Multiview 2D/3D Rigid Registration via a Point-Of-Interest Network for Tracking and Triangulation

Haofu Liao[1]*, Wei-An Lin[2]*, Jiarui Zhang[3], Jingdan Zhang[4], Jiebo Luo[1], S. Kevin Zhou[5,6]
[1]University of Rochester [2]University of Maryland, College Park [3]Rutgers University
[4]Z2W Corporation [5]Chinese Academy of Sciences [6]Peng Cheng Laboratory, Shenzhen

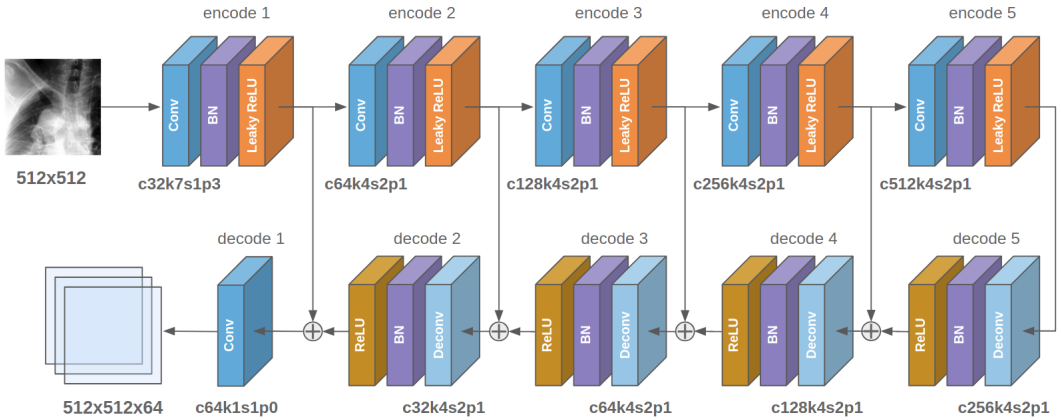

Figure 1: The detailed architecture of the U-Net branch of the POINT network.

A. Network Architecture

Figure 1 shows details of the U-Net branches of the POINT network. There are five pairs of encoding and decoding blocks in the network. Each pair is connected with a skip-connection that shuttles the outputs of the encoding block to the decoding block directly. The layer configurations are given blew the convolutional and deconvolutional layers, where "c", "k", "s" and "p" denote the channel, kernel, stride and padding size, respectively. The "+" symbol at the end of a skip-connection denotes a concatenation operation that combines feature maps from the encoding block "encode n" and a lower level decoding block "decode n+1".

Figure 2 illustrates the feature extraction process for a 3×3 feature kernel. Given a feature map $\mathbf{F}^D$ of size $M \times N \times C$ obtained from the DRR U-Net branch and a set of POIs $\mathcal{P} = \{\mathbf{x}_1^{\mathrm{D}}, \mathbf{x}_2^{\mathrm{D}}, \ldots, \mathbf{x}_m^{\mathrm{D}}\}$, the feature extraction layer first interpolates each channel of the feature map $\mathbf{F}^D$ at $\forall \mathbf{x}_i^{\mathrm{D}} \in \mathcal{P}$, which gives a feature vector of size $1 \times 1 \times C$. Then, to obtain a richer feature representation, it also interpolates the neighborhood of $\mathbf{x}_i^{\mathrm{D}}$ which results a feature kernel $\mathbf{F}^D(\mathbf{x}_i^{\mathrm{D}})$ of size $3 \times 3 \times C$.

* indicates equal contributions.

B. Derivation

In this section, we show how

$$\mathbf{D}(\mathbf{x})\mathbf{R}_{\text{view}}\mathbf{X} = c\mathbf{x} - \mathbf{D}(\mathbf{x})\mathbf{t}_{\text{view}} \tag{1}$$

can be derived from

$$\mathbf{x}' = \mathbf{K}\begin{bmatrix}\mathbf{R}_{\text{view}} & \mathbf{t}_{\text{view}} + \mathbf{h}\end{bmatrix}\begin{pmatrix}\mathbf{X}\\ 1\end{pmatrix}. \tag{2}$$

Proof. Let

$$\tilde{\mathbf{X}} = \mathbf{R}_{\text{view}}\mathbf{X} + \mathbf{t}_{\text{view}} \tag{3}$$

and recall that

$$\mathbf{K} = \begin{bmatrix}-d\mathbf{I} & 0\\ \mathbf{0} & 1\end{bmatrix}, \tag{4}$$

then (2) can be written as

$$\begin{aligned}\mathbf{x}' = \begin{pmatrix}x'\\ y'\\ z'\end{pmatrix} &= \begin{bmatrix}-d\mathbf{I} & 0\\ \mathbf{0} & 1\end{bmatrix}(\tilde{\mathbf{X}} + \mathbf{h})\\ &= \begin{bmatrix}\begin{bmatrix}-d\mathbf{I} & 0\end{bmatrix}(\tilde{\mathbf{X}} + \mathbf{h})\\ \begin{bmatrix}\mathbf{0} & 1\end{bmatrix}(\tilde{\mathbf{X}} + \mathbf{h})\end{bmatrix}\end{aligned} \tag{5}$$

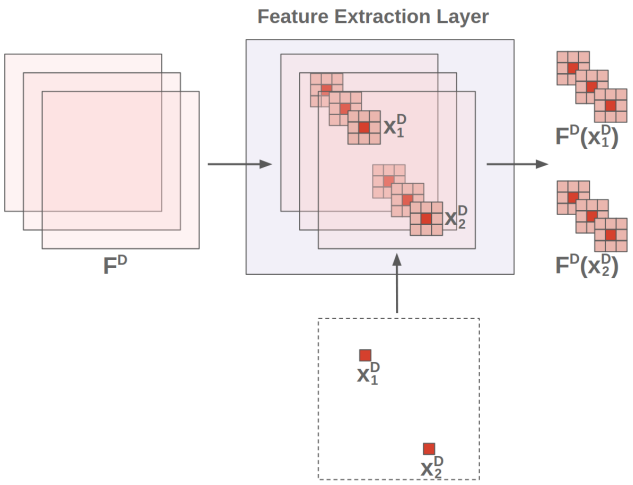

Figure 2: An illustration of the feature extraction process of a 3×3 kernel.

Since $\mathbf{x} = (x'/z', y'/z')^T$, we can get

$$\mathbf{x} = \frac{\begin{bmatrix} -d\mathbf{I} & 0 \end{bmatrix} (\tilde{\mathbf{X}} + \mathbf{h})}{\begin{bmatrix} \mathbf{0} & 1 \end{bmatrix} (\tilde{\mathbf{X}} + \mathbf{h})} \tag{6}$$

$$\begin{bmatrix} d\mathbf{I} & \mathbf{x} \end{bmatrix} \tilde{\mathbf{X}} = \begin{bmatrix} -d\mathbf{I} & -\mathbf{x} \end{bmatrix} \mathbf{h} \tag{7}$$

Also, recall that

$$\mathbf{h} = \begin{pmatrix} 0 \\ 0 \\ -c \end{pmatrix}, \mathbf{D}(\mathbf{x}) = \begin{bmatrix} d & 0 & \mathbf{x} \\ 0 & d & \end{bmatrix}, \tag{8}$$

we can simplify (7) as

$$\mathbf{D}(\mathbf{x})\tilde{\mathbf{X}} = c\mathbf{x}. \tag{9}$$

Finally, using (3), we have

$$\mathbf{D}(\mathbf{x})\mathbf{R}_{\text{view}}\mathbf{X} = c\mathbf{x} - \mathbf{D}(\mathbf{x})\mathbf{t}_{\text{view}} \tag{10}$$

□

C. POI Tracking Results

See next page.