho(i, j, k)$ the ray passes when traveling from the source to the target pixel, with:

$$RPL = d_{12} \sum_{m=1}^N \rho(i(m), j(m), k(m)) * l(m)$$

where d_{12} is the source-destination pixel distance, ρ is the Hounsfield CT value at a given voxel (i, j, k) , and $l(m)$ is a weighting factor corresponding to the length traveled in the m -th traversed voxel (i, j, k) :

$$l(m) = (\bar{\alpha}(m+1) - \bar{\alpha}(m)).$$

5.5.2 *HipHop* implementation

For the serial implementation of Siddon's algorithm, Jacob pointed out that the speed was greatly reduced by the frequent use of Equations 5.6 where floating point are converted to integer values to represent voxel indeces. For implementation of the Siddon's algorithm on the graphics card, storage of the three lists of floating point numbers $\overline{\alpha_x}, \overline{\alpha_y}, \overline{\alpha_z}$ requires an amount of memory that exceeds the capacity of the graphic card, for usual CT scan sizes [126]. Thus, the part of the algorithm where $\overline{\alpha_x}, \overline{\alpha_y}, \overline{\alpha_z}$ are computed and stored must be rewritten.

The final algorithm for this thesis is then written as proposed by Greef et al. [126], by using a stepping approach after an initialization step. The initialization step aims at computing a) the *min* and *max* values of α , b) the indices relative to the first and last intersected planes along the three directions, c) the indices of the first intersected voxel. After this initialization step, the algorithm continues with a loop where the next intersected voxel is iteratively determined from the computation of the next intersected plane.

The initialization step is:

- a) Instead of calculating the arrays $\overline{\alpha_x}, \overline{\alpha_y}, \overline{\alpha_z}$ (Equations 5.6), only the *min* and *max* values of α corresponding to the first and the last intersection of the X-ray with the image planes, respectively, are calculated by:

$$\alpha_{min} = \max(\alpha_{xmin}, \alpha_{ymin}, \alpha_{zmin})$$

$$\alpha_{max} = \min(\alpha_{xmax}, \alpha_{ymax}, \alpha_{zmax})$$

With

$$\alpha_{xmin} = \min(\alpha_x(0), \alpha_x(N_x - 1)) , \quad \alpha_{xmax} = \max(\alpha_x(0), \alpha_x(N_x - 1))$$

$$\alpha_{ymin} = \min(\alpha_y(0), \alpha_y(N_y - 1)) , \quad \alpha_{ymax} = \max(\alpha_y(0), \alpha_y(N_y - 1))$$

$$\alpha_{zmin} = \min(\alpha_z(0), \alpha_z(N_z - 1)) , \quad \alpha_{zmax} = \max(\alpha_z(0), \alpha_z(N_z - 1)),$$

where N_x, N_y, N_z are the number of planes along the x, y, z directions, respectively, and equations 7-9 are used for computation of the α values.

- b) The indices (i_{min}, i_{max}) relative to the first and last intersected planes along the x direction are calculated with Equations 5.7 if $p_{1x} < p_{2x}$ or with Equations 5.8 if $p_{1x} > p_{2x}$. The definition of $\varphi(\alpha)$ is given by Equations 5.6. Similar formulas hold for (j_{min}, j_{max}) and for (k_{min}, k_{max}) .

$$\begin{aligned} \alpha_{min} = \alpha_{xmin} &\rightarrow i_{min} = 1 \\ \alpha_{min} \neq \alpha_{xmin} &\rightarrow i_{min} = \lceil \varphi_x(\alpha_{min}) \rceil \\ \alpha_{max} = \alpha_{xmax} &\rightarrow i_{max} = N_x - 1 \\ \alpha_{max} \neq \alpha_{xmax} &\rightarrow i_{max} = \lfloor \varphi_x(\alpha_{max}) \rfloor \end{aligned} \quad (5.7)$$

$$\begin{aligned} \alpha_{min} = \alpha_{xmin} &\rightarrow i_{max} = N_x - 2 \\ \alpha_{min} \neq \alpha_{xmin} &\rightarrow i_{max} = \lfloor \varphi_x(\alpha_{min}) \rfloor \\ \alpha_{max} \neq \alpha_{xmax} &\rightarrow i_{min} = \lceil \varphi_x(\alpha_{max}) \rceil \\ \alpha_{max} = \alpha_{xmax} &\rightarrow i_{min} = \lceil \varphi_x(\alpha_{max}) \rceil \end{aligned} \quad (5.8)$$

- c) The indices of the first intersected voxel are calculated with Equations 5.6:

$$i(0) = \lfloor \varphi_x(\alpha_{mid0}) \rfloor = \left\lfloor \frac{p_{1x} + \alpha_{mid}(p_{2x} - p_{1x}) - X_0}{\Delta X} \right\rfloor$$

$$j(0) = \lfloor \varphi_y(\alpha_{mid0}) \rfloor = \left\lfloor \frac{p_{1y} + \alpha_{mid}(p_{2y} - p_{1y}) - Y_0}{\Delta Y} \right\rfloor$$

$$k(0) = \lfloor \varphi_z(\alpha_{mid0}) \rfloor = \left\lfloor \frac{p_{1z} + \alpha_{mid}(p_{2z} - p_{1z}) - Z_0}{\Delta Z} \right\rfloor$$

With:

$$\alpha_{mido} = \frac{\min(\alpha_x(i_{min}), \alpha_y(j_{min}), \alpha_z(k_{min})) + \alpha_{min}}{2}$$

After this initialization step, the algorithm continues with a loop where the next intersecting plane is determined. This is achieved by computing the minimum of the values of alpha for the three potential next planes of intersection. This next intersecting plane now becomes the current plane of intersection and a new potential plane of intersection is calculated. This loop is continued until the value of alpha for the current intersecting plane is larger than or equal to α_{max} . See following pseudocode for details.

5.5.3 *HipHop* pseudocode

Initialization step

$$RPL = 0$$

$$\alpha_{current} = \alpha_{min}$$

α values of potential next intersecting planes:

$$\alpha_{x,next} = \alpha_x(i_{min})$$

$$\alpha_{y,next} = \alpha_y(j_{min})$$

$$\alpha_{z,next} = \alpha_z(k_{min})$$

Initialize first voxel ($i(0), j(0), k(0)$)

Update step

While $\alpha_{current} < \alpha_{max}$

If $\min(\alpha_{x,next}, \alpha_{y,next}, \alpha_{z,next}) = \alpha_{x,next}$

(the next intersecting plane is an x plane)

Update radiological path length

$$l(i, j, k) = \alpha_{x,next} - \alpha_{current}$$

$$RPL = RPL + \rho(i, j, k) * l(i, j, k)$$

Update voxel index i

$$i = \begin{cases} i + 1 & \text{if } p_{1x} < p_{2x} \\ i - 1 & \text{if } p_{1x} > p_{2x} \end{cases}$$

Update α values

$$\alpha_{current} = \alpha_{x,next}$$

$$\alpha_{x,next} = \alpha_{x,next} + \delta\alpha_x \text{ (from equation 10)}$$

If $\min(\alpha_{x,next}, \alpha_{y,next}, \alpha_{z,next}) = \alpha_{y,next}$

(the next intersecting plane is an y plane)

Update radiological path length

$$l(i, j, k) = \alpha_{y,next} - \alpha_{current}$$

$$RPL = RPL + \rho(i, j, k) * l(i, j, k)$$

Update voxel index j

$$j = \begin{cases} j + 1 & \text{if } p_{1y} < p_{2y} \\ j - 1 & \text{if } p_{1y} > p_{2y} \end{cases}$$

Update α values

$$\alpha_{current} = \alpha_{y,next}$$

$$\alpha_{y,next} = \alpha_{y,next} + \delta\alpha_y \text{ (from equation 11)}$$

$$\text{If } \min(\alpha_{x,next}, \alpha_{y,next}, \alpha_{z,next}) = \alpha_{z,next}$$

(the next intersecting plane is an z plane)

Update radiological path length

$$l(i,j,k) = \alpha_{z,next} - \alpha_{current}$$

$$RPL = RPL + \rho(i,j,k) * l(i,j,k)$$

Update voxel index k

$$k = \begin{cases} k + 1 & \text{if } p_{1z} < p_{2z} \\ k - 1 & \text{if } p_{1z} > p_{2z} \end{cases}$$

Update α values

$$\alpha_{current} = \alpha_{z,next}$$

$$\alpha_{z,next} = \alpha_{z,next} + \delta\alpha_z \text{ (from equation 12)}$$

Update final radiological path length

$$RPL = RPL * d_{12}$$