

Deep Learning Architecture for Dense Synthetic Aperture Radar Image Registration

Cole Reynolds, Brandon Boucher, John Lusk, W. Bryan Bell

***Abstract*—Synthetic Aperture Radar (SAR) images are large, non-optical, complex-valued images commonly captured from high-altitude planes or satellites. The registration of two SAR images involves applying a spatial mapping to the coordinates of the first image (referred to as the moving image) in order to maximize some similarity metric with the second image (referred to as the fixed image), and serves as a crucial first step for many tasks such as interferometry, image fusion, and target recognition. The unique characteristics of SAR images (speckle statistics, geometric distortions, image formatting, etc...), corroborated by the lack of standardized datasets, severely limits the applications of existing computer vision architectures to SAR images, resulting in the use of less robust, but more approachable and conventional registration techniques such as SAR-SIFT. To address these issues, we propose an intensity-based deep learning architecture that competes with SAR-SIFT in various similarity metrics while having a significantly faster registration time. Moreover, it can be trained on minimal data with a single laptop GPU, making it approachable to researchers and practitioners with limited computational resources.**

***Index Terms*—Image registration, Auto-encoder, Computer vision, Synthetic Aperture Radar (SAR).**

I. Introduction

SYNTHETIC aperture radar (SAR) is a well known medium for high-altitude remote sensing, capable of producing images independent of weather or solar illumination, making it a great tool from a surveying perspective. Unfortunately, SAR images are very difficult to handle from a computer vision perspective. Most famously, speckle noise caused by coherent interference degrades the final image quality, making edges and corners harder to identify, and renders many conventional feature-based approaches like SIFT [1] unreliable. Furthermore, SAR's larger image sizes combined with the lack of universal, quality training datasets [2] make deep learning approaches impractical and/or unfeasible. One reason for the lack of useful datasets can be attributed to SAR's active sensing method. The variety of processing techniques and use cases [3], exacerbated by government, military, and intelligence interests, often result in pixel statistics and radiometric/geometric distortions that are unique to the user and SAR platform, and leads to a lack of standardization in the SAR community. Medical imaging, as a comparison, benefits from well-established standards and protocols that ensure consistent data acquisition, allowing for the creation of large, universal datasets that can be used to train deep learning models [4], [5], [6], [7], [8]. Medical imaging also tends to focus on specific, well-defined regions of interest, reducing the complexity of deep learning tasks.

As a result, conventional techniques like SAR-SIFT [9] remain the preferred choice for SAR image registration, since it seems unlikely that a single deep learning model can be adopted by the general community while effectively register SAR images across all environments. The sheer diversity of image parameters and statistics, combined with the aforementioned limited availability of large-scale, annotated datasets, makes it impractical to expect a single network to generalize well to all scenarios. Even if such a network were possible, fine-tuning its weights to achieve optimal performance for a specific environment would likely require significant computational resources, which may not be feasible for many researchers. We argue that building light-weight, stable neural networks tailored to specific environments of interest offer a more practical direction for the field at this time. In this paper we propose and architecture that's accurate enough to be used alone, yet fast and light enough to be easily integrated alongside other registration techniques. We showcase a model's results on synthetically generated 1024×1024 sized image pairs taken from an open source SAR image library, and trained on an NVIDIA RTX 3500 Ada Generation Laptop GPU. We discuss related work in section II, and present the architecture in section III. Data generation is described in section IV. Results are described in section V, followed by a discussion in section VI, and the paper is concluded in section VII.

II. Related Work

Image warping is a fundamental technique in computer vision and image processing, with a wide range of applications in various fields. One of the primary applications of image warping is in image registration, where it is used to align multiple images taken from different viewpoints or at different times [10], [11], [12]. Image warping is also commonly used in stereo vision to rectify stereo images and compute depth maps [13], as well as video editing, special effects, and augmented reality [14], [15], [16], [17]. Deep learning techniques have been increasingly applied to image registration tasks, achieving state-of-the-art performance in various fields, and with the introduction of the Spatial Transformer [18], spatial transformations have become increasingly popular as intermediate steps within deep learning architectures [19], [20], [21]. Deep learning-based methods for deformable medical image registration [22], [23], [24], [25] and determining optical flow [26], [27], [28], [29] all use Convolutional Neural Networks

C. Reynolds, B. Boucher, J. Lusk, and W. B. Bell are with the Advanced Development Program, Lockheed Martin Aeronautics, Fort Worth, Texas.

Fig. 1. Image registration of 1024×1024 sub-images from the Umbra Open Data Catalog's Texas A&M Farm Plot SAR data. From top to bottom: Moving image, Fixed image, Registered image.

(CNN) [30], but typically come in one of two architecture types. 1) The U-Net [24], where the images are first paired by concatenation along the channel dimension and the spatial transformation is the result of applying the U-Net to the pair. 2) A cosine-similarity based cost-volume predicated on a Siamese CNN architecture, where the CNN encodes features of each image separately, before they are cross-coupled to each other by a similarity metric. The most common similarity metric is the dot product, but other similarities such as a weighted dot-product or non-linear cross-attention could be applied (we consider vision transformers [31] and their variants to be a generalization of this architecture). Unfortunately, both architectures tend to fail when applied to our images of interest - large, high-resolution SAR images. We believe this is due to an over-emphasis of encoding independent image features and an under-emphasis of encoding the relations between image pairs. To elaborate, we consider a canonical example. Let $\boldsymbol{x}(\boldsymbol{r})$ and $\boldsymbol{y}(\boldsymbol{r})$ be our image pair, and let $\boldsymbol{y}$ be a uniform translation of $\boldsymbol{x}$ such that $\boldsymbol{y}(\boldsymbol{r}) = \boldsymbol{x}(\boldsymbol{r} + \boldsymbol{r}_0)$ for some fixed $\boldsymbol{r}_0$. Then it can be easily verified by the Fourier shift theorem that the phase-correlation, $R(\boldsymbol{r})$, of the two images is a delta function centered at $\boldsymbol{r}_0$,

$$
\begin{aligned}
R(\boldsymbol{r}) &= \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{x(\boldsymbol{r})\} \circ \mathcal{F}\{y(\boldsymbol{r})\}^*}{|\mathcal{F}\{x(\boldsymbol{r})\} \circ \mathcal{F}\{y(\boldsymbol{r})\}^*|}\right\} && \text{(1a)}\\
&= \mathcal{F}^{-1}\left\{\frac{\hat{x}(\boldsymbol{k}) \circ \hat{y}(\boldsymbol{k})^*}{|\hat{x}(\boldsymbol{k}) \circ \hat{y}(\boldsymbol{k})^*|}\right\} && \text{(1b)}\\
&= \mathcal{F}^{-1}\left\{\frac{\hat{x}(\boldsymbol{k}) \circ \hat{x}(\boldsymbol{k})^* e^{-i\boldsymbol{k}\cdot\boldsymbol{r}_0}}{|\hat{x}(\boldsymbol{k}) \circ \hat{x}(\boldsymbol{k})^* e^{-i\boldsymbol{k}\cdot\boldsymbol{r}_0}|}\right\} && \text{(1c)}\\
&= \mathcal{F}^{-1}\left\{e^{-i\boldsymbol{k}\cdot\boldsymbol{r}_0}\right\} && \text{(1d)}\\
&= \delta(\boldsymbol{r} - \boldsymbol{r}_0), && \text{(1e)}
\end{aligned}
$$

where $\mathcal{F}\{\cdot\}$ is the 2-d Fourier transform, $(\cdot)^*$ denotes complex conjugation, $\circ$ is the element-wise (Hadamart) product, and $\hat{x}(\boldsymbol{k}) = \mathcal{F}\{\boldsymbol{x}(\boldsymbol{r})\}$. The global transformation is thus encoded as a delta function, and can be decoded by a simple integral:

$$
\begin{aligned}
\boldsymbol{\phi}(\boldsymbol{r}) &= \boldsymbol{r} + \int \delta(\boldsymbol{r}' - \boldsymbol{r}_0)\boldsymbol{r}' \, \mathrm{d}\boldsymbol{r}' \\
&= \boldsymbol{r} + \boldsymbol{r}_0.
\end{aligned} \tag{2}
$$

Our architecture is heavily inspired by this canonical example, foregoing learned feature maps in favor of learned embedding pairs that encode non-trivial transformations between images. In [32], [33], [34], gated networks and factored gated networks were used to generate transformed images from translated, rotated, and piecewise translated image pairs. In fact, factored gated networks are nearly identical in structure to the Fourier-correlation, although this discussion might have been avoided in [33] since their objective was to generate transformed images as opposed to determining the underlying transformation.

III. Network Architecture

The primary component of the model is the auto-encoder, taking the moving, $\boldsymbol{x}(\boldsymbol{r})$, and fixed, $\boldsymbol{y}(\boldsymbol{r})$, images as inputs and outputting a down-sampled version of the spatial transformation function $\boldsymbol{\phi}(\boldsymbol{r})$ that should be applied to $\boldsymbol{x}$ such that $(\boldsymbol{x} \circ \boldsymbol{\phi})(\boldsymbol{r}) \approx \boldsymbol{y}(\boldsymbol{r})$. For 256×256 sized image pairs, we found that a single auto-encoder is sufficient for image registration. Unfortunately, larger image pairs require a larger

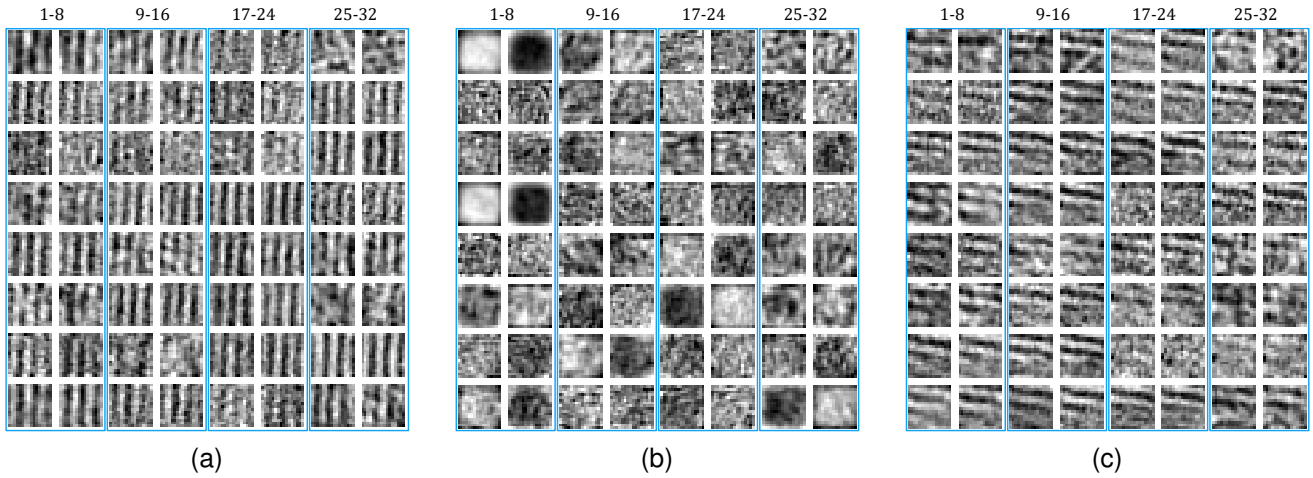

Fig. 2. Learned embedding pairs for 32 moving/fixed features maps that represent different spatial transformations for the presented model. Within each column is a pair of embeddings for a moving image feature map (left) and a fixed image feature map (right). (a) Uniform spatial transformation. (b) Identity spatial transformation. (c) Localized spatial transformation.

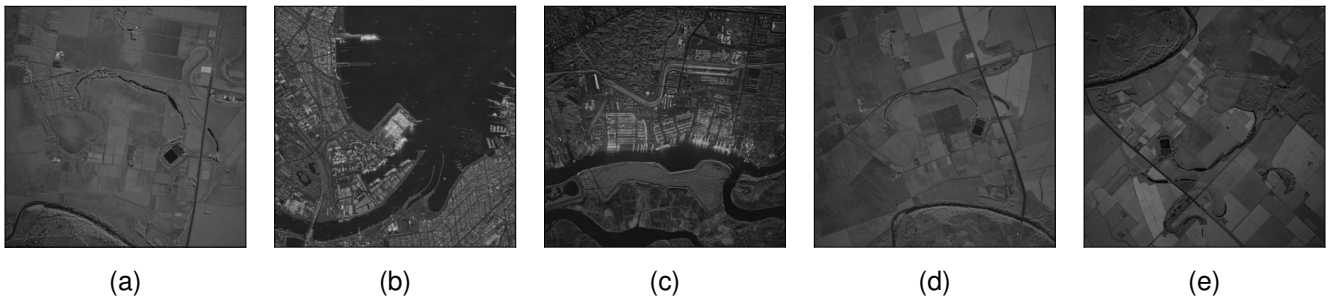

Fig. 3. Training and testing SAR images used for the presented model, each at 0.25 meter resolution. Despite the testing farm plot images having different geographical features from the single training farm plot images, the model is still able to accurately register the testing images. (a) Texas A&M Farm Plot training image. (b) Melbourne, Australia training image. (c) Port of Savannah, Georgia, USA training image. (d) Texas A&M Farm Plot testing image 1. (e) Texas A&M Farm Plot testing image 2.

architecture, as naively down-sampling large $h \times w$ image pairs to 256×256, determining $\boldsymbol{\phi}$, and then interpolating $\boldsymbol{\phi}$ up to $h \times w$ only interpolates the warping error, and doesn't use the increased resolution of a larger image to refine the spatial transformation. The fine registration auto-encoder addresses this issue. For 1024×1024 sized image pairs, we down-sample the images to 256×256, determine the coarse scale warping $\boldsymbol{\phi}_c$, and then use $(\boldsymbol{x} \circ \boldsymbol{\phi}_c)$ and $\boldsymbol{y}$ as the image pair input for the fine registration auto-encoder.

A. Encoder

The general encoder architecture is illustrated in Fig 4. The encoder first employs a small CNN with the primary purpose of performing spatial dimensionality reduction on the input images, transforming the initial $[\,1,\, h,\, w\,]$ representations into more compact $[\,c,\, h',\, w'\,]$ feature spaces. Moving spatial information from the spatial dimensions to the channel dimension (c) reduces the computational complexity of subsequent spatial attention operations in the decoder.

Let $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$ denote the feature space representations of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. We partition the representations along the spatial dimensions into blocks of size $[c, \Delta, \Delta]$, and view the representations as a coarse, $[h'/\Delta, w'/\Delta]$ sized grid of $[c, \Delta, \Delta]$ sized blocks (see Fig. 5). Going forward, we re-parameterize $\tilde{\boldsymbol{x}}$ as $\tilde{x}^{\alpha\beta}_{c,\,ij}$, where c denotes the channels, α and β denote the grid indices, and i and j denote the spatial indices within the grid. An identical treatment is applied to $\tilde{\boldsymbol{y}}$. Following the structure of Eq. 1b, learned projections $W^{(x)}_{cn,ij}$ and $W^{(y)}_{cn,ij}$ map $\tilde{x}^{\alpha\beta}_{c,\,ij}$ to $\hat{x}^{\alpha\beta}_{n}$, and $\tilde{y}^{\alpha\beta}_{c,\,ij}$ to $\hat{y}^{\alpha\beta}_{n}$, respectively, before being multiplied entry-wise and fed through a final feed-forward, fully connected network to produce the encoded units $\tilde{g}^{\alpha\beta}_{k}$ (see Eq. 3 and Fig. 4a). This part of the encoder acts as a generalized and localized version of Eq. 1.

$$\begin{aligned}\tilde{g}^{\alpha\beta}_{k} &= \sigma\left(\left[\sum_{c,\,i,\,j} \tilde{x}^{\alpha\beta}_{c,\,ij} W^{(x)}_{cn,ij}\right] \circ \left[\sum_{c,\,i,\,j} \tilde{y}^{\alpha\beta}_{c,\,ij} W^{(y)}_{cn,ij}\right]\right) \\ &= \sigma\left(\hat{x}^{\alpha\beta}_{n} \circ \hat{y}^{\alpha\beta}_{n}\right).\end{aligned} \tag{3}$$

The inverse Fourier transform in Eq. 1 is replaced by $\sigma(\cdot)$ in Eq. 3. The linear projections $W^{(x)}_{cn,ij}$ and $W^{(y)}_{cn,ij}$ become highly correlated during training, as shown in Fig. 2, suggesting that the model does learn a generalization of the phase correlation operation in order to perform the image registration.

We noticed that the activation functions used within the feed-forward network greatly influenced the convergence of the model, with the ones presented yielding the best results.

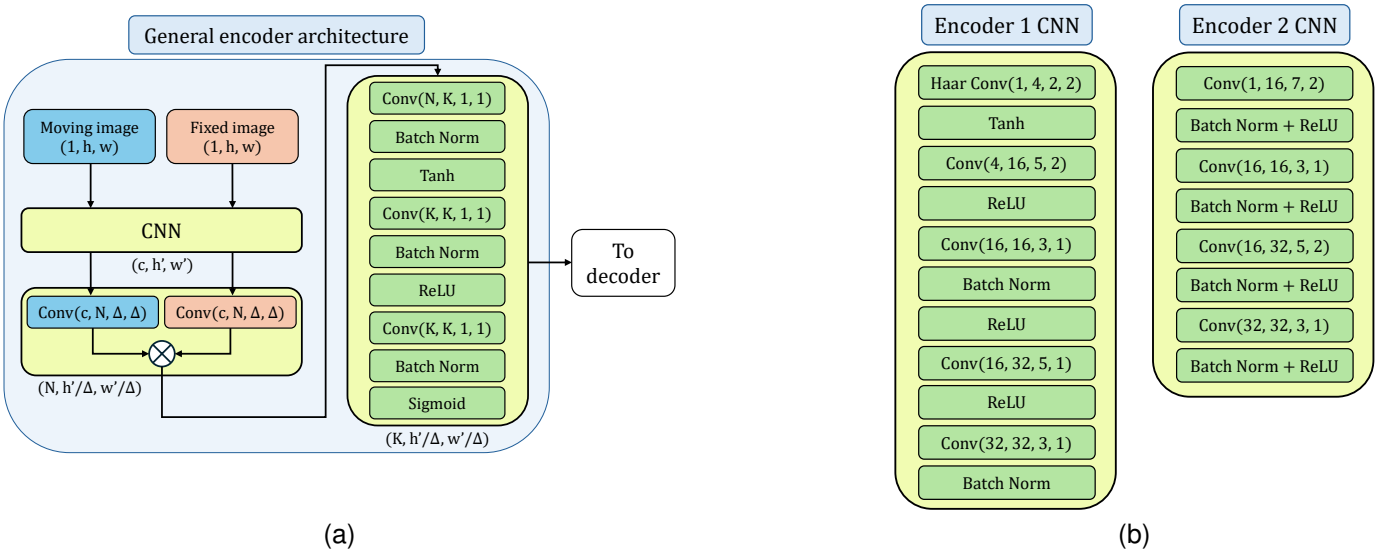

Fig. 4. Encoder architecture. The parenthesis underneath each section indicates the output size (channel, height, width) of that section, and the indexes within Conv(·) represent (in order) input channels, output channels, kernel size, stride. Convolutions with odd-sized kernels are zero-padded to maintain original spatial dimensions (before accounting for stride). (a) General encoder architecture. The ⊗ indicates entry-wise (Hadamart) product (b) Specific encoder CNN architectures for the presented model's two encoders used to register 1024×1024 image pairs. The Haar Conv(·) is the Haar discrete wavelet transform and those kernels are not updating during training.

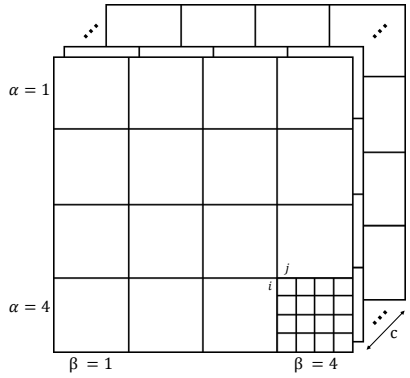

Fig. 5. Partition of $[c, h', w']$ sized feature space into spatially localized blocks (indexed by α, β) of size $[c, \Delta, \Delta]$ and the relative features within each block (indexed by $i, j \in \{1, 2, \ldots, \Delta\}$).

B. Decoder

The output from the encoder, $\tilde{g}_k^{\alpha\beta}$, requires two separate types of decoding: Spatial decoding and transformation decoding. Our spatial decoding architecture is influenced by the integral operation in Eq. 2, while the transformation decoding architecture uses more traditional deep learning architectures to convert the high dimensional embeddings at each pixel location to a 2-dimensional vector.

1) Spatial Decoding: One major limitation of the encoder described above is the independence of each grid index. To output a smooth transformation function, the decoder needs to couple the spatially localized encoding blocks, and we do so with two separate non-standard form of **QKV** self-attention [35], [36], using fully convolutional networks instead of a linear projection, and is illustrated in Fig. 7. The primary purpose of the self-attention network (Eq. 4a) depicted on the left in Fig. 6 is to encourage continuity of the final spatial transformation by encouraging similarity of the encoded transformation embeddings.

$$v_k^{\alpha\beta} = \tilde{g}_k^{\alpha\beta} + \sum_{\mu\nu} \underset{\mu,\,\nu}{\text{Softmax}} \left(\sum_k \mathbf{Q}(\tilde{g})_k^{\alpha\beta}\, \mathbf{K}(\tilde{g})_k^{\mu\nu} \right) \mathbf{V}(\tilde{g})_k^{\mu\nu} \tag{4a}$$

$$q_{ij}^{\alpha\beta} = \underset{\alpha,\,\beta}{\text{Softmax}} \left(\sum_k \mathbf{Q}'(\tilde{g})_k^{\alpha\beta}\, \mathbf{P}_{ij}^k \right) \tag{4b}$$

$$\boldsymbol{\phi}(\boldsymbol{r}) = \boldsymbol{r} + \sigma \left(\sum_{\alpha\beta} v_k^{\alpha\beta} q_{ij}^{\alpha\beta} \right). \tag{4c}$$

The network depicted on the right in Fig. 6 and described by Eq. 4b, which we call the spatial projection network, takes the self-attention network output and a learned "embedding spatial projections" tensor to establish a projection between the grid indices α, β and the image domain coordinate indices i, j. The summation over the grid coordinates α, β in Eq. 4c to produce spatial coordinates is akin to the integral in Eq. 2, where the output of the spatial projection network $q_{ij}^{\alpha\beta}$ is used instead of the delta function.

2) Transformation Decoding: The $\sigma(\cdot)$ function in Eq. 4c is the transformation decoding network, illustrated in Fig. 7, and it outputs the dense displacement for each pixel in the moving image. The fine registration auto-encoder uses a shallow U-Net to increase the spatial dimensions, concatenating the shallow feature maps from the encoder to bilinearly upsampled convolution layers.

C. End-to-End Architecture

Given two $h \times w$ SAR images, we first determine the $2 \times h \times w$ coarse warping $\boldsymbol{\phi}_c(\boldsymbol{r})$ by sub-sampling the original

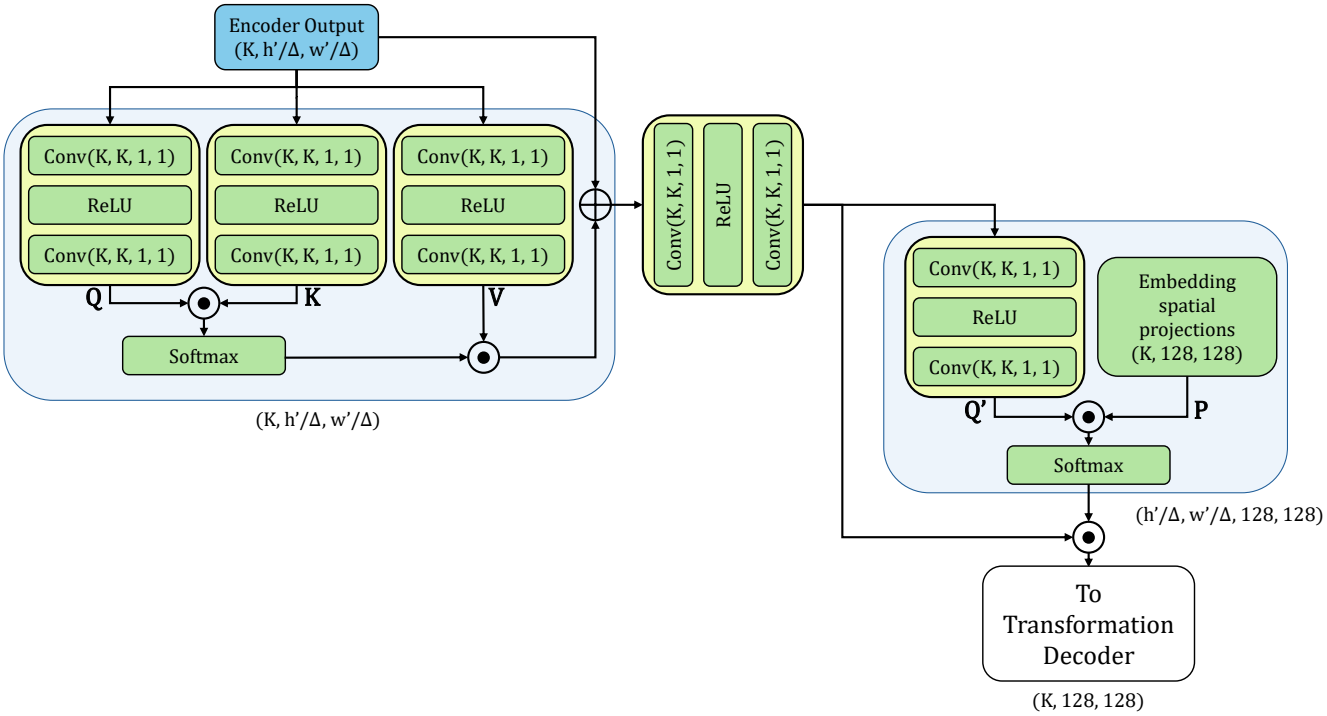

Fig. 6. Spatial decoder architecture. The $\odot$ operation denotes tensor contraction. See Eq. 4 for explicit details regarding which dimensions are contracted for each operation.

images down to 256×256, determining the $2 \times 256 \times 256$ coarse warping from the coarse registration auto-encoder, and then interpolating it up to $2 \times h \times w$. The coarsely registered moving image, $(\boldsymbol{x} \circ \boldsymbol{\phi}_c)(\boldsymbol{r})$, and the fixed image, $\boldsymbol{y}(\boldsymbol{r})$ are then fed into the fine registration auto-encoder to produce the final $\boldsymbol{\phi}(\boldsymbol{r})$. The coarse and fine registration auto-encoders have identical architectures except for the initial CNNs and decoder. Since we down-sample the images to 256×256 for the coarse registration, smaller kernel sizes in the CNN can be used to extract meaningful information from the image. For larger and higher-resolution images, we use larger kernel sizes in the CNN to suppress the influence of the higher frequency speckle noise. The larger image size of the fine registration encoder leads to a larger grid size within the auto-encoder. For $\Delta = 16$, $h = 1024$ and $w = 1024$, the coarse registration encoder produces a 4×4 spatial grid while the fine registration encoder produces a 16×16 spatial grid.

IV. Data Generation and Training

We trained the entire network for 1024×1024 image pairs, end-to-end, in just a few hours on an HP laptop with an Intel i9-13950HX CPU, 64GB of RAM, and an NVIDIA RTX 3500 Ada Generation Laptop GPU.

A. Data Generation

To our knowledge, there is no publicly available dataset for SAR-SAR image registration, so we constructed our own small dataset for this paper from five separate SAR images in the Umbra Open Data Catalog [37]. We selected one image from each of the Texas A&M Farm Plot, the Port of Savanna, Georgia, United States, and the Melbourne, Australia datasets, and used two separate images from the Texas A&M Farm Plot dataset as testing data. Each image was rescaled to 8-bit integer values. The five images are displayed in Fig. 3.

To generate a moving image in our dataset, we randomly sample from the three training images, and then randomly down-sample the image to 0.5m-1m resolution before randomly rotating and selecting a 1024×1024 region within the selected image. We repeat these steps to generate our moving image set, and denote this set by $\{\boldsymbol{x}_i\}$, where $\boldsymbol{x}_i$ is the i-th image. To construct our fixed image set, denoted by $\{\boldsymbol{y}_i\}$, we applied a set of randomly generated affine transformations $\{\boldsymbol{\phi}_i\}$ to $\{\boldsymbol{x}_i\}$, i.e. $\boldsymbol{y}_i(\boldsymbol{r}) = (\boldsymbol{x}_i \circ \boldsymbol{\phi}_i)(\boldsymbol{r})$ and $\boldsymbol{x}_i(\boldsymbol{r}) = (\mathbf{y}_i \circ \boldsymbol{\phi}_i^{-1})(\boldsymbol{r})$. The affine transformation matrix $\boldsymbol{\phi}_i$ is generated by randomly sampling from a normal distribution,

$$\phi_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \mathcal{N}_{2\times 3}(0, \sigma), \tag{5}$$

where $\mathcal{N}_{2\times 3}(0, \sigma)$ is a 2×3 matrix with each entry sampled from $\mathcal{N}(0, \sigma)$. We chose $\sigma = 0.05$ for this paper. During training and testing, we added uniform random noise between 0 and 10 to each pixel.

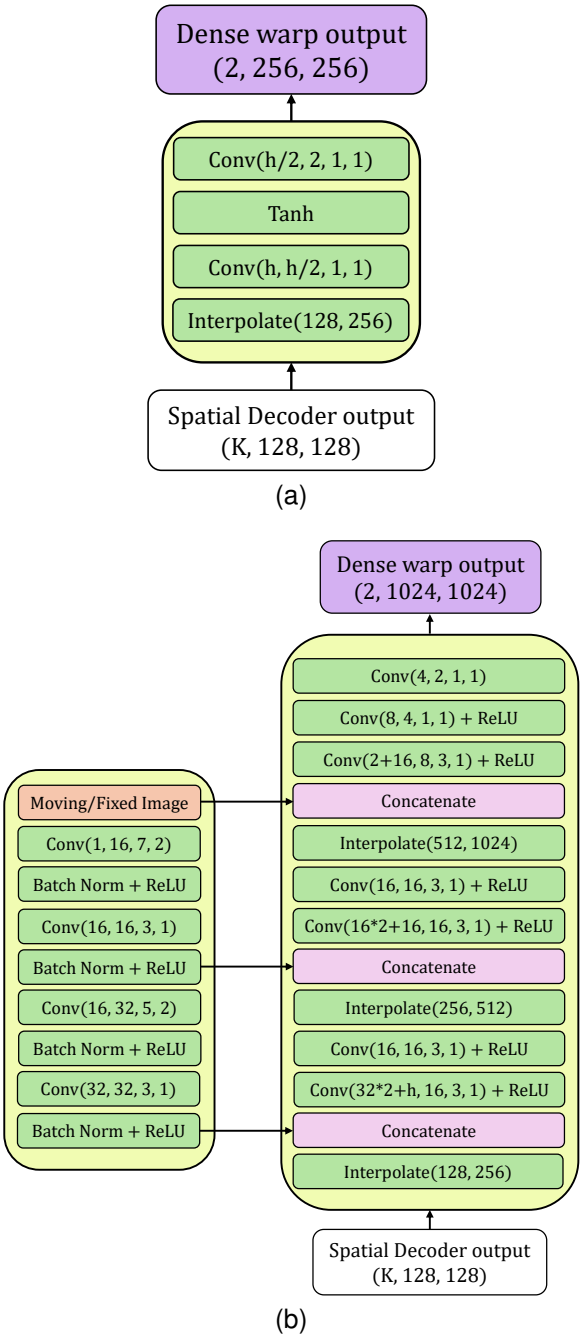

Fig. 7. Transformation decoder architecture. (a) 256×256 scale architecture. (b) 1024×1024 scale architecture.

B. Training

For training we used the standard L2 norm of the difference between the output warping and the ground truth warpings,

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\phi}) = \left(\left\| \hat{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y}) - \boldsymbol{\phi} \right\|_2 + \left\| \hat{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{x}) - \boldsymbol{\phi}^{-1} \right\|_2 \right) + \lambda \left(\left\| \nabla \cdot \hat{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y}) \right\|_1 + \left\| \nabla \cdot \hat{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{x}) \right\|_1 \right), \quad (6)$$

where $\hat{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y})$ is the output registration of the network given moving image $\mathbf{x}$ and fixed image $\mathbf{y}$, $\hat{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{x})$ is the output registration of the network given moving image $\mathbf{y}$ and fixed image $\mathbf{x}$, and λ is a hyperparameter for regularization.

TABLE I
REGISTRATION RESULTS

Similarity Metric	Coarse Network 256×256	Fine Network 1024×1024	SAR-SIFT 1024×1024
L_2 Error	1.702	6.045	4.986
DICE	0.931	0.736	0.744
LNCC	0.237	0.030	0.009
Inference (CPU)	0.7 s	1.6 s	41.2 s

Training was done in three steps using the PyTorch library [38]. We first trained the coarse registration auto-encoder separately, followed by training the entire network with the coarse registration auto-encoder parameters fixed, and finally concluded training with all weights and biases (aside from the Haar wavelet kernels mentioned earlier) updating during training. When the learned weights and biases were first initialized by randomly sampling from $\mathcal{N}(0, 1)$, the models would often converge to a minima far from the expected global minima, so we instead initialized the learned parameters by sampling from $\mathcal{N}(0, 0.1)$), which causes the model to start with a transformation much closer to the identity transformation (see Eq. 4c). All training was done with the Adam [39] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

We used 20000 pairs of 256×256 images ranging from 2m to 4m in resolution to train the coarse registration auto-encoder and reserved 4000 for testing. The batch size was 10, the learning rate was 1×10^{-3} (1×10^{-4} for fine-tuning), and $\lambda = 2$.

For the last two steps of training, we used 1500 pairs of 1024×1024 images ranging from 0.5m to 1m in resolution for training, 300 pairs for testing, a batch size of 2, a learning rate of 1×10^{-3} (1×10^{-4} for fine-tuning), and $\lambda = 2$.

V. RESULTS

We describe the results for both the coarse registration and the end-to-end network below. Since we have the ground truth registrations, it is most natural to use the L_2 norm error between the network's output registration and the ground truth registration. For tasks such as interferometry, which require accurate registration throughout the entire image, this is an ideal metric. For other tasks such as target recognition or multi-pass collection, similarity metrics like the continuous-valued DICE score [40] or a variant of the local normalized cross-correlation (LNCC) would be more appropriate, since a unique deformation is not required between large feature-less regions. Results for all three aforementioned similarity metrics are shown in Table I. For the LNCC, we used a window size of 19 pixels without any further pre-processing. We compare the network against a SAR-SIFT registration method applied to the same testing image pairs, using a thin plate spline (TPS) to turn matching feature points into a dense warping [41].

A. Coarse Registration

Testing examples from the coarse registration auto-encoder are presented in Fig. 8. The network achieved an average pixel displacement L_2 norm error of roughly 1.7 pixels, but visual inspection shows precise alignment of borders and edges,

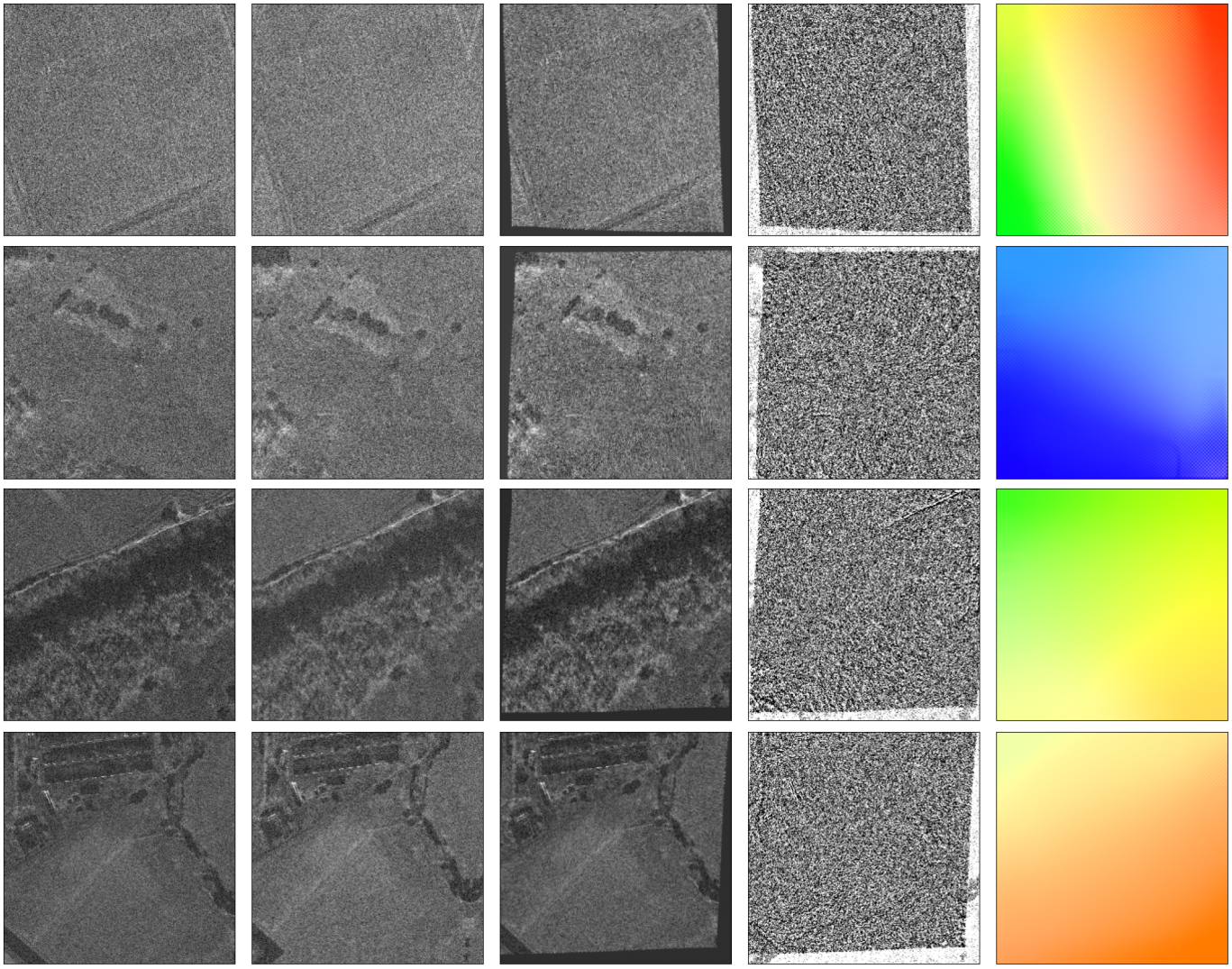

Fig. 8. Results of coarse registration network on 256×256 sub-images from the large testing images (see Fig. 3). The average pixel displacement L_2 norm error is $\sim$1.7. From left to right: moving image, fixed image, registered image, difference between registered image and fixed image, standard optical-flow visualization of spatial warping [42].

which is in agreement with the high DICE score. As mentioned previously, the contribution to the error is dominated by pixels located in feature-less areas of the image, which is penalized less in the DICE score.

B. Fine Registration

Testing examples from the fine registration auto-encoder are presented in Fig. 9. The network achieved an average pixel displacement L_2 norm error of roughly 6.0 pixels, but upon visual inspection and comparison of similarity metrics, it is clearly seen that the higher resolution and speckle noise lead to less precise alignment of borders and edges. We discuss possible improvements to the network's fine registration auto-encoder in VI.

VI. Discussion

As mentioned in section V, the presented architecture sacrifices precision (but not accuracy) and places a stronger emphasis on model size and inference time when handing larger SAR images. SAR-SIFT slightly outperforms the presented model at the 1024×1024 scale in all metrics aside from LNCC, but does so with a $\sim$ 26 times slower inference time on average. We believe most of the error and inefficiency of the fine registration auto-encoder originates from the spatial decoding network using global coordinate domain projections instead of local coordinate domain projections within the decoder (see Eq. 4b and Fig. 6). Given the encoder output, the simplest architecture would be an analogously structured decoder (i.e. local), similar in structure to the decoder in [33], that is convolved across the grid indices to determine the warping function for the (non-overlapping) corresponding image pixels. This would be a more natural approach, and would better mirror the structure of the phase-correlation encoder/decoder example in section II, but it requires continuity across image pixels in neighboring grid indices, and therefore requires coupling between grid indices, which is a necessary deviation from the phase-correlation architecture. We considered this approach with a separate architecture in which the fine registration

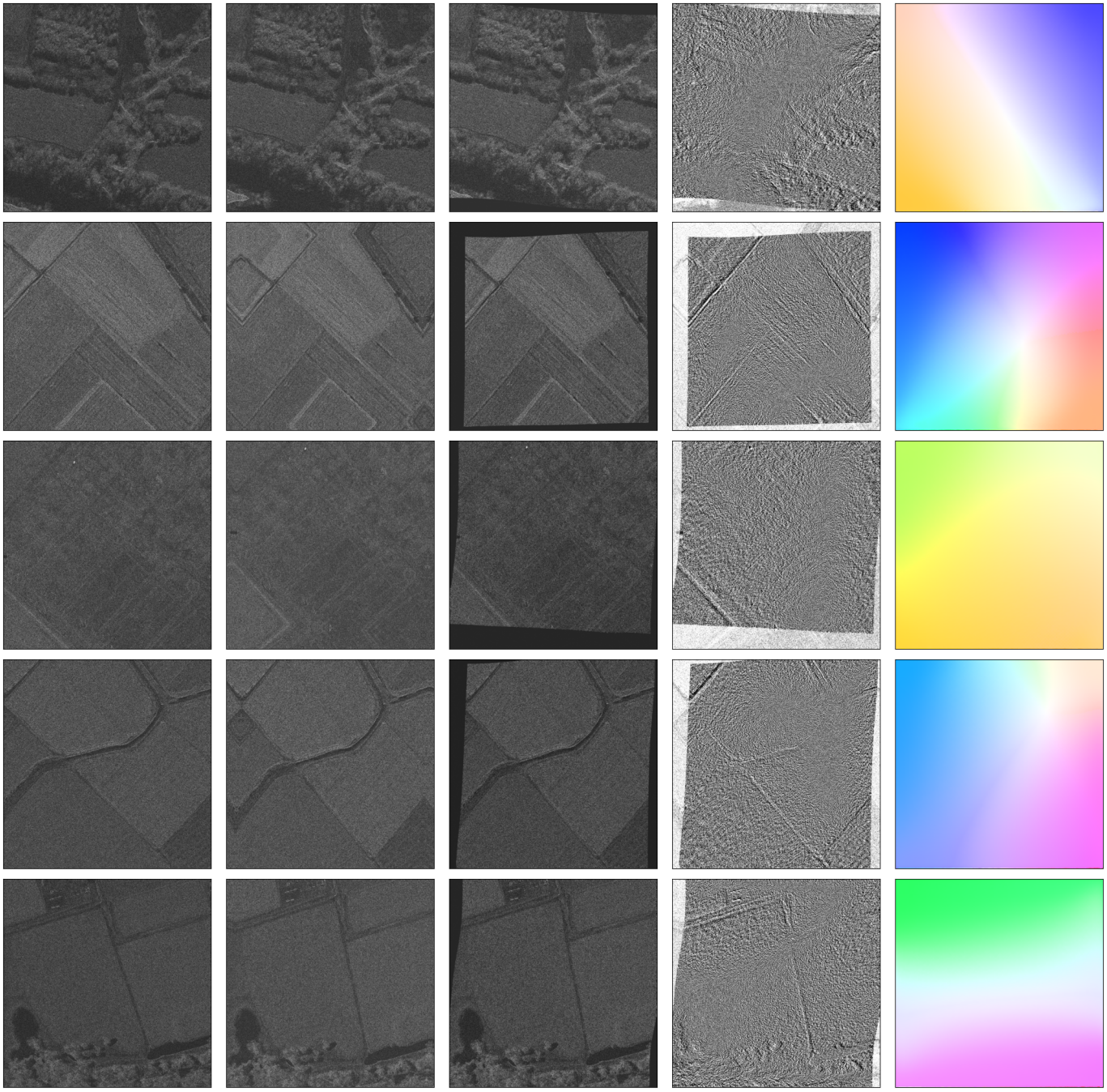

Fig. 9. Results of end-to-end registration network on 1024×1024 sub-images from the large testing images (see Fig. 3). The average pixel displacement L2 norm error is From left to right: moving image, fixed image, registered image, difference between registered image and fixed image, standard optical-flow visualization of spatial warping [42].

auto-encoder is replaced by non-overlapping convolutions of a second coarse registration auto-encoder. The results of that architecture are not presented here, since it performed worse on average than naively up-sampling the coarse registration warping, but we remain optimistic since there were no visible discontinuities in the warpings across neighboring grid indices, and we reserve this architecture and analysis thereof for a future paper.

VII. Conclusion

To conclude, this paper highlights the effectiveness of encoding relations between image pairs instead of independent image features, and presents a architecture capable of registering unseen SAR images of known environments with very little training data. Despite this success, several challenges remain - Finite spatial kernel sizes in the encoder place an upper bound on the maximum displacement between image pairs when encoding relations. Moreover, finite embedding dimensions in the encoder place an upper bound on the

number of spatial transformations learned by the network. This can be problematic for image pairs with very non-linear spatial transformations. Lastly, the general architecture of the presented decoder cannot effectively scale to large images without localized coordinate projections. This was addressed in section VI, but we consider it to be the greatest bottleneck of the presented architecture. Future work will focus on this bottleneck, deviating from the naive upscale + U-Net architecture in favor of a single-pass scalable decoder capable of handing larger sized SAR images in a more "natural" way.

REFERENCES

[1] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.

[2] X. X. Zhu, S. Montazeri, M. Ali, Y. Hua, Y. Wang, L. Mou, Y. Shi, F. Xu, and R. Bamler, "Deep learning meets sar: Concepts, models, pitfalls, and perspectives," *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 4, pp. 143–172, 2021.

[3] C. V. Jakowatz, D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*. Kluwer Academic Publishers, 1996.

[4] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M.-A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H.-C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015.

[5] X. Zhuang, L. Li, C. Payer, D. Stern, M. Urschler, M. P. Heinrich, J. Oster, C. Wang, O. Smedby, C. Bian, X. Yang, P.-A. Heng, A. Mortazi, U. Bagci, G. Yang, C. Sun, G. Galisot, J.-Y. Ramel, T. Brouard, Q. Tong, W. Si, X. Liao, G. Zeng, Z. Shi, G. Zheng, C. Wang, T. MacGillivray, D. Newby, K. Rhode, S. Ourselin, R. Mohiaddin, J. Keegan, D. Firmin, and G. Yang, "Evaluation of algorithms for multi-modality whole heart segmentation: An open-access grand challenge," 2019. [Online]. Available: https://arxiv.org/abs/1902.07880

[6] P. Bilic, P. Christ, E. Vorontsov, G. Chlebus, A. Ben-Cohen, S. Hashoul, N. Helbawi, P. Vološin, C. Saillard, M. Ferreira *et al.*, "The liver tumor segmentation benchmark (lits)," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 11765, 2019, pp. 547–555.

[7] N. Heller, N. Sathianathen, A. Kalapara, E. Walczak, K. Moore, H. Kaluzniak, J. Rosenberg, P. Blake, Z. Rengel, M. Oestreich, J. Dean, M. Tradewell, A. Shah, R. Tejpaul, Z. Edgerton, M. Peterson, S. Raza, S. Regmi, N. Papanikolopoulos, and C. Weight, "The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes," 2020. [Online]. Available: https://arxiv.org/abs/1904.00445

[8] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, and J. M. Stuart, "The cancer genome atlas pan-cancer analysis project," *Nature Genetics*, vol. 45, no. 10, pp. 1113–1120, 2013.

[9] F. Dellinger, J. Delon, Y. Gousseau, J. Michel, and F. Tupin, "Sar-sift: A sift-like algorithm for sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 1, pp. 453–466, 2015.

[10] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885603001379

[11] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.

[12] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2626–2635.

[13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[14] G. Wolberg, "Digital image warping," *IEEE Computer Graphics and Applications*, vol. 10, no. 2, pp. 30–44, 1990.

[15] J. Ping, Y. Liu, and D. Weng, "Comparison in depth perception between virtual reality and augmented reality systems," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2019, pp. 1124–1125.

[16] S. Thanyadit, P. Punpongsanon, and T.-C. Pong, "Investigating visualization techniques for observing a group of virtual reality users using augmented reality," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2019, pp. 1189–1190.

[17] A. Genay, A. Lécuyer, and M. Hachet, "Being an avatar "for real": A survey on virtual embodiment in augmented reality," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 12, pp. 5071–5090, Dec 2022.

[18] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *Advances in Neural Information Processing Systems (NIPS)*, pp. 2017–2025, 2015.

[19] S. K. Sønderby, C. K. Sønderby, L. Maaløe, and O. Winther, "Recurrent spatial transformer networks," 2015. [Online]. Available: https://arxiv.org/abs/1509.05329

[20] C.-Y. Lin, E. Yumer, O. Wang, E. Shechtman, and R. Szeliski, "Spatial transformer networks for image generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[21] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[22] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum, *End-to-End Unsupervised Deformable Image Registration with a Convolutional Neural Network*. Springer International Publishing, 2017, p. 204–212. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-67558-9_24

[23] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "Voxelmorph: A learning framework for deformable medical image registration," *IEEE Transactions on Medical Imaging*, vol. 38, no. 9, pp. 2048–2058, 2019.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.

[25] J. Chen, E. C. Frey, Y. He, W. P. Segars, Y. Li, and Y. Du, "Transmorph: Transformer for unsupervised medical image registration," *Medical Image Analysis*, vol. 82, p. 102615, 2022.

[26] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.

[27] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3243–3251.

[28] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 502–518.

[29] Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, H. Qin, J. Dai, and H. Li, "Flowformer: A transformer architecture for optical flow," 2022. [Online]. Available: https://arxiv.org/abs/2203.16194

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[32] R. Memisevic and G. Hinton, "Unsupervised learning of image transformations," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[33] ——, "Learning to represent spatial transformations with factored higher-order boltzmann machines," *Neural Computation*, vol. 22, no. 6, pp. 1473–1492, 2010.

[34] R. Memisevic, "Gradient-based learning of higher-order image features," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. USA: IEEE Computer Society, 2011, p. 1591–1598. [Online]. Available: https://doi.org/10.1109/ICCV.2011.6126419

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5998–6008.
[36] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 073–10 082.
[37] Umbra, "Umbra open data catalog," 2023, accessed: 9/10/2025. [Online]. Available: http://umbra-open-data-catalog.s3-website.us-west-2.amazonaws.com/
[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Chinton, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," https://pytorch.org/, 2019.
[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: https://arxiv.org/abs/1412.6980
[40] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.
[41] A. Goshtasby, "Registration of images with geometric distortions," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 26, no. 1, pp. 60–64, 1988.
[42] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. Lewis, and R. Szeliski, "A database and evaluation methodology for optical flow," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

Cole Reynolds received the PhD degree in Physics from the Institute for Quantum Sciences and Engineering at Texas A&M University, TX, USA. He is currently a Senior Research Scientist at Lockheed Martin's Advanced Development Program. His research interests include machine learning, computer vision, and electrodynamics.

Brandon Boucher is pursuing a PhD degree in Electrical and Computer Engineering at Georgia Institute of Technology, GA, USA. He is currently a Senior Research Scientist at Lockheed Martin's Advanced Development Program. His research interests include compressed sensing, AI/ML, and SAR/ISAR image processing techniques.

John Lusk received the Master's degree in Electrical Engineering from Texas Tech University, TX, USA. He is currently an Engineering Fellow within the Advanced Development Programs at Lockheed Martin Aeronautics. His research interests include computer vision, compressed sensing, and SAR/ISAR processing techniques

W. Bryan Bell received the PhD degree in Electrical Engineering from The University of Texas at Arlington, TX, USA. He is currently a Senior Technology Fellow within the Advanced Development Programs at Lockheed Martin Aeronautics. His research interest include signal and image processing, compressed sensing, and SAR/ISAR processing techniques.