Julio Alvarez-Gomez*, Hubert Roth, Jürgen Wahrburg

Intraoperative Registration of 2D C-arm Images with Preoperative CT Data in Computer Assisted Spine Surgery: Motivation to Use Convolutional Neural Networks for Initial Pose Generator

<https://doi.org/10.1515/cdbme-2019-0007>

Abstract: In this paper, we present an approach for getting an initial pose to use in a 2D/3D registration process for computer-assisted spine surgery. This is an iterative process that requires an initial pose close to the actual final pose. When using a proper initial pose, we get registrations within two millimeters of accuracy. Consequently, we developed a fully connected neural network (FCNN), which predicts the pose of a specific 2D image within an acceptable range. Therefore, we can use this result as the initial pose for the registration process. However, the inability of the FCNN for learning spatial attributes, and the decrease of the resolution of the images before inserting them in the FCNN, make the variance of the prediction large enough to make some of the predictions entirely out of the acceptable range. Additionally, new researches in deep learning field have shown that convolutional neural networks (CNNs) offer high advantages when the inputs of the net are images. We consider that using CNNs can help to improve our results, generalizing the system for a greater variety of inputs, and facilitating the integration with our current workflow. Then we present an outline for a CNN for our application, and some further steps we need to complete to achieve this implementation.

Keywords: 2D/3D registration, intraoperative registration, computer assisted surgery, spine surgery, neural network, convolutional neural network.

<https://doi.org/10.1515/cdbme-2019-XXXX>

***Corresponding author: Julio Alvarez-Gomez:** Institute of Control Engineering (RST) and Center for Sensor Systems (ZESS), University of Siegen, Siegen, Germany, e-mail: julio.agomez@uni-siegen.de.

Hubert Roth, Jürgen Wahrburg: Institute of Control Engineering (RST) and Center for Sensor Systems (ZESS), University of Siegen, Siegen, Germany.

1 Introduction

2D to 3D registration algorithms have been studied intensively based on intensity-base and stochastic-base methods. We use preoperative CT-Scans taken for planning and register them using intraoperative C-arm images in our research on spine surgery and implantation of pedicle screws. Previous works using multimodality images have found that the initial guess used for the registration has a high impact on the outcome [1][2]. We experience the same phenomena when carrying out some tests using two different lumbar spine phantoms from whom we took CT-Scans and X-ray images. Using measurement methods and optimizing cost functions, we have achieved registrations that result in errors within $\pm 2\text{mm}$ when the initial pose was no more than $\pm 5^\circ$ and $\pm 10\text{mm}$ away in each axis respecting to the actual pose. At this point in our study, the initial pose is chosen manually, but it is necessary to remove user intervention to the minimum extent. Consequently, using an image processing method, that gives a pose within our critical range, is our next step to support intraoperative navigation based on CT-based planning. For automating this process, we gathered a set of X-ray images of both phantoms from whom we know the C-arm poses. Starting for predicting the rotation, we split the data set in training and testing sets, and used them for training and testing the fully connected neural network. The inputs of the FCNN were the X-ray image and CT-scans lateral and anteroposterior projection. We got pose predictions, which were within our range for being inputted as initial pose for the registration process, but others were completely out of range. Other researches have proposed using CNNs [2] and Regression learning [3] for developing the entire registration process reporting high rates of successful registrations. In our case, to improve the prediction rate, and making a better generalization, we propose an approach using CNNs as a mean

for obtaining the initial pose for our registration procedure. CNNs have been the most selected algorithm for computer vision since last years because they can learn autonomously more features from input images than any other neural networks [4]. That means making a better generalization over a dataset and improving the results. Additionally, Python allows deploying trained CNN into C++, which makes the predictor compatible with our current development.

2 Materials and Methods

2.1 Software Tools

We are working under the Mevislab framework, which includes a digital reconstructed radiography (DRR) module. It has as user inputs the positions and orientation of the DICOM volume respecting to the rendering element. We then called registration pose to the position and orientation obtained after the registration process. It consists of 6 degrees of freedom (DOF), three in translation $\{x,y,z\}$ and three in rotation $\{\alpha_z, \alpha_x, \alpha_y\}$. The order of the rotations is defined as so by the DRR module, which applies a Eulerian ZXY rotation. From the perspective of the render interface, the last rotation is performed over the rendered image after applying the Z and X rotation. In other words, once the render of Z and X rotation is made, the resulting image is rotated, which completes the Y rotation.

For developing the FCNN, we used Matlab. There we used some native libraries for reading DICOM files, our X-ray images.

2.2 Phantoms and C-arm

In our laboratory, we have two lumbar phantoms. One of them contains the vertebra L2-L5, the other L1-L5 including the sacrum. From both phantoms, we have their CT-scans. In our facilities, it is available a Ziehm Vario 3D C-arm, which was used to take about 2500 X-ray images. They were taken using the auto scanning mode of the C-arm, which allows to take up to 120 images while the C-arm rotates 130° around its center. Playing with the location of the Phantoms respecting to the C-arm's center as well as their orientation, we consider every captured image unique. That means every image is different from any other from the dataset in at least one element of its 6 DOF.

2.3 Data Preparation

Later, all of the images were registered, which means we computed the registration pose of all of them. We created the dataset file relating each image with its registration pose. Moreover, the input images have a native resolution of 568×568 pixels, but we reduced their resolution to 30×30 pixels using an average pooling layer. Later on, we used 70% of the data for the training set and the remaining 30% as the testing set.

2.4 Defining the FCNN

Stating the fact that we do not have enough samples for training a deep neural network, we decided to go for a fully connected neural network because it requires less parameters to be computed. The reduction of input images resolution was carried out consequently to keep a balance between the dataset size and the number of weights to train, i.e., to avoid underfitting. The inputs of the FCNN are one actual X-ray image, and the lateral and AP images of the respective DICOM set created by the DRR module. All in all, our FCNN has 2700 inputs (30×30 pixels per image by 3 images). In the beginning, we defined the outputs of the FCNN as the three angles of the registration pose, but we decided to simplify them. Having only the predictions of the Z and X angles, we could render a DRR image and estimate the Y angle by traditional image processing methods. Therefore, we have only the Z and X rotation as outputs of our FCNN. The activation function used between hidden layers was the sigmoid, the used cost function was mean squared error with regularization, and the used optimization function was conjugate gradients. For the training, we grouped the dataset as follows: 80% of the samples as the training set and the 20% remaining as the testing set.

There are no equations for defining the proper topology of a FCNN, or how many neurons per layer must be used. There are, nevertheless, some experimental rules of thumbs that show some ranges of values, which could lead to good results. For this reason, we followed some recommendation about the number of neurons depending on the number of layers. However, there is still some uncertainties in the number of hidden layers, and the exact number of neurons in each specific layer.

We addressed this problem by creating different topologies, with different amount of neurons in each layer and comparing the error prediction in each of these networks, but avoiding deeper layers to have more neurons than outer layers. We ran the tests changing the number of neurons in topologies

containing one, two, and three hidden layers. That means, we tested the following topologies:

- 2700 - H1 - 2
 - 2700 - H1 - H2 - 2
 - 2700 - H1 - H2 - H3 - 2
- where: $H1 \geq H2 \geq H3$

Additionally, each layer was changed in intervals of 200 neurons moving from 200 until 1000, so

$$H1, H2, H3 \rightarrow \{200, 400, 600, 800, 1000\}.$$

After finding the best layout, we changed the number of epochs used for each training over the best network and also the regularization value to avoid overfitting the network. The regularization term (λ) was moved from 3×10^{-7} until 3 with logarithmic increments of 0.5, so

$$\lambda \rightarrow \{3 \times 10^{-7}, 1 \times 10^{-7}, 3 \times 10^{-6}, 1 \times 10^{-6}, \dots, 0.3, 1, 3\}.$$

The number of epochs, on the other hand, were started in 800 increasing to the next one in roughly 66%, so

$$\text{Epochs} \rightarrow \{0.8, 1.3, 2, 3, 4.5, 6.8, 10, 15, 22, 34\} \times 10^3$$

3 Results

3.1 Finding the FCNN Topology

The best topologies for each hidden layer are shown in Table 1. For the topology with three hidden layers, there were two topologies with closed values, therefore, both are displayed.

Table 1: Errors for the best topologies.

Topology	Error Z°	Error X°	Average Error
2700-1000-2	3.23	5.11	4.17
2700-1000-1000-2	2.55	4.12	3.34
2700-1000-1000-400-2	2.47	4.44	3.46
2700-1000-1000-800-2	2.71	4.22	3.47

We noticed that the best result was always the greatest value of the range, so we decided to adjust the ranges as follows:

- 2700 - H1 - 2
H1 $\rightarrow \{800, 900, 1000, \dots, 1900, 2000\}$
- 2700 - H1 - H2 - 2
H1 $\rightarrow \{1000, 1100, 1200, 1300, 1400\}$
H2 $\rightarrow \{600, 800, 900, 1000\}$

- 2700 - H1 - H2 - H3 - 2
H1, H2 = 1000
H3 $\rightarrow \{200, 300, 400, 500, 600, 700, 800\}$

In this case, we collected the best results in Table 2. We noticed that both topologies with three and two hidden layers report similar results. However, it is significant to reduce the number of neurons in our case because of the available number of our training samples, so we selected the topology 2700-1400-800-2 as the most suitable for our application.

Table 2: Errors of the best topologies with extended range.

Topology	Error Z°	Error X°	Average Error
2700-1900-2	3.23	4.49	3.86
2700-1400-800-2	2.48	4.11	3.30
2700-1000-1000-500-2	2.46	4.19	3.33

Figure 1: Testing set predictions of the FCNN 2700-1400-800-2

3.2 Finding regularization term and number of Epochs

With this topology, we moved through the range of the regularization and using the epochs described in section 2.4. We gathered the best results in Table 3. From these, the selected values were the regularization term equals to 3 and 15000 epochs for training. Figure 1 shows the prediction and actual angle of each sample over the testing set as well as its full-scale error. We noticed how most of the error values lay below the average line, but there are some peaks reaching 15% of error. These outliers are a risk for the outcome of our application. They may lead the following registration process

to wrong results, and as a consequence, to perform inaccurate spine procedures. It makes us consider a radical change in our topology by using instead a CNN.

Table 3: Errors of the best regularization term and amount of epochs of the topology 2700-1400-800-2.

Regularization term (λ)	Epochs	Error Z°	Error X°	Average Error
3	15000	1.98	3.06	2.52
3	22000	1.92	3.88	2.9
1	15000	2.17	3.52	2.84
1	22000	2.29	3.6	2.95
1	34000	2.17	3.19	2.68

4 Outlook for CNN

One of the reasons why the full-scale error of our results does not decrease less than 2.68%, could be the reduction of the image resolution before inputting them to the FCNN. Additionally, the FCNN can lack generalization regarding spatial information of the image, i.e., the same image with the same rotations but different translations could be wrongly predicted. For these reasons, we propose using a CNN instead our current FCNN for our initial pose generator.

Initially, images could be entered as full resolution and reduce their resolution by using max-pooling layers. Additional convolutional kernels could be added to extract features. The final stage of the CNN would be a FCNN for doing the final prediction.

For training a bigger net, we will have to use a larger dataset. As a rule of thumb, a deep learning algorithm typically requires 5000 samples for having acceptable performance [5]. Therefore, we must use some data augmentation techniques suitable for deep learning [6]. However, we must carefully evaluate them because applying a common technique as image rotation will spoil the results instead of improving them.

Finally, once the CNN is trained and validated, it can be deployed in C++, which is the language used for our framework, and where the planning software for the spine surgery is done.

5 Discussion and Conclusion

We found out that for our application in spine surgeries, a FCNN with a topology 2700-1400-800-2 gives the best

performance when it is used with a regularization coefficient of 3 and trained in a range of around 15000 epochs.

Our results using a FCNN lay within the range that can be used as the initial pose by our registration algorithm for spine surgery. The average errors were 1.98% in the Z-axis and 3.06% in X-axis, and when using the full-scale conversion, they are equivalent to 3.564° and 2.754 in Z- and X- axis respectively. However, the peaks of the errors reach 15%, which are equivalent to 27° and 13.5° so completely out of the acceptable range. Therefore, we consider that our next improvement will be using CNNs that have been reported to produce better results with image processing. We expect the results with CNN to be more trustworthy and easy to integrate with our current workflow for computer-assisted spine surgery. Additionally, we want to increase our dataset size to have a better description of the problem, and also to train the network correctly. Such a task is time-consuming, yet it is the main reason for our current dataset size.

Author Statement

Conflict of interest: Authors state no conflict of interest. Informed consent: Informed consent has been obtained from all individuals included in this study. Ethical approval: The research related to human use complies with all the relevant national regulations, institutional policies and was performed in accordance with the tenets of the Helsinki Declaration, and has been approved by the authors' institutional review board or equivalent committee.

Acknowledgement: Part of this work is funded by the German Federal Ministry of Education and Research (KMU-innovativ: contract number 13GW0175B)

References

- [1] Miao S, Liao J, Lucas J, et al. Toward accurate and robust 2-D/3-D registration of implant models to single-plane fluoroscopy, in *Augmented Reality Environments for Medical Imaging Computer-Assisted Interventions*, 2013.
- [2] Miao S, Wang Jane, Liao R, A CNN Regression Approach for Real-Time 2D/3D Registration, in *IEEE Transactions on Medical Imaging*, Vol. 35, No. 5. 2016.
- [3] C. Chou, B. Frederick, et al. 2D/3D Image Registration Using Regression Learning. *Comput Vis Image Underst.* 2013 September 1; 117(9): 1095–1106.
- [4] Chollet F, *Deep Learning with Python*, Manning Publication Co. 2018
- [5] Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press.
- [6] Perez L, Wang J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *CoRR* abs/1712.04621