CHAPTER

Convolutional Neural Networks for Robust and Real-Time 2-D/3-D Registration

12

Shun Miao[*], **Jane Z. Wang**[†], **Rui Liao**[*]

Siemens Medical Solutions USA, Inc., Princeton, NJ, United States[*] *University of British Columbia, Vancouver, BC, Canada*[†]

CHAPTER OUTLINE

Deep Learning for Medical Image Analysis. DOI: 10.1016/B978-0-12-810408-8.00016-X

12.1 INTRODUCTION

Two-dimensional (2-D) to three-dimensional (3-D) registration represents one of the key enabling technologies in medical imaging and image-guided interventions [1]. It can bring the pre-operative 3-D data and intra-operative 2-D data into the same coordinate system, to facilitate accurate diagnosis and/or provide advanced image guidance. The pre-operative 3-D data generally includes Computed Tomography (CT), Cone-Beam CT (CBCT), Magnetic Resonance Imaging (MRI), and Computer Aided Design (CAD) model of medical devices, while the intra-operative 2-D data is dominantly X-ray images. In this paper, we focus on registering a 3-D X-ray attenuation map provided by CT or CBCT with a 2-D X-ray image in real-time. Depending on the application, other 3-D modalities (e.g., MRI and CAD model) can be converted to a 3-D X-ray attenuation map before performing 2-D/3-D registration.

Accurate 2-D/3-D registration is typically achieved by intensity-based methods, where a simulated X-ray image, referred to as Digitally Reconstructed Radiograph (DRR), is derived from the 3-D X-ray attenuation map by simulating the attenuation of virtual X-rays, and an optimizer is employed to maximize an intensity-based similarity measure between the DRR and X-ray images [2–5] (see Fig. 12.1). This is indeed an ineffective formulation because the information provided by the DRR and fluoroscopic image are compressed to the scalar-valued similarity measure, while other higher-dimensional information from the images is completely unexploited. The unexploited information includes the appearance of the mismatch, the direction of the offset, etc., which could potentially provide a clue on how the registration should be adjusted to align the two images. In addition, the matching metrics employed are often ineffective in continuously measuring the alignment of the target object in the two images, as they are typically highly non-convex (i.e., have local maxima in false positions), which makes optimization a challenging task. Due to the above shortcomings, intensity-based 2-D/3-D registration methods typically have low computational efficiency (due to iterative optimization over a 1-D metric) and small capture range (due to optimization over a highly non-convex metric).

In intensity-based 2-D/3-D registration, DRRs need to be repeatedly calculated during the iterative optimization, which means heavy computation. Some efforts have been made toward accelerating DRR generation. One strategy for faster DRR generation is sparse sampling, where a subset of the pixels are statistically chosen for DRR rendering and similarity measure calculation [6,7]. However, only a few similarity measures are suitable to be calculated on a random subset of the image, e.g., Mutual Information (MI) [6] and Stochastic Rank Correlation (SRC) [7]. Another strategy is splatting, which is a voxel-based volume rendering technique that directly projects single voxels to the imaging plane [8,9]. Splatting allows us to only use voxels with intensity above certain threshold for rendering, which significantly reduces the number of voxels to be visited. However, one inherent problem of splatting is that, due to aliasing artifacts, the image quality of the generated DRR is significantly degraded compared to the DRR generated by the standard Ray Casting algorithm [10], which subsequently degrades the registration accuracy.

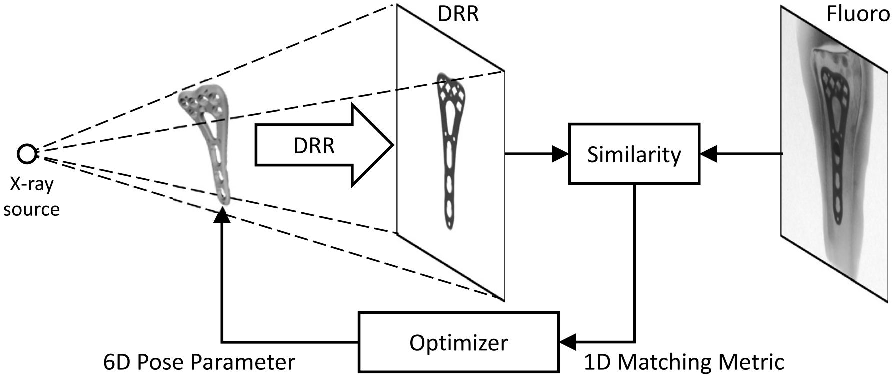


FIGURE 12.1

Formulation of intensity-based 2-D/3-D registration.

Motivated by the success of machine learning in computer vision, supervised learning has also been explored for 2-D/3-D registration. Several metric learning methods have been proposed to learn similarity measures using supervised learning [11,12]. While learned metrics could have better capture range and/or accuracy over general purpose similarity measures on specific applications or image modalities, 2-D/3-D registration methods using learned metrics still fall into the category of intensity-based methods with a high computational cost. As a new direction, several attempts have been made recently toward learning regressors to solve 2-D/3-D registration problems in real-time [13,14]. Gouveia et al. [13] extracted a handcrafted feature from the X-ray image and trained a Multi-Layer Perceptron (MLP) regressor to estimate the 3-D transformation parameters. However, the reported accuracy is much lower than can be achieved using intensity-based methods, suggesting that the handcrafted feature and MLP are unable to accurately recover the underlying complex transformation. Chou et al. [14] computed the residual between the DRR and X-ray images as a feature and trained linear regressors to estimate the transformation parameters to reduce the residual. Since the residual is a low-level feature, the mapping from it to the transformation parameters is highly nonlinear, which cannot be reliably recovered using linear regressors, as will be shown in our experiment.

In recent years, promising results on object matching for computer vision tasks have been reported using machine learning methods [15–18]. While these methods are capable of reliably recovering the object's location and/or pose for computer vision tasks, they are unable to meet the accuracy requirement of 2-D/3-D registration tasks in medical imaging, which often require a very high accuracy (i.e., sub-millimeter) for diagnosis and surgery guidance purposes. For example, Wohlhart et al. [15] proposed to train a Convolutional Neural Network (CNN) to learn a pose differentiating descriptor from range images, and use *k*-Nearest Neighbor for pose estimation. While global pose estimation can be achieved using this method, its ac-

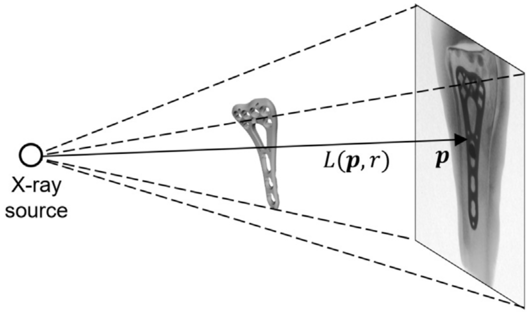


FIGURE 12.2

X-ray imaging geometry.

curacy is relatively low, i.e., the success rate for angle error less than 5 degrees is below 60% for $k = 1$. Dollár et al. [16] proposed to train cascaded regressors on a pose-indexed feature that is only affected by the difference between the ground truth and initial pose parameters for pose estimation. This method aims to solve 2-D pose estimation from RGB images, but is not applicable for 2-D/3-D registration problems.

In this chapter, we propose a CNN regression-based method, referred to as Pose Estimation via Hierarchical Learning (PEHL), to achieve real-time 2-D/3-D registration with a large capture range and high accuracy. The key of our method is to train CNN regressors to recover the mapping from the DRR and X-ray images to the difference of their underlying transformation parameters. This mapping is highly complex, and training regressors to recover the mapping is far from being trivial. In the proposed method, we achieve this by first simplifying the nonlinear relationship using three algorithmic strategies and then capturing the mapping using CNN regressors with a strong nonlinear modeling capability.

12.2 X-RAY IMAGING MODEL

Fig. 12.2 shows the geometry of X-ray imaging. Assuming that the X-ray imaging system corrects the beam divergence and the X-ray sensor has a logarithm static response, X-ray image generation can be described by the following model:

$$I(\boldsymbol{p}) = \int \mu\Big(\boldsymbol{L}(\boldsymbol{p}, r)\Big) dr, \tag{12.1}$$

where $I(\boldsymbol{p})$ is the intensity of the X-ray image at point $\boldsymbol{p}$, $\boldsymbol{L}(\boldsymbol{p}, r)$ is the ray from the X-ray source to point $\boldsymbol{p}$, parameterized by r, and $\mu(\cdot)$ is the X-ray attenuation coefficient. Denoting the X-ray attenuation map of the object to be imaged as $J : \mathbb{R}^3 \to \mathbb{R}$, and the 3-D transformation from the object coordinate system to the X-ray imaging coordinate system as $T : \mathbb{R}^3 \to \mathbb{R}^3$, the attenuation coefficient at point $\boldsymbol{x}$ in the X-ray imaging coordinate system is

$$\mu(\boldsymbol{x}) = J(T^{-1} \circ \boldsymbol{x}). \tag{12.2}$$

Combining Eq. (12.1) and Eq. (12.2), we have

$$I(\boldsymbol{p}) = \int J\Big(T^{-1} \circ \boldsymbol{L}(\boldsymbol{p}, r)\Big) dr. \tag{12.3}$$

In 2-D/3-D registration problems, $\boldsymbol{L}$ is determined by the X-ray imaging system, J is provided by the 3-D data (e.g., CT intensity), and the transformation T is to be estimated from the input X-ray image I. Given J, $\boldsymbol{L}$, and T, a synthetic X-ray image $I(\cdot)$ (i.e., DRR) can be computed following Eq. (12.3) using the Ray-casting algorithm [10].

12.3 PROBLEM FORMULATION

Given the ineffectiveness of intensity-based formulation of 2-D/3-D registration, we propose a more effective regression-based formulation, where a CNN regressor takes the DRR and fluoroscopic image as input, and produces an update of the transformation parameters. Instead of converting the two images to a 1-D metric, in the regression-based formulation, we employ CNNs to learn representations from the image intensities, and reveal the mapping from the representations to the parameter updates. Using this formulation, the information in the DRR and fluoroscopic image can be fully exploited to guide the update of the transformation parameters. The regression-based formulation is mathematically described as follows.

Based on Eq. (12.3), we denote the X-ray image with transformation parameters $\boldsymbol{t}$ as $I_{\boldsymbol{t}}$, where the variables $\boldsymbol{L}$ and J are omitted for simplicity because they are non-varying for a given 2-D/3-D registration task. The inputs for 2-D/3-D registration are: (i) a 3-D object described by its X-ray attenuation map J, (ii) an X-ray image $I_{\boldsymbol{t}_{gt}}$, where $\boldsymbol{t}_{gt}$ denotes the unknown ground truth transformation parameters, and (iii) initial transformation parameters $\boldsymbol{t}_{ini}$. The 2-D/3-D registration problem is formulated as a regression problem, where a set of regressors $\boldsymbol{f}(\cdot)$ are trained to reveal the mapping from a feature $X(\boldsymbol{t}_{ini}, I_{\boldsymbol{t}_{gt}})$ extracted from the inputs to the parameter residuals, $\boldsymbol{t}_{gt} - \boldsymbol{t}_{ini}$, as long as it is within a capture range $\boldsymbol{\epsilon}$:

$$\boldsymbol{t}_{gt} - \boldsymbol{t}_{ini} \approx \boldsymbol{f}\big(X(\boldsymbol{t}_{ini}, I_{\boldsymbol{t}_{gt}})\big), \quad \forall \boldsymbol{t}_{gt} - \boldsymbol{t}_{ini} \in \boldsymbol{\epsilon}. \tag{12.4}$$

An estimate of $\boldsymbol{t}_{gt}$ is then obtained by applying the regressors and incorporating the estimated parameter residuals into $\boldsymbol{t}_{ini}$:

$$\hat{\boldsymbol{t}}_{gt} = \boldsymbol{t}_{ini} + \boldsymbol{f}\big(X(\boldsymbol{t}_{ini}, I_{\boldsymbol{t}_{gt}})\big). \tag{12.5}$$

It is worth noting that the range $\boldsymbol{\epsilon}$ in Eq. (12.4) is equivalent to the capture range of optimization-based registration methods. Based on Eq. (12.4), our problem formu-

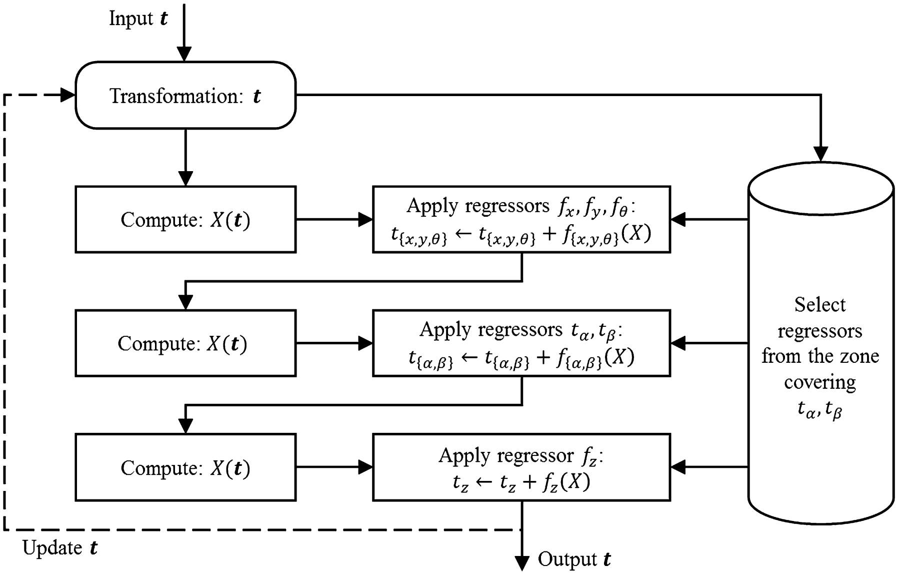

Input $\boldsymbol{t}$
Transformation: $\boldsymbol{t}$
Compute: $X(\boldsymbol{t})$
Apply regressors f_x, f_y, f_θ:
$t_{\{x,y,\theta\}} \leftarrow t_{\{x,y,\theta\}} + f_{\{x,y,\theta\}}(X)$
Compute: $X(\boldsymbol{t})$
Apply regressors t_α, t_β:
$t_{\{\alpha,\beta\}} \leftarrow t_{\{\alpha,\beta\}} + f_{\{\alpha,\beta\}}(X)$
Compute: $X(\boldsymbol{t})$
Apply regressor f_z:
$t_z \leftarrow t_z + f_z(X)$
Select regressors from the zone covering t_α, t_β
Update $\boldsymbol{t}$
Output $\boldsymbol{t}$

FIGURE 12.3

Workflow of the proposed regression strategy.

lation can be expressed as designing a feature extractor $X(\cdot)$ and training regressors $\boldsymbol{f}(\cdot)$ such that

$$\delta \boldsymbol{t} \approx \boldsymbol{f}\big(X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta \boldsymbol{t}})\big), \quad \forall \delta \boldsymbol{t} \in \boldsymbol{\epsilon}. \tag{12.6}$$

In the next section, we will discuss in detail (i) how the feature $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta \boldsymbol{t}})$ is calculated and (ii) how the regressors $\boldsymbol{f}(\cdot)$ are designed, trained, and applied.

12.4 REGRESSION STRATEGY

In this section, we describe the proposed regression strategy. We first partition the parameter space into multiple zones, for which the regressors are trained and applied separately. We then divide the parameters into three groups and regress them in marginal spaces. Fig. 12.3 shows the workflow of the proposed regression strategy.

12.4.1 PARAMETER SPACE PARTITIONING

We parameterize the transformation by 3 in-plane and 3 out-of-plane transformation parameters [19]. In particular, in-plane transformation parameters include 2 translation parameters, t_x and t_y, and 1 rotation parameter, t_θ. The effects of in-plane transformation parameters are approximately 2-D rigid-body transformations. Out-of-plane transformation parameters include 1 out-of-plane translation parameter, t_z, and 2 out-of-plane rotation parameters, t_α and t_β. The effects of out-of-plane translation and rotations are scaling and shape changes, respectively.

To simplify the problem, partition the 360×360 degrees parameter space spanned by t_α and t_β into an 18×18 grid (empirically selected in our experiment). Each square in the grid covers a 20×20 degrees area, and is referred to as a *zone*. For each zone, the parameter regressors are trained separately, so that each regressor only need to solve a constrained and simplified problem. In the application stage, the regressors corresponding to the current t_α and t_β (provided as initial registration parameters) are retrieved and applied.

12.4.2 MARGINAL SPACE REGRESSION

Instead of jointly regressing the 6 parameters together, they are divided into 3 groups, and regressed in the marginal space, which reduces confounding factors in the regression tasks and hence simplifies the problem. The 3 groups of parameters are defined as follows:

- *Group 1*. In-plane parameters, δt_x, δt_y, δt_θ
- *Group 2*. Out-of-plane rotation parameters, δt_α, δt_β
- *Group 3*. Out-of-plane translation parameter, δt_z

Among the 3 groups, the parameters in Group 1 are considered to be the easiest to be estimated because they cause simple while dominant rigid-body 2-D transformation of the object in the projection image that is less affected by the variations of the parameters in the other two groups. The parameter in Group 3 is the most difficult to be estimated because it only causes subtle scaling of the object in the projection image. The difficulty in estimating parameters in Group 2 falls in-between. Therefore we regress the 3 groups of parameters sequentially, from the easiest group to the most difficult. After a group of parameters are regressed, the feature $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta \boldsymbol{t}})$ is re-calculated using the already-estimated parameters for the regression of the parameters in the next group. This way the mapping to be regressed for each group is simplified by limiting the dimension and removing the compounding factors coming from those parameters in the previous groups.

12.5 FEATURE EXTRACTION

12.5.1 LOCAL IMAGE RESIDUAL

We calculate the residual between DRR rendered using transformation parameters $\boldsymbol{t}$, denoted by $I_{\boldsymbol{t}}$ and the X-ray image $I_{\boldsymbol{t}+\delta \boldsymbol{t}}$ in local patches of attention as the input feature for regression. To determine the locations, sizes, and orientations of the local patches of attention, a number of 3-D points of interest are extracted from the 3-D model of the target object following the steps described in Section 12.5.2. Given a point $\boldsymbol{p}$ and parameters $\boldsymbol{t}$, a square local ROI is uniquely determined in the 2-D

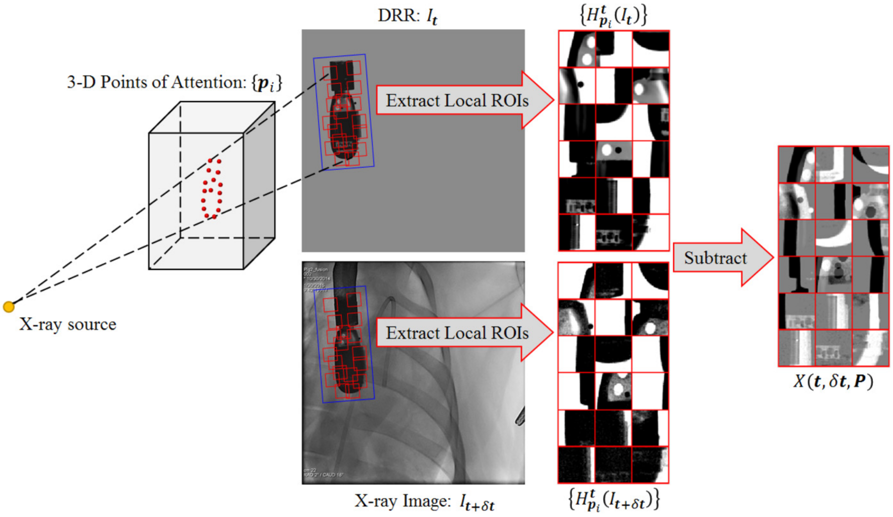


FIGURE 12.4

Workflow of LIR feature extraction, demonstrated on X-ray Echo Fusion data. The local ROIs determined by the 3-D points $\boldsymbol{P}$ and the transformation parameters $\boldsymbol{t}$ are shown as red boxes. The blue box shows a large ROI that covers the entire object, used in compared methods as will be discussed in Section 12.7.3.

imaging plane, which can be described by a triplet, $(\boldsymbol{q}, w, \phi)$, denoting the ROI's center, width, and orientation, respectively. The center $\boldsymbol{q}$ is the 2-D projection of $\boldsymbol{p}$ using transformation parameters $\boldsymbol{t}$. The width $w = w_0 \cdot D/t_z$, where w_0 is the size of the ROI in mm and D is the distance between the X-ray source and detector. The orientation $\phi = t_\theta$, so that it is always aligned with the object. We define an operator $H^{\boldsymbol{t}}_{\boldsymbol{p}}(\cdot)$ that extracts the image patch in the ROI determined by $\boldsymbol{p}$ and $\boldsymbol{t}$, and re-sample it to a fixed size (52 × 52 in our applications). Given N 3-D points, $\boldsymbol{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$, the LIR feature is then computed as

$$X(\boldsymbol{t}, I_{t+\delta t}, \boldsymbol{P}) = \left\{H^{\boldsymbol{t}}_{\boldsymbol{p}_i}(I_{\boldsymbol{t}}) - H^{\boldsymbol{t}}_{\boldsymbol{p}_i}(I_{\boldsymbol{t}+\delta \boldsymbol{t}})\right\}_{i=1,\ldots,N}. \tag{12.7}$$

In a local area of $I_{\boldsymbol{t}}$, the effect of varying t_α and t_β within a zone is approximately a 2-D translation. Therefore, by extracting local patches from ROIs selected based on $\boldsymbol{t}$, the effects of all 6 transformation parameters in $\boldsymbol{t}$ are compensated, making $H^{\boldsymbol{t}}_{\boldsymbol{p}}(I_{\boldsymbol{t}})$ approximately invariant to $\boldsymbol{t}$. Since the difference between $H^{\boldsymbol{t}}_{\boldsymbol{p}}(I_{\boldsymbol{t}+\delta \boldsymbol{t}})$ and $H^{\boldsymbol{t}}_{\boldsymbol{p}}(I_{\boldsymbol{t}})$ is merely additional 2-D transformation caused by $\delta \boldsymbol{t}$, $H^{\boldsymbol{t}}_{\boldsymbol{p}}(I_{\boldsymbol{t}+\delta \boldsymbol{t}})$ is also approximately invariant to $\boldsymbol{t}$. The workflow of LIR feature extraction is shown in Fig. 12.4.

12.5.2 3-D POINTS OF INTEREST

The 3-D points of interest used for calculating the LIR feature are extracted separately for each zone in two steps. First, 3-D points that correspond to 2-D edges are extracted as candidates. Specifically, the candidates are extracted by thresholding pixels with high gradient magnitudes in a synthetic X-ray image (i.e., generated using DRR) with t_α and t_β at the center of the zone, and then back-projecting them to the corresponding 3-D structures. The formation model of gradients in X-ray images has been shown in [20] as

$$g(\boldsymbol{p}) = \int \eta\Big(\boldsymbol{L}(\boldsymbol{p}, r)\Big) dr, \tag{12.8}$$

where $g(\boldsymbol{p})$ is the magnitude of the X-ray image gradient at the point $\boldsymbol{p}$, and $\eta(\cdot)$ can be computed from $\mu(\cdot)$ and the X-ray perspective geometry [20]. We back-project $\boldsymbol{p}$ to $\boldsymbol{L}(\boldsymbol{p}, r_0)$, where

$$r_0 = \arg\max_r \boldsymbol{L}(\boldsymbol{p}, r), \tag{12.9}$$

if

$$\int\limits_{r_0-\sigma}^{r_0+\sigma} \eta\Big(\boldsymbol{L}(\boldsymbol{p}, r)\Big) dr \geq 0.9 \cdot g(\boldsymbol{p}). \tag{12.10}$$

The condition in Eq. (12.10) ensures that the 3-D structure around $\boldsymbol{L}(\boldsymbol{p}, r_0)$ "essentially generates" the 2-D gradient $g(\boldsymbol{p})$, because the contribution of $\eta(\cdot)$ within a small neighborhood (i.e., $\sigma = 2$ mm) of $\boldsymbol{L}(\boldsymbol{p}, r_0)$ leads to the majority (i.e., $\geq 90\%$) of the magnitude of $g(\boldsymbol{p})$. In other words, we find the dominant 3-D structure corresponding to the gradient in the X-ray image.

Second, the candidates are filtered so that only those leading to the most discriminative LIRs are kept. To achieve this, we randomly generate $\{\boldsymbol{t}_j\}_{j=1}^M$ with t_α and t_β within the zone and $\{\delta\boldsymbol{t}_k\}_{k=1}^M$ within the capture range $\boldsymbol{\epsilon}$ ($M = 1000$ in our applications). The intensity of the nth pixel of $H_{\boldsymbol{p}_i}^{\boldsymbol{t}_j}(I_{\boldsymbol{t}_j}) - H_{\boldsymbol{p}_i}^{\boldsymbol{t}_j}(I_{\boldsymbol{t}_j+\delta\boldsymbol{t}_k})$ is denoted as $h_{n,i,j,k}$. The following two measurements are computed for all candidates:

$$E_i = \Big\langle (h_{n,i,j,k} - \langle h_{n,i,j,k}\rangle_j)^2 \Big\rangle_{n,j,k}, \tag{12.11}$$

$$F_i = \Big\langle (h_{n,i,j,k} - \langle h_{n,i,j,k}\rangle_k)^2 \Big\rangle_{n,j,k}, \tag{12.12}$$

where $\langle\cdot\rangle$ is an average operator with respect to all indexes in the subscript. Since E_i and F_i measure the sensitivity of $H_{\boldsymbol{p}_i}^{\boldsymbol{t}}(I_{\boldsymbol{t}}) - H_{\boldsymbol{p}_i}^{\boldsymbol{t}}(I_{\boldsymbol{t}+\delta\boldsymbol{t}})$ with respect to $\boldsymbol{t}$ and $\delta\boldsymbol{t}$, respectively, an ideal LIR should have a small E_i so that it is less affected by $\boldsymbol{t}$ and a large F_i for regressing $\delta\boldsymbol{t}$. Therefore, the candidate list is filtered by picking the candidate with the largest F_i/E_i in the list, and then removing other candidates with ROIs that have more than 25% overlapping area. This process repeats until the list is empty.

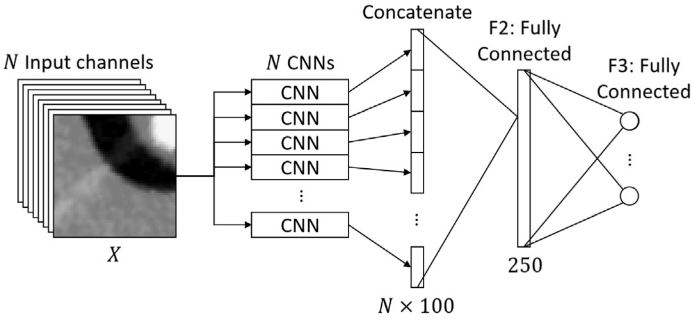


FIGURE 12.5

Structure of the CNN regression model.

12.6 CONVOLUTIONAL NEURAL NETWORK

In our regression problem, there are two challenges in designing the CNN regression model: (i) it needs to be flexible enough to capture the complex mapping from $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}})$ to $\delta\boldsymbol{t}$, and (ii) it needs to be light-weighted enough to be forwarded in real-time and stored in Random-Access Memory (RAM). Managing memory footprint is particularly important because regressors for all zones (in total 324) need to be loaded to RAM for optimal speed. Another challenge is the training of CNN requires a large number of labeled data, which is often difficult to obtain for many clinical applications. We employ the following CNN regression model and training procedures to address the above challenges.

12.6.1 NETWORK STRUCTURE

A CNN regression model with the architecture shown in Fig. 12.5 is trained for each group in each zone. According to Eq. (12.7), the input of the regression model consists of N channels, corresponding to N LIRs. The CNN shown in Fig. 12.6 is applied on each channel for feature extraction. The CNN consists of five layers, including two 5×5 convolutional layers (C1 and C2), each followed by a 2×2 max-pooling layers (P1 and P2) with stride 2, and a fully-connected layer (F1) with 100 Rectified Linear Unit (ReLU) activations neurons. The feature vectors extracted from all input channels are then concatenated and connected to another fully-connected layer (F2) with 250 ReLU activations neurons. The output layer (F3) is fully-connected to F2, with each output node corresponding to one parameter in the group. Since all input channels are LIR, it is reasonable to extract feature from them using the same CNN. Therefore, the feature extraction CNNs in Fig. 12.5 share weights.

In our experiment, we empirically selected the size of the ROI, which led to $N \approx 18$. Using the CNN model shown in Fig. 12.5 with weight sharing, there are 660,500 weights in total for each group in each zone, excluding the output layer, which only has $250 \times N_t$ weights, where N_t is the number of parameters in the group.

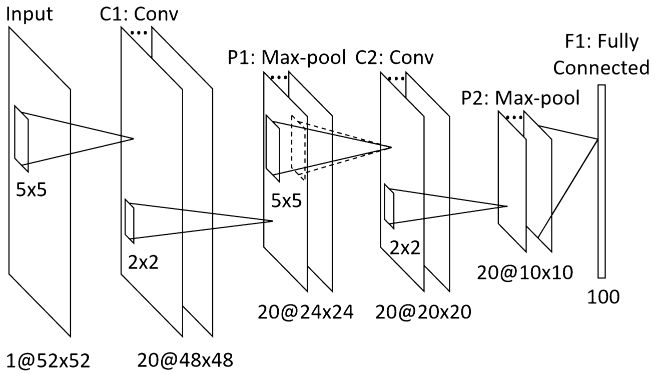


FIGURE 12.6

Structure of the CNN applied for each input channel.

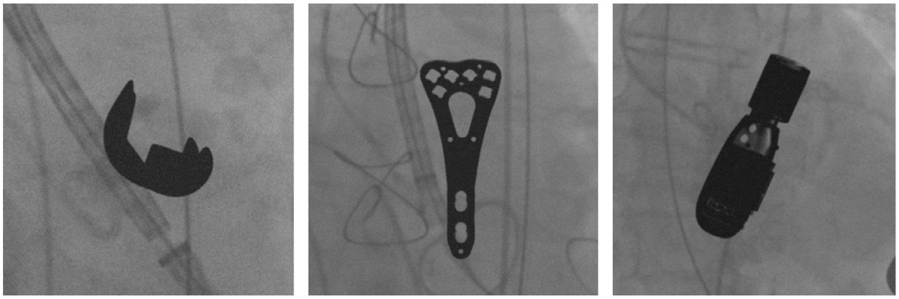
If the weights are stored as 32-bit float, around 2.5 MB is required for each group in each zone. Given 3 groups and 324 zones, there are 972 CNN regression models in total, and pre-loading all of them into RAM requires 2.39 GB, which is manageable for modern computers.

12.6.2 TRAINING DATA

The CNN regression models are trained exclusively on synthetic X-ray images, because they provide reliable ground truth labels without the need of laborious manual annotation, and the amount of real X-ray images could be limited. The synthetic X-ray images were generated by blending a DRR of the object with a background from real X-ray images:

$$I = I_{Xray} + \gamma \cdot G_\sigma * I_{DRR} + \mathcal{N}(a, b), \qquad (12.13)$$

where I_{Xray} is the real X-ray image, I_{DRR} is the DRR, G_σ denotes a Gaussian smoothing kernel with a standard deviation σ simulating X-ray scattering effect, $f * g$ denotes the convolution of f and g, γ is the blending factor, and $\mathcal{N}(a, b)$ is a random noise uniformly distributed on $[a, b]$. The parameters (γ, σ, a, b) were empirically tuned for each object (i.e., implants and TEE probe) to make the appearance of the synthetic X-ray image realistic. These parameters were also randomly perturbed within a neighborhood for each synthetic X-ray image to increase the variation of the appearance of the synthetic X-ray images, so that the regressors trained on them can be generalized well on real X-ray images. The background image used for a given synthetic image was randomly picked from a group of real X-ray images irrespective of the underlying clinical procedures so that the trained network would not be over-fitted for any specific type of background, which could vary sig-



A TKA B VIPS C XEF

FIGURE 12.7

Example synthetic X-ray images used for training.

Table 12.1 Distributions of randomly generated $\delta \boldsymbol{t}$. $\mathcal{U}(a, b)$ denotes the uniform distribution between a and b. The units for translation and rotations are mm and degree, respectively

Group 1	Group 2	Group 3
$\delta t_x \sim \mathcal{U}(-1.5, 1.5)$	$\delta t_x \sim \mathcal{U}(-0.2, 0.2)$	$\delta t_x \sim \mathcal{U}(-0.15, 0.15)$
$\delta t_y \sim \mathcal{U}(-1.5, 1.5)$	$\delta t_y \sim \mathcal{U}(-0.2, 0.2)$	$\delta t_y \sim \mathcal{U}(-0.15, 0.15)$
$\delta t_z \sim \mathcal{U}(-15, 15)$	$\delta t_z \sim \mathcal{U}(-15, 15)$	$\delta t_z \sim \mathcal{U}(-15, 15)$
$\delta t_\theta \sim \mathcal{U}(-3, 3)$	$\delta t_\theta \sim \mathcal{U}(-0.5, 0.5)$	$\delta t_\theta \sim \mathcal{U}(-0.5, 0.5)$
$\delta t_\alpha \sim \mathcal{U}(-15, 15)$	$\delta t_\alpha \sim \mathcal{U}(-15, 15)$	$\delta t_\alpha \sim \mathcal{U}(-0.75, 0.75)$
$\delta t_\beta \sim \mathcal{U}(-15, 15)$	$\delta t_\beta \sim \mathcal{U}(-15, 15)$	$\delta t_\beta \sim \mathcal{U}(-0.75, 0.75)$

nificantly from case to case clinically. Examples of synthetic X-ray images are shown in Fig. 12.7.

For each group in each zone, we randomly generate 25,000 pairs of $\boldsymbol{t}$ and $\delta \boldsymbol{t}$. The parameters $\boldsymbol{t}$ follow a uniform distribution with t_α and t_β constrained in the zone. The parameter errors $\delta \boldsymbol{t}$ also follow a uniform distribution, while 3 different distribution ranges are used for the 3 groups, as shown in Table 12.1. The distribution ranges of $\delta \boldsymbol{t}$ for Group 1 are the target capture range that the regressors are designed for. The distribution ranges of δt_x, δt_y, and δt_θ are reduced for Group 2, because they are reduced by applying the regressors in the first group. For the same reason, the distribution ranges of δt_α and t_β are reduced for Group 3. For each pair of $\boldsymbol{t}$ and $\delta \boldsymbol{t}$, a synthetic X-ray image $I_{\boldsymbol{t}+\delta \boldsymbol{t}}$ is generated, and the feature $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta \boldsymbol{t}})$ is calculated following Eq. (12.7).

12.6.3 SOLVER

The objective function to be minimized during the training is Euclidean loss, defined as

$$\Phi = \frac{1}{K}\sum_{i=1}^{K} \|\boldsymbol{y}_i - \boldsymbol{f}(X_i; \mathbf{W})\|_2^2, \tag{12.14}$$

where K is the number of training samples, $\boldsymbol{y}_i$ is the label for the ith training sample, $\mathbf{W}$ is a vector of weights to be learned, $\boldsymbol{f}(X_i; \mathbf{W})$ is the output of the regression model parameterized by $\mathbf{W}$ on the ith training sample. The weights $\mathbf{W}$ are learned using Stochastic Gradient Descent (SGD) [21], with a batch size of 64, momentum of $m = 0.9$, and weight decay of $d = 0.0001$. The update rule for $\mathbf{W}$ is:

$$\mathbf{V}_{i+1} := m \cdot \mathbf{V}_i - d \cdot \kappa_i \cdot \mathbf{W}_i - \kappa_i \cdot \left\langle \frac{\partial \Phi}{\partial \mathbf{W}}\bigg|_{\mathbf{W}_i} \right\rangle_{D_i}, \tag{12.15}$$

$$\mathbf{W}_{i+1} := \mathbf{W}_i + \mathbf{V}_{i+1}, \tag{12.16}$$

where i is the iteration index, $\mathbf{V}$ is the momentum variable, κ_i is the learning rate at the ith iteration, and $\left\langle \frac{\partial \Phi}{\partial \mathbf{W}}\big|_{\mathbf{W}_i} \right\rangle_{D_i}$ is the derivative of the objective function computed on the ith batch D_i with respect to $\mathbf{W}$, evaluated at $\mathbf{W}_i$. The learning rate κ_i is decayed in each iteration following

$$\kappa_i = 0.0025 \cdot (1 + 0.0001 \cdot i)^{-0.75}. \tag{12.17}$$

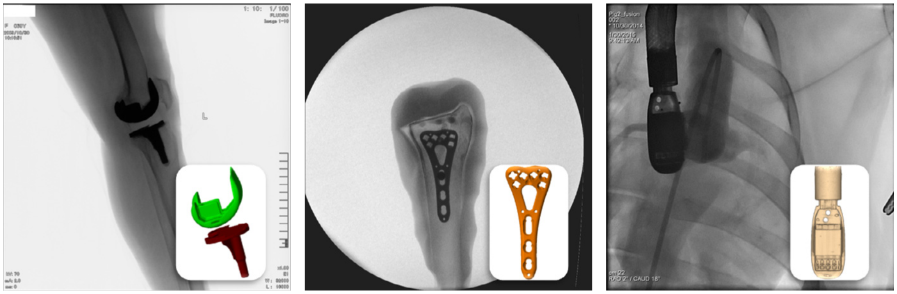
The derivative $\frac{\partial \Phi}{\partial \mathbf{W}}$ is calculated using back-propagation. For weights shared in multiple paths, their derivatives in all paths are back-propagated separately and summed up for the weight update. The weights are initialized using the Xavier method [22], and mini-batch SGD is performed for 12,500 iterations (32 epochs).

12.7 EXPERIMENTS AND RESULTS

12.7.1 EXPERIMENT SETUP

We conducted experiments on datasets from the following 3 clinical applications:

1. *Total Knee Arthroplasty (TKA) kinematics*. In the study of the kinematics of TKA, 3-D kinematics of knee prosthesis can be estimated by matching the 3-D model of the knee prosthesis with the fluoroscopic video of the prosthesis using 2-D/3-D registration [23]. We evaluated PEHL on a fluoroscopic video consisting of 100 X-ray images of a patient's knee joint taken at the phases from full extension to maximum flexion after TKA. The size of the X-ray images is 1024 × 1024 with a pixel spacing of 0.36 mm. A 3-D surface model of the prosthesis was acquired by a laser scanner, and was converted to a binary volume for registration.
2. *Virtual Implant Planning System (VIPS)*. VIPS is an intraoperative application that was established to facilitate the planning of implant placement in terms of orientation, angulation and length of the screws [24]. In VIPS, 2-D/3-D registration is performed to match the 3-D virtual implant with the fluoroscopic image of



A TKA B VIPS C XEF

FIGURE 12.8

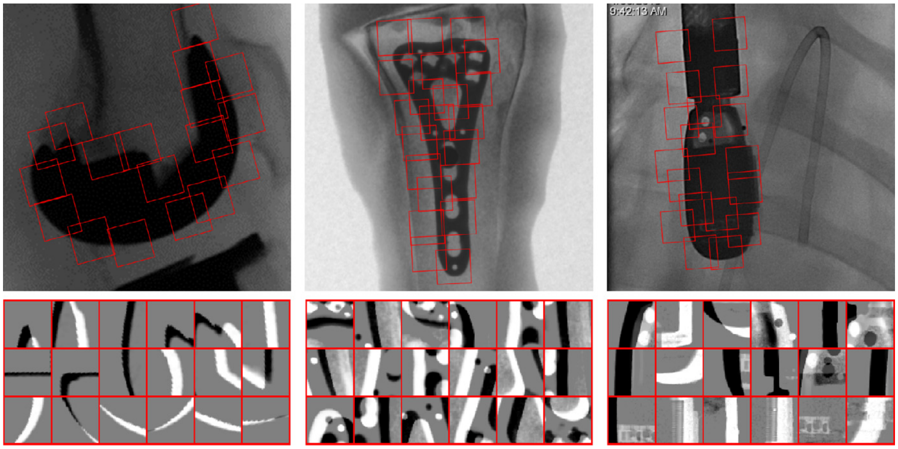
Example data, including a 3-D model and a 2-D X-ray image of the object.

the real implant. We evaluated PEHL on 7 X-ray images of a volar plate implant mounted onto a phantom model of the distal radius. The size of the X-ray images is 1024 × 1024 with a pixel spacing of 0.223 mm. A 3-D CAD model of the volar plate was converted to a binary volume for registration.

3. *X-ray Echo Fusion (XEF).* 2-D/3-D registration can be applied to estimate the 3-D pose of a transesophageal echocardiography (TEE) probe from X-ray images, which brings the X-ray and TEE images into the same coordinate system and enables the fusion of the two modalities [25]. We evaluated PEHL on 2 fluoroscopic videos with in total 94 X-ray images acquired during an animal study using a Siemens Artis Zeego C-Arm system. The size of the X-ray images is 1024 × 1024 with a pixel spacing of 0.154 mm. A micro-CT scan of the Siemens TEE probe was used for registration.

Example datasets of the above 3 clinical applications are shown in Fig. 12.8. Examples of local ROIs and LIRs extracted from the 3 datasets are also shown in Fig. 12.9.

Ground truth transformation parameters used for quantifying registration error were generated by first manually registering the target object and then applying an intensity-based 2-D/3-D registration method using Powell's method combined with Gradient Correlation (GC) [9]. Perturbations of the ground truth were then generated as initial transformation parameters for 2-D/3-D registration. For TKA and XEF, 10 perturbations were generated for each X-ray image, leading to 1000 and 940 test cases, respectively. Since the number of X-ray images for VIPS is limited (i.e., 7), 140 perturbations were generated for each X-ray image to create 980 test cases. The perturbation for each parameter followed the normal distribution with a standard deviation equal to 2/3 of the training range of the same parameter (i.e., Group 1 in Table 12.1). In particular, the standard deviations for $(t_x, t_y, t_z, t_\theta, t_\alpha, t_\beta)$ are 1 mm, 1 mm, 10 mm, 2 degrees, 10 degrees, 10 degrees, respectively. With this distribution, 42.18% of the perturbations have all 6 parameters within the training range, while the other 57.82% have at least one parameter outside of the training range.



A TKA B VIPS C XEF

FIGURE 12.9

Examples of local ROIs and LIRs.

The registration accuracy was assessed with the mean Target Registration Error in the projection direction (mTREproj) [26], calculated at the 8 corners of the bounding box of the target object. We regard mTREproj less than 1% of the size of the target object (i.e., diagonal of the bounding box) as a successful registration. For TKA, VIPS, and XEF, the sizes of the target objects are 110, 61, and 37 mm, respectively. Therefore, the success criterion for the three applications was set to mTREproj less than 1.10, 0.61, and 0.37 mm, which is equivalent to 2.8, 3.7, and 3.5 pixels on the X-ray image, respectively. Success rate was defined as the percentage of successful registrations. Capture range was defined as the initial mTREproj for which 95% of the registrations were successful [26]. Capture range is only reported for experiments where there are more than 20 samples within the capture range.

12.7.2 HARDWARE & SOFTWARE

The experiments were conducted on a workstation with Intel Core i7-4790k CPU, 16 GB RAM and Nvidia GeForce GTX 980 GPU. For intensity-based methods, the most computationally intensive component, DRR renderer, was implemented using the Ray-casting algorithm with hardware-accelerated 3-D texture lookups on GPU. Similarity measures were implemented in C++ and executed in a single CPU core. Both DRRs and similarity measures were only calculated within an ROI surrounding the target object, for better computational efficiency. In particular, ROIs of size 256 × 256, 512 × 512, and 400 × 400 were used for TKA, VIPS, and XEF, respectively. For PEHL, the neural network was implemented with cuDNN acceleration using an open-source deep learning framework, Caffe [27].

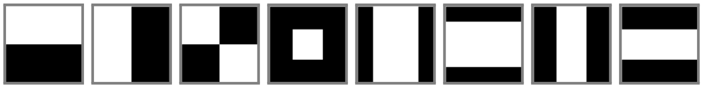


FIGURE 12.10

HAAR features used in the experiment "Without CNN".

12.7.3 PERFORMANCE ANALYSIS

We first conducted the following experiments for detailed analysis of the performance and property of PEHL. The dataset from XEF was used for the demonstration of performance analysis because the structure of the TEE probe is more complex than the implants in TKA and VIPS, leading to an increased difficulty for an accurate registration. As described in Section 12.4.2, PEHL can be applied for multiple iterations. We demonstrate the impact of the number of iterations on performance, by applying PEHL for 10 iterations and showing the registration success rate after each iteration. We also demonstrate the importance of the individual core components of PEHL, i.e., the CNN regression model and 3 algorithmic strategies, Local Image Residual (LIR), Marginal Space Regression (MSR), and Parameter Space Partitioning (PSP), by disabling them and demonstrating the detrimental effects on performance. The following 4 scenarios were evaluated for 10 iterations to compare with PEHL:

- **Without CNN.** We implemented a companion algorithm using HAAR feature with Regression Forest as an alternative to the proposed CNN regression model. We extract 8 HAAR features as shown in Fig. 12.10 from the same training data used for training the CNNs. We mainly used edge and line features because $\delta \boldsymbol{t}$ largely corresponds to lines and edges in LIR. On these HAAR features, we trained a Regression Forest with 500 trees.
- **Without LIR.** A global image residual covering the whole object was used as the input for regression (shown in Fig. 12.4 as blue boxes). The CNN regression model was adapted accordingly. It has five hidden layers: two 5 × 5 convolutional layers, each followed by a 3 × 3 max-pooling layer with stride 3, and a fully-connected layer with 250 ReLU activation neurons. For each group in each zone, the network was trained on the same dataset used for training PEHL.
- **Without MSR.** The proposed CNN regression model shown in Fig. 12.6 was employed, but the output layer has 6 nodes, corresponding to the 6 parameters for rigid-body transformation. For each zone, the network was trained on the dataset used for training PEHL for Group 1.
- **Without PSP.** For each group of parameters, one CNN regression model was applied for the whole parameter space. Because LIP cannot be applied without PSP, the CNN regression model described in "without LIR" was employed in this scenario. The network was trained on 500,000 synthetic training samples with t_α and t_β uniformly distributed in the parameter space.

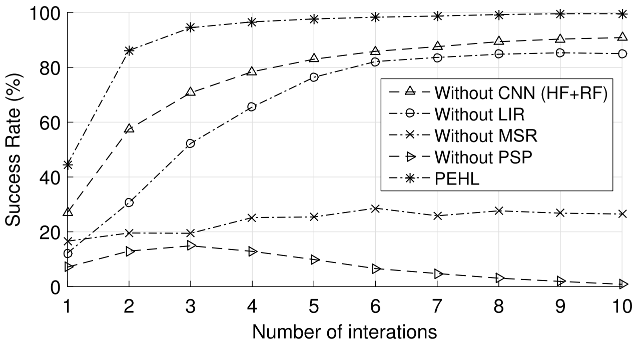


FIGURE 12.11

Success rates of PEHL with 1 to 10 iterations. Four individual core components of PEHL, i.e., CNN, LIR, MSR, and PSP, were disabled one at a time to demonstrate their detrimental effects on performance. Harr feature (HR) + Regression Forest (RF) was implemented to show the effect on performance without CNN. These results were generated on the XEF dataset.

We also conducted an experiment to analyze the precision of PEHL (i.e., the ability to generate consistent results starting from different initial parameters). To measure the precision, we randomly selected an X-ray image from the XEF dataset, and generated 100 perturbations of the ground truth following the same distribution described in Section 12.7.1. PEHL and the best performed intensity-based method, MI_GC_Powell (which will be detailed in Section 12.7.4), were applied starting from the 100 perturbations. The precision of the registration method was then quantified by the root mean squared distance in the projection direction (RMSDproj) from the registered locations of each target to their centroid. Smaller RMSDproj indicates higher precision.

12.7.3.1 *Results*

Fig. 12.11 shows the success rate as the number of iterations increases from 1 to 10 for the five analyzed scenarios. The results show that the success rate of PEHL increased rapidly in the first 3 iterations (i.e., from 44.6% to 94.8%), and kept rising slowly afterward until 9 iterations (i.e., to 99.6%). The computation time of PEHL is linear to the number of iterations, i.e., each iteration takes ~34 ms. Therefore, applying PEHL for 3 iterations is the optimal setting for the trade-off between accuracy and efficiency, which achieves close to the optimal success rate and a real-time registration of ~10 frames per second (fps). Therefore, in the rest of the experiment, PEHL was tested with 3 iterations unless stated otherwise.

The results show that the 3 proposed strategies, LIR, MSR, and PSP, and the use of CNN all noticeably contributed to the final registration accuracy of PEHL. In particular, if the CNN regression model is replaced with HAAR feature + Regression Forest, the success rate at the third iteration dropped to 70.7%, indicating that the

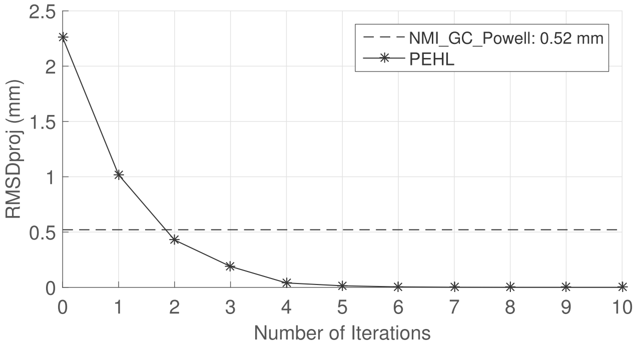


FIGURE 12.12

RMSDproj from the registered locations of each target to their centroid using MI_GC_Powell and PEHL with 1 to 10 iterations. At Number of Iterations = 0, the RMSEproj at the perturbed positions without registration is shown. These results were generated on the XEF dataset.

strong nonlinear modeling capability of CNN is critical to the success of PEHL. If the LIP is replaced with a global image residual, the success rate at the third iteration dropped significantly to 52.2%, showing that LIR is a necessary component to simplify the target mapping so that it can be robustly regressed with the desired accuracy. When MSR and PSP are disabled, the system almost completely failed, dropping the success rate at the third iteration to 19.5% and 14.9%, respectively, suggesting that MSR and PSP are key components that make the regression problem solvable using the proposed CNN regression model.

Fig. 12.12 shows the RMSDproj from registered target points to their corresponding centroid using both MI_GC_Powell and PEHL with 1 to 10 iterations. The results show that as the number of iteration increases, the RMSDproj of PEHL approaches zero, indicating that with sufficient number of iterations, PEHL can reliably reproduce the same result starting from different positions (e.g., 6 iterations lead to RMSEproj = 0.005 mm). At the third iteration, the RMSDproj of PEHL is 0.198 mm, which is 62% smaller than that of MI_GC_Powell, i.e., 0.52 mm. In Fig. 12.13, we show the distribution of a landmark at the center of the imaging cone of the TEE probe before registration, after registration using MI_GC_Powell and after 3 iterations of PEHL. Both the RMSDproj result and the distribution of the landmarks suggest that PEHL has a significant advantage over MI_GC_Powell in terms of precision.

12.7.4 COMPARISON WITH STATE-OF-THE-ART METHODS

We also conducted experiments to compare PEHL with state-of-the-art 2-D/3-D registration methods, including several variations of intensity-based methods and a recent linear regression-based method.

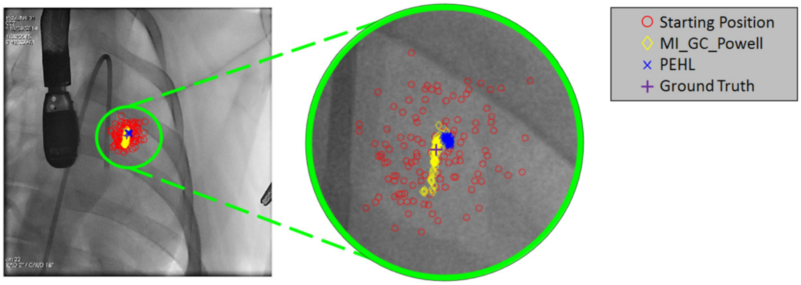


FIGURE 12.13

Distribution of projections of a landmark after registration using MI_GC_Powell and PEHL.

12.7.4.1 *Evaluated Methods*

We first evaluated intensity-based 2-D/3-D registration methods, where we have multiple options optimizer and similarity measure. We used Powell's method as the optimizer, as it is shown to be the most effective optimizer for intensity-based 2-D/3-D registration in a recent study [28]. We evaluated three popular similarity measures, MI, Cross-Correlation (CC), and GC, which have also been reported to be effective in recent literature [3,4,9]. The above three intensity-based methods are referred to as *MI_Powell*, *CC_Powell,* and *GC_Powell*, indicating the adopted similarity measure and optimization method. We also implemented another intensity-based method combining MI and GC to achieve improved robustness and accuracy for comparison. MI focuses on the match of the histograms at the global scale, which leads to a relatively large capture range, but lacks fine accuracy. GC focuses on matching image gradients, which leads to high registration accuracy, but limits the capture range. The combined method, referred to as *MI_GC_Powell*, first applies MI_Powell to bring the registration into the capture range of GC, and then applies GC_Powell to refine the registration.

We also evaluated CLARET, a linear regression-based 2-D/3-D registration method introduced in [14], which is closely related to PEHL, as it iteratively applies regressors on the image residual to estimate the transformation parameters. In [14], the linear regressors were reported to be trained on X-ray images with fixed ground truth transformation parameters, and therefore can only be applied on X-ray images with poses within a limited range. Since the input X-ray images used in our experiment do not have such limitation, we applied the PSP strategy to train linear regressors separately for each zone. For each zone, the linear regressor was trained on the dataset used for training PEHL for Group 1.

12.7.4.2 *Results*

We first observed that the linear regressor in CLARET completely failed in our experiment setup. Table 12.2 shows the root mean squared error (RMSE) of the 6 parameters yielded by PEHL and CLARET on the synthetic training data for XEF.

Table 12.2 RMSE of the 6 transformation parameters yielded by PEHL and CLARET on the training data for XEF

	t_x (mm)	t_y (mm)	t_z (mm)	t_θ (°)	t_α (°)	t_β (°)
Start	0.86	0.86	8.65	1.71	8.66	8.66
PEHL	0.04	0.04	0.32	0.06	0.18	0.18
CLARET	0.51	0.88	34.85	2.00	19.41	17.52

The linear regression resulted in very large errors on the training data (i.e., larger than the perturbation), indicating that the mapping from the global image residual to the underlying transformation parameters is highly nonlinear, and therefore cannot be reliably captured by a linear regressor. In comparison, PEHL employs the 3 algorithmic strategies to simplify the nonlinear relationship and captures it using a CNN with a strong nonlinear modeling capability. As a result, PEHL resulted in a very small error on the synthetic training data. Its ability to generalize the performance to unseen testing data was then assessed on real data from the three clinical applications.

Table 12.3 summarizes the success rate, capture range, percentiles of mTREproj and average running time per registration for PEHL and four intensity-based methods on the three applications. The results show that the four intensity-based methods, MI_Powell, GC_Powell, CC_Powell, and MI_GC_Powell, all resulted in relatively small capture ranges and slow speeds that are incapable for real-time registration. The small capture range is due to the limitation of non-convex optimization. Because the intensity-based similarity measures are highly non-convex, the optimizer is likely to get trapped in local maxima if the starting position is not close enough to the global maxima. The relatively slow speed is due to the large number of DRR renderings and similarity measure calculations during the optimization. The fastest intensity-based method is CC_Powell, which took 0.4–0.9 s per registration, is still significantly slower than typical fluoroscopic frame rates (i.e., 10–15 fps). The success rates for MI_Powell, GC_Powell, and CC_Powell are also very low, mainly due to two different reasons: (i) MI and CC are unable to resolve a small mismatch; (ii) GC is unable to recover a large mismatch. By employing MI_Powell to recover a large mismatch and GC_Powell to resolve a small mismatch, MI_GC_Powell achieved much higher success rates than using MI or GC alone.

The results show that PEHL achieved the best success rate and capture range among all evaluated methods on all three applications, and is capable for real-time registration. The advantage of PEHL in capture range compared to the second best-performed method, i.e., MI_GC_Powell, is significant. In particular, on the three applications, PEHL resulted in 155% (on TKA), 99% (on VIPS), and 306% (on XEF) larger capture range than MI_GC_Powell, respectively. The success rates of PEHL are also higher than that of MI_GC_Powell by 27.8% (on TKA), 5% (on VIPS), and 5.4% (on XEF). The advantage of PEHL in capture range and robustness is primarily due to the learning of the direct mapping from the LIR to the residual of the transformation parameters, which eliminates the need of optimizing over a highly non-convex similarity measure. PEHL resulted in a running time of ~0.1 s per registration for all

Table 12.3 Quantitative experiment results of PEHL and baseline methods. Success rate is the percentage of successful registrations in each experiment. Capture range is the initial mTREproj for which 95% of the registrations were successful. The 10th, 25th, 50th, 75th, and 90th percentiles of mTREproj are reported. Running time records the average and standard deviation of the computation time for each registration computed in each experiment. Capture range is only reported for experiments where the there are more than 20 samples within the capture range

Application	Method	SR	CR (mm)	mTREproj percentile (mm)					RT (s)
				10th	25th	50th	75th	90th	
TKA	Start	N/A	N/A	3.29	4.63	6.98	10.1	12.7	N/A
	MI_Powell	36.2%	N/A	0.44	0.75	1.69	6.24	8.42	1.37 ± 0.44
	CC_Powell	43.8%	1.88	0.35	0.64	1.36	6.32	8.40	0.92 ± 0.27
	GC_Powell	45.2%	2.14	0.33	0.59	1.31	7.64	9.77	2.52 ± 1.22
	MI_GC_Powell	51.8%	2.83	**0.30**	0.52	1.05	6.41	8.61	3.11 ± 0.94
	PEHL	**79.6%**	**7.23**	0.33	**0.44**	**0.59**	**0.90**	**6.73**	**0.11 ± 0.00**
VIPS	Start	N/A	N/A	1.18	1.52	2.00	2.59	3.10	N/A
	MI_Powell	75.1%	N/A	0.16	0.23	0.38	0.60	0.92	1.66 ± 0.60
	CC_Powell	57.7%	0.89	0.19	0.30	0.54	0.85	1.29	0.91 ± 0.31
	GC_Powell	78.7%	1.12	0.12	0.21	0.33	0.54	2.28	3.91 ± 1.55
	MI_GC_Powell	92.7%	2.77	**0.11**	**0.17**	0.26	**0.37**	0.54	4.71 ± 1.59
	PEHL	**99.7%**	**5.51**	0.15	0.18	**0.24**	0.39	**0.45**	**0.10 ± 0.00**
XEF	Start	N/A	N/A	1.05	1.37	1.83	2.31	2.79	N/A
	MI_Powell	69.7%	N/A	0.17	0.21	0.28	0.40	0.60	0.79 ± 0.29
	CC_Powell	54.8%	N/A	0.12	0.17	0.32	0.89	1.17	0.40 ± 0.10
	GC_Powell	56.9%	N/A	0.07	0.14	0.28	1.06	3.15	2.06 ± 1.05
	MI_GC_Powell	89.1%	0.84	**0.05**	**0.10**	0.17	0.27	0.38	2.03 ± 0.69
	PEHL	**94.5%**	**3.33**	0.08	0.11	**0.15**	**0.20**	**0.24**	**0.10 ± 0.00**

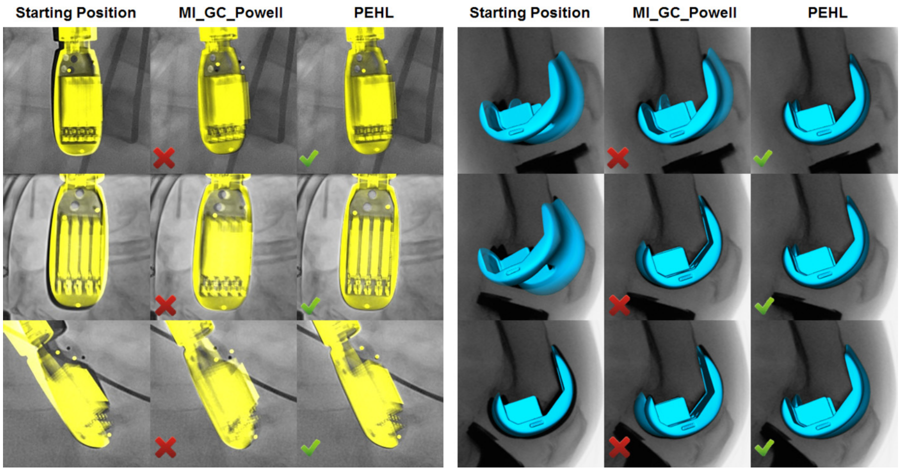


FIGURE 12.14

Visual examples of registration results using MI_GC_Powell and PEHL.

three applications, which is 20–45 times faster than that of MI_GC_Powell and leads to real-time registration at ∼10 fps. In addition, because the computation involved in PEHL is fixed for each registration, the standard deviation of the running time of PEHL is almost zero, so that PEHL can provide real-time registration at a stable frame rate. In comparison, intensity-based methods require different numbers of iterations for each registration, depending on the starting position, which leads to a relatively large standard deviation of the running time. The mTREproj percentiles show that at lower percentiles (e.g., 10th and 25th), the mTREproj of PEHL is in general larger than that of MI_GC_Powell. This is partially due to the fact that the ground truth parameters were generated using GC, which could bear a slight bias toward intensity-based methods using GC as the similarity measure. For higher percentiles (e.g., 75th and 90th), the mTREproj of PHEL becomes smaller than that of MI_GC_Powell, showing that PHEL is more robust than MI_GC_Powell. Some visual examples of registration results using MI_GC_Powell and PEHL are shown in Fig. 12.14.

12.8 DISCUSSION

In this chapter, we presented a CNN regression-based method, PEHL, for real-time 2-D/3-D registration. To successfully solve 2-D/3-D registration problems using regression, we introduced 3 novel algorithmic strategies, LIR, MSR, and PSP, to simplify the underlying mapping to be regressed, and designed a CNN regression model with strong nonlinear modeling capability to capture the mapping. We furthermore validated that all 3 algorithmic strategies and the CNN model are important to the success of PEHL, by disabling them from PEHL and showing the detrimental effect on performance. We empirically found that applying PEHL for 3 iterations is the optimal setting, which leads to close to the optimal success rate and a real-time registration speed of ∼10 fps. We also demonstrated that PEHL has a strong ability to reproduce the same registration result from different initial positions, by showing that the RMSDproj of registered targets approaches to almost zero (i.e., 0.005 mm) as the number of iterations of PEHL increases to 6. In comparison, the RMSEproj using the best performed intensity-based method, MI_GC_Powell, is 0.52 mm. On three potential clinical applications, we compared PEHL with 4 intensity-based 2-D/3-D registration methods and a linear regression-based method, and showed that PEHL achieved much higher robustness and larger capture range. In particular, PEHL increased the capture range by 99–306% and the success rate by 5–27.8%, compared to MI_GC_Powell. We also showed that PEHL achieved significantly higher computational efficiency than intensity-based methods, and is capable of real-time registration.

The significant advantage of PEHL in robustness and computational efficiency over intensity-based methods is mainly due to the fact that CNN regressors are trained to capture the mapping from LIRs to the underlying transformation parameters. In every iteration, PEHL fully exploits the rich information embedded in LIR to make an informed estimation of the transformation parameters, and therefore it is able to achieve highly robust and accurate registration with only a minimum number of iterations. In comparison, intensity-based methods always map the DRR and X-ray images to a scalar-valued merit function, where the information about the transformation parameters embedded in the image intensities is largely lost. The registration problem is then solved by heuristically optimizing this scalar-valued merit function, which leads to an inefficient iterative computation and a high chance of getting trapped into local maxima.

The results also show that PEHL is more accurate and robust than two accelerated intensity-based 2-D/3-D registration methods, Sparse Histogramming MI (SHMI) [6] and Direct Splatting Correlation (DSC) [9], which employ sub-sampled DRR and splatting DRR to quickly compute approximated MI and CC, respectively. Because of the approximation, SHMI and DSC theoretically achieve the same or degraded accuracy compared to using original MI and CC. As shown in Table 12.3, all reported mTREproj percentiles of PEHL are lower than that of MI_Powell and CC_Powell, and the differences at mid-range percentiles (i.e., 25th, 50th, and 75th) are quite significant. In particular, at the 50th percentile, the mTREproj of PEHL are 25–65% lower than that of MI_Powell and CC_Powell on all three applications. These results suggest that PEHL significantly outperforms SHMI and DSC in terms of robustness and accuracy. In terms of computational efficiency, while all three methods are capable of real-time registration, with an efficient GPU implementation, DSC reported the highest registration speed (i.e., 23.6–92.3 fps) [9].

Like for any machine learning-based method, an important factor for the success of PEHL is the quantity and quality of the training data. For PEHL, it has been a challenge to obtain sufficient amount of annotated real X-ray images for training, because accurate annotation of 3-D transformation on X-ray projection image is very difficult, especially for those out-of-plane parameters. We have shown that by generating well-simulated synthetic data and training the CNN network on synthetic data only, we could achieve high performance when applying PEHL on real X-ray images. However, it is worth noting that if the object to be registered is a device or implant that is manufactured with a fixed design, it is also possible to have a factory setup to massively acquire real X-ray images with a known ground truth for training PEHL.

DISCLAIMER

This feature is based on research, and is not commercially available. Due to regulatory reasons its future availability cannot be guaranteed.

REFERENCES

1. Rui Liao, Li Zhang, Ying Sun, Shun Miao, Christophe Chefd'Hotel, A review of recent advances in registration techniques applied to minimally invasive therapy, IEEE Trans. Multimed. 15 (5) (2013) 983–1000.
2. Primoz Markelj, D. Tomaževič, Bostjan Likar, F. Pernuš, A review of 3d/2d registration methods for image-guided interventions, Med. Image Anal. 16 (3) (2012) 642–661.
3. Christelle Gendrin, Hugo Furtado, Christoph Weber, Christoph Bloch, Michael Figl, Supriyanto Ardjo Pawiro, Helmar Bergmann, Markus Stock, Gabor Fichtinger, Dietmar Georg, et al., Monitoring tumor motion by real time 2d/3d registration during radiotherapy, Radiother. Oncol. 102 (2) (2012) 274–280.
4. Jérôme Schmid, Christophe Chênes, Segmentation of X-ray images by 3d–2d registration based on multibody physics, in: Computer Vision – ACCV 2014, Springer, 2014, pp. 674–687.
5. Shun Miao, Rui Liao, Yefeng Zheng, A hybrid method for 2-d/3-d registration between 3-d volumes and 2-d angiography for trans-catheter aortic valve implantation (TAVI), in: 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, IEEE, 2011, pp. 1215–1218.
6. Lilla Zöllei, E. Grimson, Alexander Norbash, W. Wells, 2d–3d rigid registration of X-ray fluoroscopy and CT images using mutual information and sparsely sampled histogram estimators, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, CVPR 2001, IEEE, 2001, pp. II-609–II-703.
7. Wolfgang Birkfellner, Markus Stock, Michael Figl, Christelle Gendrin, Johann Hummel, Shuo Dong, Joachim Kettenbach, Dietmar Georg, Helmar Bergmann, Stochastic rank correlation: a robust merit function for 2d/3d registration of image data obtained at different energies, Med. Phys. 36 (8) (2009) 3420–3428.
8. Lee Westover, Footprint evaluation for volume rendering, in: ACM SIGGRAPH Computer Graphics, vol. 24, ACM, 1990, pp. 367–376.

9. Charles Hatt, Michael Speidel, Amish Raval, Robust 5DOF transesophageal echo probe tracking at fluoroscopic frame rates, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2002, Springer, 2015.
10. Jens Kruger, Rüdiger Westermann, Acceleration techniques for GPU-based volume rendering, in: Proceedings of the 14th IEEE Visualization 2003, VIS'03, IEEE Computer Society, 2003, p. 38.
11. Michael M. Bronstein, Alexander M. Bronstein, Fabrice Michel, Nikos Paragios, Data fusion through cross-modality metric learning using similarity-sensitive hashing, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2010, pp. 3594–3601.
12. Fabrice Michel, Michael Bronstein, Alex Bronstein, Nikos Paragios, Boosted metric learning for 3d multi-modal deformable registration, in: 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, IEEE, 2011, pp. 1209–1214.
13. Ana Rodrigues Gouveia, Coert Metz, Luis Freire, Pedro Almeida, Stefan Klein, Registration-by-regression of coronary CTA and X-ray angiography, Comput. Methods Biomech. Biomed. Eng. (2015) 1–13.
14. Chen-Rui Chou, Brandon Frederick, Gig Mageras, Sha Chang, Stephen Pizer, 2d/3d image registration using regression learning, Comput. Vis. Image Underst. 117 (9) (2013) 1095–1106.
15. Paul Wohlhart, Vincent Lepetit, Learning descriptors for object recognition and 3d pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3109–3118.
16. Piotr Dollár, Peter Welinder, Pietro Perona, Cascaded pose regression, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2010, pp. 1078–1085.
17. Christopher Zach, Adrian Penate-Sanchez, Minh-Tri Pham, A dynamic programming approach for fast and robust object pose recognition from range images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 196–203.
18. Roozbeh Mottaghi, Yu Xiang, Silvio Savarese, A coarse-to-fine model for 3d pose estimation and sub-category recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 418–426.
19. Markus Kaiser, Matthias John, Tobias Heimann, Alexander Brost, Thomas Neumuth, Georg Rose, 2d/3d registration of tee probe from two non-orthogonal C-arm directions, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014, Springer, 2014, pp. 283–290.
20. Dejan Tomaževič, Bostjan Likar, Tomaž Slivnik, Franjo Pernuš, 3-d/2-d registration of CT and MR to X-ray images, IEEE Trans. Med. Imaging 22 (11) (2003) 1407–1416.
21. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
22. Xavier Glorot, Yoshua Bengio, Understanding the difficulty of training deep feedforward neural networks, in: International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.
23. Z. Zhu, S. Ji, M. Yang, H. Ding, G. Wang, An application of the automatic 2d–3d image matching technique to study the in-vivo knee joint kinematics before and after TKA, in: World Congress on Medical Physics and Biomedical Engineering, May 26–31, 2012, Beijing, China, Springer, 2013, pp. 230–233.

24. S.Y. Vetter, I. Mühlhäuser, J. von Recum, P.-A. Grützner, J. Franke, Validation of a virtual implant planning system (VIPS) in distal radius fractures, Bone Joint J. Orthop. Proc. Suppl. 96 (SUPP 16) (2014) 50.
25. Gang Gao, Graeme Penney, Yingliang Ma, Nicolas Gogin, Pascal Cathier, Aruna Arujuna, Geraint Morton, Dennis Caulfield, Jaswinder Gill, C. Aldo Rinaldi, et al., Registration of 3d trans-esophageal echocardiography to X-ray fluoroscopy using image-based probe tracking, Med. Image Anal. 16 (1) (2012) 38–49.
26. Everine B. De Kraats, Graeme P. Penney, Dejan Tomaževič, Theo Van Walsum, Wiro J. Niessen, Standardized evaluation methodology for 2-d–3-d registration, IEEE Trans. Med. Imaging 24 (9) (2005) 1177–1189.
27. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell, Caffe: convolutional architecture for fast feature embedding, arXiv:1408.5093, 2014.
28. Markus Kaiser, Matthias John, Tobia Heimann, Thomas Neumuth, Georg Rose, Comparison of optimizers for 2d/3d registration for fusion of ultrasound and X-ray, in: Bildverarbeitung für die Medizin 2014: Algorithmen–Systeme–Anwendungen, Proceedings des Workshops vom 16. bis 18. März 2014 in Aachen, Springer, 2014, p. 312.