A CNN Regression Approach for Real-Time 2D/3D Registration

Shun Miao*, *Member, IEEE*, Z. Jane Wang, *Senior Member, IEEE*, and Rui Liao, *Senior Member, IEEE*

***Abstract*—In this paper, we present a Convolutional Neural Network (CNN) regression approach to address the two major limitations of existing intensity-based 2-D/3-D registration technology: 1) slow computation and 2) small capture range. Different from optimization-based methods, which iteratively optimize the transformation parameters over a scalar-valued metric function representing the quality of the registration, the proposed method exploits the information embedded in the appearances of the digitally reconstructed radiograph and X-ray images, and employs CNN regressors to directly estimate the transformation parameters. An automatic feature extraction step is introduced to calculate 3-D pose-indexed features that are sensitive to the variables to be regressed while robust to other factors. The CNN regressors are then trained for local zones and applied in a hierarchical manner to break down the complex regression task into multiple simpler sub-tasks that can be learned separately. Weight sharing is furthermore employed in the CNN regression model to reduce the memory footprint. The proposed approach has been quantitatively evaluated on 3 potential clinical applications, demonstrating its significant advantage in providing highly accurate real-time 2-D/3-D registration with a significantly enlarged capture range when compared to intensity-based methods.**

***Index Terms*—2-D/3-D registration, convolutional neural network, deep learning, image guided intervention.**

I. Introduction

TWO-DIMENSIONAL to three-dimensional 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.

In existing methods, accurate 2-D/3-D registration is typically achieved by intensity-based 2-D/3-D registration methods [2]–[5]. In these methods, 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. An optimizer is employed to maximize an intensity-based similarity measure between the DRR and X-ray images. Intensity-based methods are known to be able to achieve high registration accuracy [6], but at the same time, they suffer from two major drawbacks: 1) long computation time and 2) small capture range. Specifically, because intensity-based methods involve a large number of evaluations of the similarity measure, each requiring heavy computation in rendering the DRR, they typically resulted in above 1 s running time, and therefore are not suitable for real-time applications. In addition, because the similarity measures to be optimized in intensity-based methods are often highly non-convex, the optimizer has a high chance of getting trapped into local maxima, which leads to a small capture range of these methods.

The small capture range of intensity-based methods is often addressed by employing initialization methods before registration [7], [8]. However, initialization methods typically utilize dominant features of the target object for pose recovery and therefore are very application specific. For example, Varnavas *et al.* [7] applied Generalized Hough Transform (GHT) for initial pose estimation of spine vertebrae. This method is specific to applications where spine vertebrae edges are clearly visible in both the X-ray and CT images. Miao *et al.* [8] proposed to use shape encoding combined with template matching for initial pose estimation. This method can only be applied on metal implants, which are highly X-ray opaque objects that can be reliably segmented from X-ray images for shape encoding.

Some efforts have been made toward accelerating DRR generation for fast 2D/3D registration. 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 [9], [10]. However, only a few similarity measures are suitable to be calculated on a random subset of the image, e.g., Mutual Information (MI) [9] and Stochastic Rank Correlation (SRC) [10]. Another strategy is splatting, which is a voxel-based volume rendering technique that directly projects single voxels to the imaging plane [11], [12]. Splatting allows to only use voxels with intensity above certain threshold for rendering, which significantly reduces the number of voxels to be

Manuscript received December 30, 2015; accepted January 15, 2016. Date of publication January 26, 2016; date of current version April 29, 2016. *Asterisk indicates corresponding author.*

*S. Miao is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, V6T 1Z4 Canada, and also with Medical Imaging Technologies, Siemens Healthcare, Princeton, NJ 08540 USA (e-mail: smiao@ece.ubc.ca).

Z. J. Wang is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, V6T 1Z4 Canada (e-mail: zjwang@ece.ubc.ca).

R. Liao is with Medical Imaging Technologies, Siemens Healthcare, Princeton, NJ 08540 USA (e-mail: rui.liao@siemens.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2016.2521800

visited. However, one inherent problem of splatting is that due to aliasing artifacts, the image quality of the generated DRR is significantly degraded, which subsequently degrades the registration accuracy compared to the DRR generated by the standard Ray Casting algorithm [13].

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 [14], [15]. 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 [16], [17]. Gouveia *et al.* [16] 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 that 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.* [17] 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 non-linear, 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 [18]–[21]. 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 target at a very high accuracy (i.e., sub-millimeter) for diagnosis and surgery guidance purposes. For example, Wohlhart *et al.* [18] proposed to train a Convolutional Neural Networks (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 accuracy is relatively low, i.e., the success rate for angle error less than 5 degrees is below 60% for $k = 1$. Dollár *et al.* [19] 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 solves 2-D pose estimation from RGB images with hundreds of iterations, and therefore are not applicable for 2-D/3-D registration problems with a real-time (e.g., a few updates) requirement.

In this paper, a CNN regression approach, referred to as Pose Estimation via Hierarchical Learning (PEHL), is proposed to achieve real-time 2-D/3-D registration with a large capture range and high accuracy. The key of our approach is to train CNN regressors to recover the mapping from the DRR and X-ray images to the difference of their underlying transformation parameters. Such 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 non-linear relationship using the following three algorithmic strategies and then capturing the mapping using CNN regressors with a strong non-linear modeling capability.

- *Local image residual (LIR)*: To simplify the underlying mapping to be captured by the regressors, we introduce an LIR feature for regression, which is approximately 3-D pose-indexed, i.e., it is only affected by the difference between the initial and ground truth transformation parameters.
- *Parameter space partitioning (PSP)*: We partition the transformation parameter space into zones and train CNN regressors in each zone separately, to break down the complex regression task into multiple simpler sub-tasks.
- *Hierarchical parameter regression (HPR)*: We decompose the transformation parameters and regress them in a hierarchical manner, to achieve highly accurate parameter estimation in each step.

The remainder of this paper is organized as follows. Section II provides backgrounds of 2-D/3-D registration, and formulates it as a regression problem. Section III presents the proposed PEHL approach. Validation datasets, evaluation metrics and experiment configurations are described in Section IV, and experimental results are shown in Section V. Section VI concludes the paper with a discussion of our findings.

II. Problem Formulation

A. X-Ray Imaging Model

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:

$$I(\boldsymbol{p}) = \int \mu\Big(\boldsymbol{L}(\boldsymbol{p}, r)\Big) dr, \tag{1}$$

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 $J : \mathbb{R}^3 \rightarrow \mathbb{R}$, and the 3-D transformation from the object coordinate system to the X-ray imaging coordinate system as $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, the attenuation coefficient at point $\boldsymbol{x}$ in the X-ray imaging coordinate system is

$$\mu(\boldsymbol{x}) = J(T^{-1} \circ \boldsymbol{x}). \tag{2}$$

Combining (1) and (2), we have

$$I(\boldsymbol{p}) = \int J\Big(T^{-1} \circ \boldsymbol{L}(\boldsymbol{p}, r)\Big) dr. \tag{3}$$

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. Note that given J, $\boldsymbol{L}$ and T, a synthetic X-ray image $I(\cdot)$ can be computed following (3) using Ray-Casting algorithm [13], and the generated image is referred to as DRR.

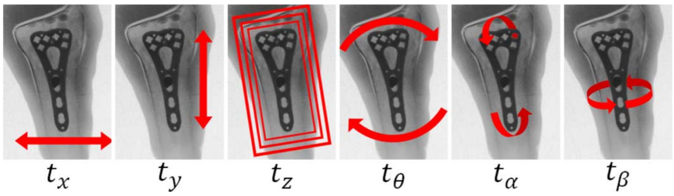

Fig. 1. Effects of the 6 transformation parameters.

B. 3-D Transformation Parameterization

A rigid-body 3-D transformation T can be parameterized by a vector $\boldsymbol{t}$ with 6 components. In our approach, we parameterize the transformation by 3 in-plane and 3 out-of-plane transformation parameters [22], as shown in Fig. 1. 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.

C. 2-D/3-D Registration via Regression

Based on (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: 1) a 3-D object described by its X-ray attenuation map J, 2) an X-ray image $I_{\boldsymbol{t}_{gt}}$, where $\boldsymbol{t}_{gt}$ denotes the unknown ground truth transformation parameters, and 3) initial transformation parameters $\boldsymbol{t}_{ini}$. The 2-D/3-D registration problem can be 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 $\boldsymbol{\epsilon}$:

$$\boldsymbol{t}_{gt} - \boldsymbol{t}_{ini} \approx \boldsymbol{f}\big(X(\boldsymbol{t}_{ini}, I_{\boldsymbol{t}_{gt}})\big), \quad \forall \boldsymbol{t}_{gt} - \boldsymbol{t}_{ini} \in \boldsymbol{\epsilon}. \tag{4}$$

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

$$\hat{\boldsymbol{t}}_{gt} = \boldsymbol{t}_{ini} + \boldsymbol{f}\big(X(\boldsymbol{t}_{ini}, I_{\boldsymbol{t}_{gt}})\big). \tag{5}$$

It is worth noting that the range $\boldsymbol{\epsilon}$ in (4) is equivalent to the capture range of optimization-based registration methods. Based on (4), our problem formulation can be expressed as designing a feature extractor $X(\cdot)$ and training regressors $\boldsymbol{f}(\cdot)$, such that

$$\delta\boldsymbol{t} \approx \boldsymbol{f}\big(X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}})\big), \quad \forall \delta\boldsymbol{t} \in \boldsymbol{\epsilon}. \tag{6}$$

In the next section, we will discuss in detail 1) how the feature $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}})$ is calculated and 2) how the regressors $\boldsymbol{f}(\cdot)$ are designed, trained and applied.

III. Pose Estimation via Hierarchical Learning

A. Parameter Space Partitioning

We aim at training regressors to recover the mapping from the feature $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}})$ to the parameter residuals $\delta\boldsymbol{t}$. Since the feature naturally depends on $\boldsymbol{t}$, the target mapping could vary significantly as $\boldsymbol{t}$ changes, which makes it highly complex and difficult to be accurately recovered. Ideally, we would like to extract a feature that is sensitive to the parameter residuals $\delta\boldsymbol{t}$, and is insensitive to the parameters $\boldsymbol{t}$. Such feature is referred to as pose-index feature, and the property can be expressed as:

$$X(\boldsymbol{t}_1, I_{\boldsymbol{t}_1+\delta\boldsymbol{t}}) \approx X(\boldsymbol{t}_2, I_{\boldsymbol{t}_2+\delta\boldsymbol{t}}) \quad \forall(\boldsymbol{t}_1, \boldsymbol{t}_2). \tag{7}$$

As we will show in Section III-B, we use ROIs to make $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}})$ invariant to the in-plane and scaling parameters, $(t_x, t_y, t_z, t_\theta)$. However, we are unable to make $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}})$ insensitive to t_α and t_β, because they cause complex appearance changes in the projection image. To solve this problem, we partition the parameter space spanned by t_α and t_β into a 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*. We will show in Section III-B that for (t_α, t_β) within each zone, our LIR feature is approximately pose-indexed, i.e.,

$$X_k(\boldsymbol{t}_1, I_{\boldsymbol{t}_1+\delta\boldsymbol{t}}) \approx X_k(\boldsymbol{t}_2, I_{\boldsymbol{t}_2+\delta\boldsymbol{t}}) \quad \forall(\boldsymbol{t}_1, \boldsymbol{t}_2) \in \Omega_k, \tag{8}$$

where $X_k(\cdot,\cdot)$ denotes the LIR feature extractor for the k-th zone, and Ω_k denotes the area covered by the k-th zone. The regressors therefore are trained separately for each zone to recover the simplified mapping that is insensitive to $\boldsymbol{t}$.

B. Local Image Residual

1) Calculation of Local Image Residual: The LIR feature is calculated as the difference between the 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. To determine the locations, sizes and orientations of the local patches, a number of 3-D points are extracted from the 3-D model of the target object following the steps described in Section III-B2. Given a 3-D point $\boldsymbol{p}$ and parameters $\boldsymbol{t}$, a square local ROI is uniquely determined in the 2-D 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 $H_{\boldsymbol{p}}^{\boldsymbol{t}}(\cdot)$ 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

$$X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}}, \boldsymbol{P}) = \big\{H_{\boldsymbol{p}_i}^{\boldsymbol{t}}(I_{\boldsymbol{t}}) - H_{\boldsymbol{p}_i}^{\boldsymbol{t}}(I_{\boldsymbol{t}+\delta\boldsymbol{t}})\big\}_{i=1,\ldots,N}. \tag{9}$$

In a local area of $I_{\boldsymbol{t}}$, the effect of varying t_α and t_β within a zone is approximately 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 $H_{\boldsymbol{p}}^{\boldsymbol{t}}(I_{\boldsymbol{t}})$ approximately invariant to $\boldsymbol{t}$. Since the difference between $H_{\boldsymbol{p}}^{\boldsymbol{t}}(I_{\boldsymbol{t}+\delta\boldsymbol{t}})$ and $H_{\boldsymbol{p}}^{\boldsymbol{t}}(I_{\boldsymbol{t}})$ is merely additional 2-D transformation caused by $\delta\boldsymbol{t}$, $H_{\boldsymbol{p}}^{\boldsymbol{t}}(I_{\boldsymbol{t}+\delta\boldsymbol{t}})$ is also approximately invariant to $\boldsymbol{t}$. The workflow of LIR feature extraction is shown in Fig. 2.

2) Extraction of 3-D Points: The 3-D points used for calculating the LIR feature are extracted separately for each zone in

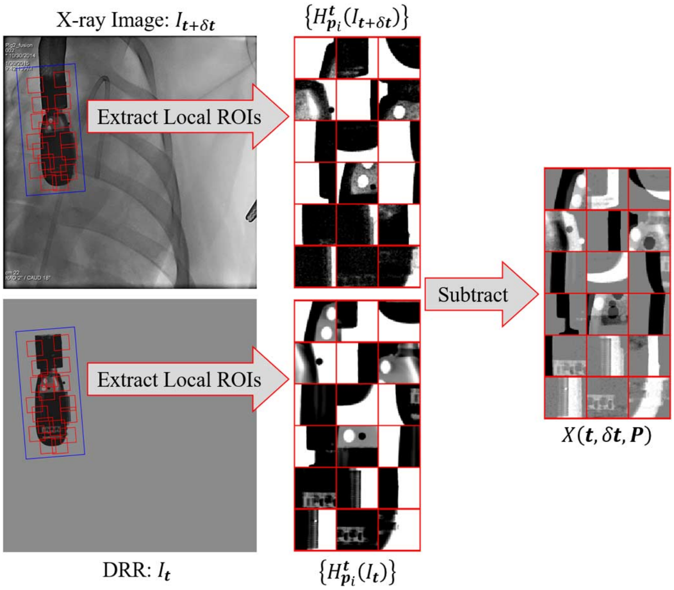
X-ray Image: $I_{t+\delta t}$
$\{H^{\boldsymbol{t}}_{\boldsymbol{p}_i}(I_{\boldsymbol{t}+\delta\boldsymbol{t}})\}$
Extract Local ROIs
Subtract
Extract Local ROIs
$X(\boldsymbol{t}, \delta\boldsymbol{t}, \boldsymbol{P})$
DRR: $I_{\boldsymbol{t}}$
$\{H^{\boldsymbol{t}}_{\boldsymbol{p}_i}(I_{\boldsymbol{t}})\}$

Fig. 2. 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 IV-D.

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 [23] as:

$$g(\boldsymbol{p}) = \int \eta\Big(\boldsymbol{L}(\boldsymbol{p}, r)\Big)\, dr, \tag{10}$$

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 [23]. We back-project $\boldsymbol{p}$ to $\boldsymbol{L}(\boldsymbol{p}, r_0)$, where

$$r_0 = \arg\max_r \boldsymbol{L}(\boldsymbol{p}, r), \tag{11}$$

if

$$\int_{r_0-\sigma}^{r_0+\sigma} \eta\Big(\boldsymbol{L}(\boldsymbol{p}, r)\Big)\, dr \geq 0.9 \cdot g(\boldsymbol{p}). \tag{12}$$

The condition in (12) 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., σ = 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 the ones leading to LIR satisfying (7) and also not significantly overlapped 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 n-th pixel of $H^{\boldsymbol{t}_j}_{\boldsymbol{p}_i}(I_{\boldsymbol{t}_j}) - H^{\boldsymbol{t}_j}_{\boldsymbol{p}_i}(I_{\boldsymbol{t}_j+\delta\boldsymbol{t}_k})$ is denoted as $h_{n,i,j,k}$. The following two measurements are computed for all candidates:

$$E_i = \left\langle \left(h_{n,i,j,k} - \langle h_{n,i,j,k}\rangle_j\right)^2 \right\rangle_{n,j,k}, \tag{13}$$

$$F_i = \left\langle \left(h_{n,i,j,k} - \langle h_{n,i,j,k}\rangle_k\right)^2 \right\rangle_{n,j,k}, \tag{14}$$

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 $H^{\boldsymbol{t}}_{\boldsymbol{p}_i}(I_{\boldsymbol{t}}) - H^{\boldsymbol{t}}_{\boldsymbol{p}_i}(I_{\boldsymbol{t}+\delta\boldsymbol{t}})$ with respect to $\boldsymbol{t}$ and $\delta\boldsymbol{t}$, respectively, an ideal LIR should have a small E_i to satisfy (7) 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.

C. Hierarchical Parameter Regression

Algorithm 1 PEHL: Application Stage

```
procedure REGISTER $(\boldsymbol{t}, I, k)$
    repeat
        Retrieve $\boldsymbol{P}$ for the zone covering $(t_\alpha, t_\beta)$
        Retrieve $\boldsymbol{f}(\cdot)$ for the zone covering $(t_\alpha, t_\beta)$
        Calculate $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}}, \boldsymbol{P})$    ▷ (9)
        $\boldsymbol{t}_{\{x,y,\theta\}} \leftarrow \boldsymbol{t}_{\{x,y,\theta\}} + \boldsymbol{f}_{\{x,y,\theta\}}(X)$
        Calculate $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}}, \boldsymbol{P})$    ▷ (9)
        $\boldsymbol{t}_{\{\alpha,\beta\}} \leftarrow \boldsymbol{t}_{\{\alpha,\beta\}} + \boldsymbol{f}_{\{\alpha,\beta\}}(X)$
        Calculate $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}}, \boldsymbol{P})$    ▷ (9)
        $t_z \leftarrow t_z + f_z(X)$
    until reaching $k$ iterations
    return $\boldsymbol{t}$
```

Instead of regressing the 6 parameters together, which makes the mapping to be regressed more complex as multiple confounding factors are involved, we divide them into the following 3 groups, and regress them hierarchically:

- *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 are less affected by the variations of the parameters in the other two groups. The parameter in Group 3 is the most difficult one 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 one. 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 te

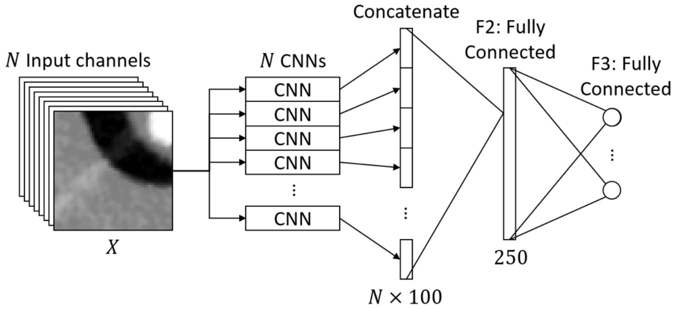

Fig. 3. Structure of the CNN regression model.

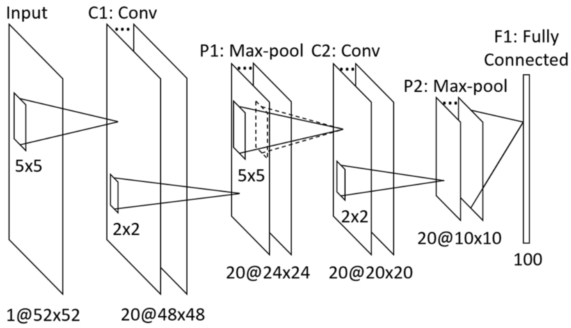

Fig. 4. Structure of the CNN applied for each input channel.

TABLE I
DISTRIBUTIONS OF RANDOMLY GENERATED $\delta\boldsymbol{t}$. $\mathcal{U}(a,b)$ DENOTES THE UNIFORM DISTRIBUTION BETWEEN a AND b. THE UNITS FOR TRANSLATIONS AND ROTATIONS ARE mm AND DEGREE, RESPECTIVELY

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

regressed for each group is simplified by limiting the dimension and removing the compounding factors coming from those parameters in the previous groups.

The above HPR process can be repeated for a few iterations to achieve the optimal accuracy, and the result of the current iteration is used as the starting position for the next iteration (Algorithm 1). The number of iterations can be empirically selected (e.g., 3 times in our application), as will be described in Section V-A.

D. CNN Regression Model

In the proposed regression approach, challenges in designing the CNN regression model are two folds: 1) 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 2) 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. We employ the following CNN regression model to address these 2 challenges.

1) Network Structure: A CNN [24] regression model with the architecture shown in Fig. 3 is trained for each group in each zone. According to (9), the input of the regression model consists of N channels, corresponding to N LIRs. The CNN shown in Fig. 4 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 the N input channels have the same nature, i.e., they are LIRs at different locations, the weights in the N CNNs are shared to reduce the memory footprint by N times.

In our experiment, we empirically selected the size of the ROI, which led to $N \approx 18$. Using the CNN model shown in Fig. 3 with weight sharing, there are in total 660,500 weights 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. 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 in total 972 CNN regression models and pre-loading all of them into RAM requires 2.39 GB, which is manageable for modern computers.

2) Training: The CNN regression models are trained exclusively on synthetic X-ray images, because they provide reliable ground truth labels with little needs on laborious manual annotation, and the quantity of real X-ray images could be limited. 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 I. 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 close to zero after the regressors in the first group are applied. 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 LIR feature $X(\boldsymbol{t}, I_{\boldsymbol{t}+\delta\boldsymbol{t}})$ is calculated following (9).

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

$$\Phi = \frac{1}{K}\sum_{i=1}^{K} \|\boldsymbol{y}_i - \boldsymbol{f}(X_i; \mathbf{W})\|_2^2, \tag{15}$$

where K is the number of training samples, $\boldsymbol{y}_i$ is the label for the i-th 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 i-th training sample. The weights $\mathbf{W}$ are learned using Stochastic Gradient Descent (SGD) [24], 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:

$$\mathbf{V}_{i+1} := m \cdot \mathbf{V}_i - d \cdot \kappa_i \cdot \mathbf{W}_i - \kappa_i \cdot \left\langle \frac{\partial \Phi}{\partial \mathbf{W}}\bigg|_{\mathbf{W}_i} \right\rangle_{D_i}, \tag{16}$$

$$\mathbf{W}_{i+1} := \mathbf{W}_i + \mathbf{V}_{i+1}, \tag{17}$$

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

$$\kappa_i = 0.0025 \cdot (1 + 0.0001 \cdot i)^{-0.75}. \tag{18}$$

The derivative $\partial\Phi/\partial\mathbf{W}$ is calculated using back-propagation. For weights shared in multiple paths, their derivatives in all paths are back-propagated separately and summed up for weight update. The weights are initialized using the Xavier method [25], and mini-batch SGD is performed for 12,500 iterations (32 epochs).

IV. Experiments

A. Datasets

We evaluated PEHL on datasets from the following 3 clinical applications to demonstrate its wide applicability for real-time 2-D/3-D registration:

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 [26]. 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 [27]. In VIPS, 2-D/3-D registration is performed to match the 3-D virtual implant with the fluoroscopic image of 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 [28]. 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.

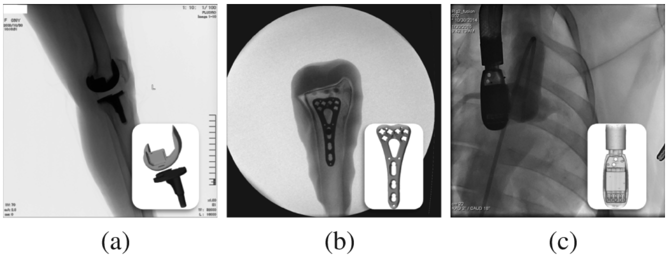

Fig. 5. Example data, including a 3-D model and a 2-D X-ray image of the object. (a) TKA. (b) VIPS. (c) XEF.

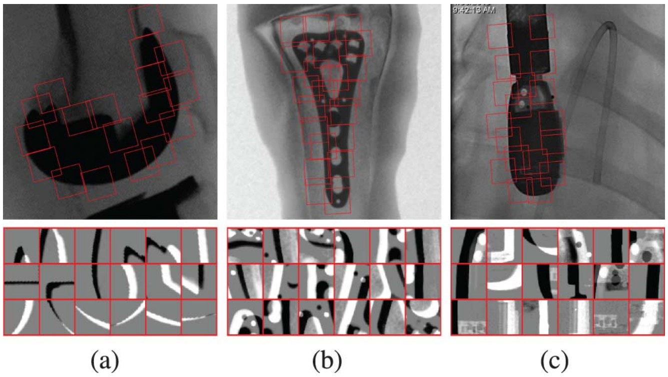

Fig. 6. Examples of local ROIs and LIRs. (a) TKA. (b) VIPS. (c) XEF.

Example datasets of the above 3 clinical applications are shown in Fig. 5. Examples of local ROIs and LIRs extracted from the 3 datasets are also shown in Fig. 6.

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) [12]. 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 1,000 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 I). 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. 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.

B. Synthetic Training Data Generation

The synthetic X-ray images used for training were generated by blending a DRR of the object with a background from real X-ray images:

$$I = I_{Xray} + \gamma \cdot G_\sigma * I_{DRR} + \mathcal{N}(a, b), \tag{19}$$

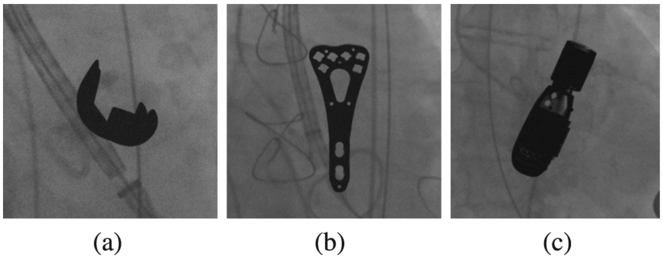

Fig. 7. Example synthetic X-ray images used for training. (a) TKA. (b) VIPS. (c) XEF.

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 between $[a, b]$. 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 use 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 will not be over-fitted for any specific type of background, which could vary significantly from case to case clinically. Examples of synthetic X-ray images of TKA, VIPS and XEF are shown in Fig. 7.

C. Evaluation Metrics

The registration accuracy was accessed with the mean Target Registration Error in the projection direction (mTREproj) [29], 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 mm, 61 mm and 37 mm, respectively. Therefore, the success criterion for the three applications were set to mTREProj less than 1.10 mm, 0.61 mm and 0.37 mm, which are equivalent to 2.8 pixels, 3.7 pixels 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 registration were successful [29]. Capture range is only reported for experiments where there are more than 20 samples within the capture range.

D. Performance Analysis

We 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 III-C, 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, LIR, HPR and PSP, by disabling them and demonstrating the detrimental effects on performance. The following 4 scenarios were evaluated for 10 iterations to compare with PEHL:

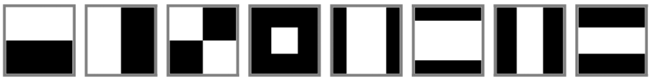

Fig. 8. HAAR features used in the experimenet “Without CNN”.

- **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. 8 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. 2 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 HPR**: The proposed CNN regression model shown in Fig. 4 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.

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 IV-A. PEHL and the best performed intensity-based method, MI_GC_Powell (will be detailed in the Section IV-E) 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.

E. Comparison With State-of-the-Art Methods

We first compare PEHL with several state-of-the-art intensity-based 2-D/3-D registration methods. An intensity-based method consists of two core components, an optimizer and a similarity measure. A recent study compared four popular optimizers (Powell's method, Nelder-Mead, BFGS, CMA-ES) for intensity-based 2-D/3-D registration, and concluded that Powell's method achieved the best performance [30]. Therefore, in all evaluated intensity-based methods, we used Powell's method as the optimizer. 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], [12]. For example, MI has been adopted in [3] for monitoring tumor motion during radiotherapy. CC computed on splatting DRR has been adopted in [12] for 5 Degree of Freedom pose estimation of TEE probe. GC has been adopted in [4] for the assessing the positioning and migration of bone implants. 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.

In addition to the above three methods, we implemented another intensity-based method combining MI and GC to achieve improved robustness and accuracy to compete with PEHL. 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 compared PEHL with CLARET, a linear regression-based 2-D/3-D registration method introduced in [16], which is closely related to PEHL, as it iteratively applies regressors on the image residual to estimate the transformation parameters. In [16], 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.

F. Experiment Environment

The experiments were conducted on a workstation with Intel Core i7-4790k CPU, 16GB 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 [31].

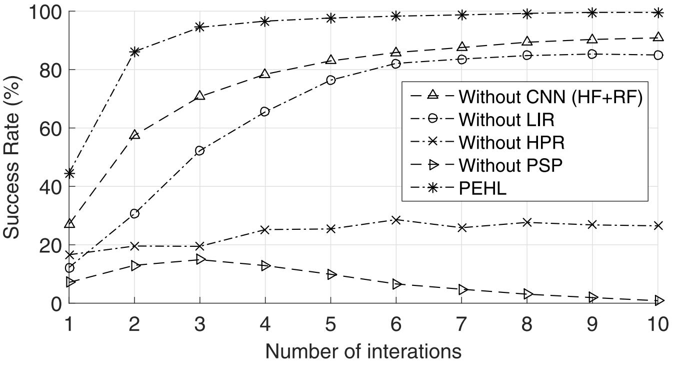

Fig. 9. Success rates of PEHL with 1 to 10 iterations. Four individual core components of PEHL, i.e., CNN, LIR, HPR and PSP, were disabled one at a time to demonstrate their detrimental effects on performance. Harr feature (HR) + Regression Forest (RF) was implemented to show the effect on performance without CNN. These results were generated on the XEF dataset.

V. Results

A. Performance Analysis

Fig. 9 shows the success rate as the number of iterations increases from 1 to 10 for 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 raising 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 $\sim$ 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 $\sim$ 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, HPR 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 3rd iteration dropped to 70.7%, indicating that the strong non-linear 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 3rd 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 HPR and PSP are disabled, the system almost completely failed, dropping the success rate at the 3rd iteration to 19.5% and 14.9%, respectively, suggesting that HPR and PSP are key components that make the regression problem solvable using the proposed CNN regression model.

Fig. 10 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 leads to RMSEproj = 0.005 mm). At the 3rd iteration, the RMSDproj of PEHL is 0.198 mm, which is 62% smaller than that of

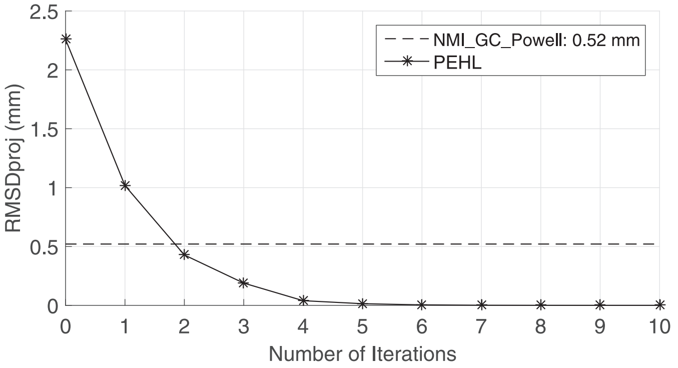

Fig. 10. RMSDproj from the registered locations of each target to their centroid using MI_GC_Powell and PEHL with 1 to 10 iterations. At $\text{Number of Iterations} = 0$, the RMSEproj at the perturbed positions without registration is shown. These results were generated on the XEF dataset.

TABLE II
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

MI_GC_Powell, i.e., 0.52 mm. These results suggest that PEHL has a significant advantage over MI_GC_Powell in terms of precision.

B. Comparison With State-of-the-Art Methods

We first observed that the linear regressor in CLARET completely failed in our experiment setup. Table II shows the root mean squared error (RMSE) of the 6 parameters yielded by PEHL and CLARET on the synthetic training data for XEF. 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 non-linear, and therefore cannot be reliably captured by a linear regressor. In comparison, PEHL employs the 3 algorithmic strategies to simplify the non-linear relationship and captures it using a CNN with a strong non-linear 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 accessed on real data from the three clinical applications.

Table III 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 owning 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 owning 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: 1) MI and CC are unable to resolve small mismatch; 2) GC is unable to recover large mismatch. By employing MI_Powell to recover large mismatch and GC_Powell to resolve small mismatch, MI_GC_Powell achieved much higher success rates, which is in line with our discussion in Section IV-E.

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 2nd 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 owning 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 $\sim$ 0.1 s per registration for all three applications, which is 20 ~ 45 times faster than that of MI_GC_Powell and leads to real-time registration at $\sim$ 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 owning 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. The distributions of mTREproj before and after registration using MI_GC_Powell and PEHL on the three applications are shown in Fig. 11.

VI. DISCUSSION AND CONCLUSION

In this paper, we presented a CNN regression approach, 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, HPR and PSP, to simplify the underlying mapping to be regressed, and designed a CNN regression model with strong non-linear 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

TABLE III

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	Success Rate	Capture Range (mm)	mTREproj Percentile (mm)					Running Time (s)
				10th	25th	50th	75th	90th	
TKA	Start	N/A	N/A	3.285	4.627	6.979	10.050	12.667	N/A
	MI_Powell	36.2%	N/A	0.437	0.746	1.693	6.238	8.421	1.37±0.44
	CC_Powell	43.8%	1.88	0.348	0.637	1.359	6.321	8.398	0.92±0.27
	GC_Powell	45.2%	2.14	0.330	0.588	1.313	7.615	9.765	2.52±1.22
	MI_GC_Powell	51.8%	2.83	**0.299**	0.521	1.048	6.408	8.614	3.11±0.94
	PEHL	**79.6%**	**7.23**	0.333	**0.444**	**0.593**	**0.903**	**6.733**	**0.11±0.00**
VIPS	Start	N/A	N/A	1.180	1.521	2.003	2.594	3.101	N/A
	MI_Powell	75.1%	N/A	0.156	0.234	0.375	0.604	0.917	1.66±0.60
	CC_Powell	57.7%	0.89	0.187	0.303	0.535	0.851	1.293	0.91±0.31
	GC_Powell	78.7%	1.12	0.121	0.207	0.325	0.543	2.283	3.91±1.55
	MI_GC_Powell	92.7%	2.77	**0.106**	**0.170**	0.259	**0.367**	0.535	4.71±1.59
	PEHL	**99.7%**	**5.51**	0.151	0.181	**0.244**	0.389	**0.451**	**0.10±0.00**
XEF	Start	N/A	N/A	1.048	1.369	1.826	2.307	2.790	N/A
	MI_Powell	69.7%	N/A	0.165	0.207	0.280	0.403	0.598	0.79±0.29
	CC_Powell	54.8%	N/A	0.117	0.168	0.321	0.893	1.173	0.40±0.10
	GC_Powell	56.9%	N/A	0.071	0.135	0.279	1.055	3.150	2.06±1.05
	MI_GC_Powell	89.1%	0.84	**0.047**	**0.098**	0.174	0.273	0.380	2.03±0.69
	PEHL	**94.5%**	**3.33**	0.082	0.113	**0.148**	**0.195**	**0.243**	**0.10±0.00**

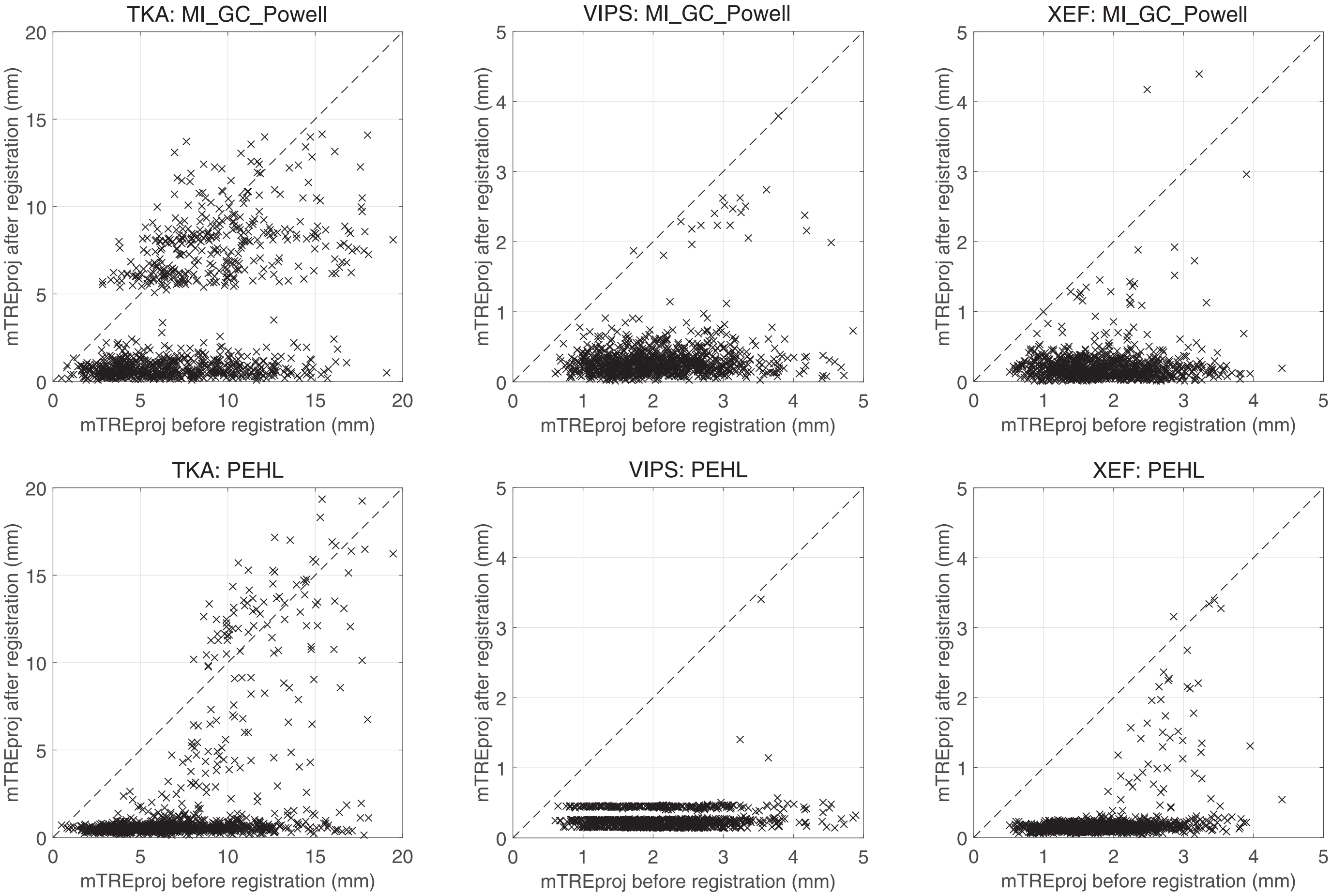

Fig. 11. mTREproj before and after registration using MI_GC_Powell and PEHL on TKA, VIPS and XEF applications.

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 owning 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) [9] and Direct Splatting Correlation (DSC) [12], 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 III, 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) [12].

PEHL employs HPR and PSP to break the complex regression problem to several much simpler problems. The PSP strategy divides the parameter space of out-of-plane rotations into small zones, and trains regressors for each zone separately. Smaller zones will make the regression task simpler, but at the same time increase the training effort and memory consumption at the runtime. In this paper, we empirically selected the size of each zone in the PSP strategy to be 20×20 degrees, which leads to satisfactory registration accuracy and robustness and keeps the number of zones manageable (i.e., 324). The HPR strategy divides the 6 parameters into 3 groups, and trains regressors for each group separately. Therefore, there are in total $324 \times 3 = 972$ regressors to be trained and loaded at the runtime. In order to make the memory footprint required by the 972 regressors manageable at runtime, we use a weight sharing mechanism in the CNN model, which leads to 2.39 GB total memory consumption. In comparison, without weight sharing, pre-loading the same CNN regression model used in PEHL requires 13.7 GB RAM, which could be impractical in a clinical setup.

Like any machine learning-based methods, 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 a high performance when applying PEHL on real X-ray images. However, it's 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.

One of our future work is to investigate the possibility of sharing the CNN weights across groups and/or zones, so that the memory footprint can be further reduced, making more complex and deeper network structures affordable. The difficulty of sharing the CNN across groups and/or zones lies in the fact that the training data for different groups and zones are different, which makes the training of the shared CNN a non-trivial task. Another future work is to extend PEHL for multi-plane 2-D/3-D registration, i.e., registering one 3-D model to multiple X-ray images acquired from different angles. This is currently a limitation of PEHL compared to intensity-based methods, which can be straightforwardly extended from mono-plane to multi-plane setup by simply combining the similarity measures from all the planes into one value. One possible way to achieve PEHL for multi-plane registration would be to apply regression on each plane separately and then combine all regression results into one estimation.

Acknowledgment

The authors would like to thank Dr. G. Wang's team from Tsinghua University for providing the TKA data, Siemens Healthcare Business Unit AX (Angiography & Interventional X-Ray Systems) for providing the XEF data and Siemens Healthcare Business Unit XP (X-Ray Products) for providing the VIPS data.

References

[1] R. Liao, L. Zhang, Y. Sun, S. Miao, and C. Chefd'Hotel, “A review of recent advances in registration techniques applied to minimally invasive therapy,” *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 983–1000, Aug. 2013.

[2] P. Markelj, D. Tomaževič, B. Likar, and F. Pernuš, “A review of 3D/2D registration methods for image-guided interventions,” *Med. Image Anal.*, vol. 16, no. 3, pp. 642–661, 2012.

[3] C. Gendrin *et al.*, “Monitoring tumor motion by real time 2D/3D registration during radiotherapy,” *Radiother. Oncol.*, vol. 102, no. 2, pp. 274–280, 2012.

[4] J. Schmid and C. Chênes, “Segmentation of X-ray images by 3D-2D registration based on multibody physics,” in *Comput. Vis.*, 2014, pp. 674–687.

[5] S. Miao, R. Liao, and Y. 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 *Proc. 2011 IEEE Int. Symp. Biomed. Imag., From Nano to Macro*, 2011, pp. 1215–1218.

[6] R. A. McLaughlin, J. Hipwell, D. J. Hawkes, J. A. Noble, J. V. Byrne, and T. Cox, "A comparison of 2D-3D intensity-based registration and feature-based registration for neurointerventions," in *Proc. MICCAI*, 2002, pp. 517–524.

[7] A. Varnavas, T. Carrell, and G. Penney, "Fully automated 2D–3D registration and verification," *Med. Image Anal.*, vol. 26, no. 1, pp. 108–119, 2015.

[8] S. Miao, R. Liao, J. Lucas, and C. Chefd'hotel, "Toward accurate and robust 2-D/3-D registration of implant models to single-plane fluoroscopy," in *Augment. Reality Environ. Med. Imag. Comput.-Assist. Intervent.*, 2013, pp. 97–106.

[9] L. Zöllei, E. Grimson, A. Norbash, and W. Wells, "2D-3D rigid registration of X-ray fluoroscopy and CT images using mutual information and sparsely sampled histogram estimators," in *Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 2, pp. II–696.

[10] W. Birkfellner *et al.*, "Stochastic rank correlation: A robust merit function for 2D/3D registration of image data obtained at different energies," *Med. Phys.*, vol. 36, no. 8, pp. 3420–3428, 2009.

[11] L. Westover, "Footprint evaluation for volume rendering," *ACM Siggraph Comput. Graph.*, vol. 24, no. 4, pp. 367–376, 1990.

[12] C. Hatt, M. Speidel, and A. Raval, "Robust 5dof transesophageal echo probe tracking at fluoroscopic frame rates," in *Proc. MICCAI*, 2015.

[13] J. Kruger, R. Westermann, and A. Raval, "Acceleration techniques for GPU-based volume rendering," in *Proc. 14th IEEE Visualizat.*, 2003, p. 38.

[14] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3594–3601.

[15] F. Michel, M. Bronstein, A. Bronstein, and N. Paragios, "Boosted metric learning for 3D multi-modal deformable registration," in *Proc. IEEE Int. Symp. Biomed. Imag. From Nano to Macro*, 2011, pp. 1209–1214.

[16] A. R. Gouveia, C. Metz, L. Freire, P. Almeida, and S. Klein, "Registration-by-regression of coronary CTA and X-ray angiography," *Comput. Methods Biomechan. Biomed. Eng., Imag. Visualizat.*, pp. 1–13, 2015.

[17] C.-R. Chou, B. Frederick, G. Mageras, S. Chang, and S. Pizer, "2D/3D image registration using regression learning," *Comput. Vis. Image Understand.*, vol. 117, no. 9, pp. 1095–1106, 2013.

[18] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3D pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3109–3118.

[19] P. Dollár, P. Welinder, and P. Perona, "Cascaded pose regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1078–1085.

[20] C. Zach, A. Penate-Sanchez, and M.-T. Pham, "A dynamic programming approach for fast and robust object pose recognition from range images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 196–203.

[21] R. Mottaghi, Y. Xiang, and S. Savarese, "A coarse-to-fine model for 3D pose estimation and sub-category recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 418–426.

[22] M. Kaiser *et al.*, "2D/3D registration of TEE probe from two non-orthogonal c-arm directions," in *Proc. MICCAI*, 2014, pp. 283–290.

[23] D. Tomaževič, B. Likar, T. Slivnik, and F. Pernuš, "3-D/2-D registration of CT and MR to X-ray images," *IEEE Trans. Med. Imag.*, vol. 22, no. 11, pp. 1407–1416, Nov. 2003.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, pp. 1097–1105, 2012.

[25] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2010, pp. 249–256.

[26] Z. Zhu, S. Ji, M. Yang, H. Ding, and 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 Congr. Med. Phys. Biomed. Eng.*, Beijing, China, May 26–31, 2012, pp. 230–233.

[27] S. Vetter, I. Mühlhäuser, J. v. Recum, P.-A. Grützner, and J. Franke, "Validation of a virtual implant planning system (VIPS) in distal radius fractures," *Bone Joint J. Orthopaed. Proc. Suppl.*, vol. 96, no. SUPP 16, pp. 50–50, 2014.

[28] G. Gao *et al.*, "Registration of 3D trans-esophageal echocardiography to X-ray fluoroscopy using image-based probe tracking," *Med. Image Anal.*, vol. 16, no. 1, pp. 38–49, 2012.

[29] E. B. De Kraats, G. P. Penney, D. Tomaževič, T. Van Walsum, and W. J. Niessen, "Standardized evaluation methodology for 2-D-3-D registration," *IEEE Trans. Med. Imag.*, vol. 24, no. 9, pp. 1177–1189, Sep. 2005.

[30] M. Kaiser, M. John, T. Heimann, T. Neumuth, and G. Rose, "Comparison of optimizers for 2D/3D registration for fusion of ultrasound and X-ray," in *Proc. Workshops Bildverarbeitung für die Medizin 2014, Algorithmen-Systeme-Anwendungen*, 2014, p. 312.

[31] Y. Jia *et al.*, Caffe: Convolutional architecture for fast feature embedding ArXiv, 2014 [Online]. Available: arXiv:1408.5093