Multi-Task Spatiotemporal Neural Networks for Structured Surface Reconstruction

Mingze Xu\textsuperscript{1} Chenyou Fan\textsuperscript{1} John D. Paden\textsuperscript{2} Geoffrey C. Fox\textsuperscript{1} David J. Crandall\textsuperscript{1}
\textsuperscript{1}School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN 47408
\textsuperscript{2}Center for Remote Sensing of Ice Sheets, University of Kansas, Lawrence, KS USA 66045
\{mx6, djcran\}@indiana.edu, paden@ku.edu

Abstract

Deep learning-based methods have surpassed more traditional techniques on a wide range of problems, but nearly all of this work has studied consumer photos, where precisely correct output is often not critical. It is less clear how well these techniques may work on structured prediction problems where finer-grained output with high precision is required, such as in scientific imaging domains. Here we consider the problem of segmenting echogram radar data collected from the polar ice sheets, which is challenging because segmentation boundaries are often very weak, and there is a high degree of noise. We propose a multi-task spatiotemporal neural network that combines 3D ConvNets and Recurrent Neural Networks (RNNs) to estimate ice surface boundaries from hundreds of sequences of tomographic slices from radar, showing that: (a) 3D ConvNets and RNNs are able to generate accurate well-structured outputs without handmade pairwise potentials; (b) our model can extract ice-air and ice-bed surfaces simultaneously; and (c) our model outperforms the state-of-the-art on this problem but uses less non-visual evidence and is about 6 times faster.

1. Introduction

Three-dimensional imaging is widely used in scientific research domains (e.g., biology, geology, medicine, and astronomy) to measure the structure of objects and how they change over time. Although the exact techniques differ depending on the problem and materials involved, the common idea is that electromagnetic waves (e.g., X-ray, radar, etc.) are sent into an object, and signal returns collected in the form of sequences of tomographic images are then analyzed to estimate the object’s 3D structure. However, analysis of these image sequences can be difficult even for humans, since they are often full of noise and require integrating evidence from multiple sources simultaneously.

As a particular example, an important part of modeling and forecasting the effects of global climate change is to understand polar ice. Hidden beneath the ice of the poles is a rich and complex structure: the ice consists of multiple layers that have accumulated over many thousands of years, and the base is bedrock that has a complicated topography just like any other place on Earth, with mountains, valleys, and other features. Moreover, the ice sheets move over time, and their movement is determined by a variety of factors, including temperature changes, flows underneath the surface, and the topography of the bedrock below and nearby. Accurately estimating all of this rich structure is crucial for understanding how ice will change over time, which in turn is important for predicting the effects of melting ice associated with climate change.

Glaciologists traditionally had to drill ice cores to probe the subsurface structure of polar ice, but advances in
ground-penetrating radar technology have revolutionized this data collection process. But while these radar observations can now be collected over very large areas, actually analyzing the radar data to determine the structure of subsurface ice is typically done by hand [24]. This is because the radar echograms produced by the data collection process are very noisy: changes in atmospheric pressure, ice composition, temperature, etc. affect radar signal returns in complex ways. Relying on humans to interpret data not only limits the rate at which datasets can be processed, but also limits the type of analysis that can be performed: while a human expert can readily mark ice sheet boundaries in a single 2D radar echogram, doing this simultaneously over thousands of echograms to produce a 3D model of an ice bed, for example, is simply not feasible.

While several recent papers have proposed automated techniques for segmenting layer boundaries in ice [5, 8, 12, 13, 17, 23, 25, 26, 35], none have approached the accuracy of even a quickly-trained undergraduate student annotator [24], much less an expert. However, these techniques have all relied on traditional image processing and computer vision techniques, like edge detection, pixel template models, active contour models, etc. Most of these techniques also rely on numerous parameters and thresholds that must be tuned by hand. Some recent work reduces the number of free parameters through graphical models that explicitly model noise and uncertainty [8, 23, 26, 35], but still rely on simple features.

In this paper, we apply deep networks to the problem of ice boundary reconstruction in polar radar data. Deep networks have become the de facto standard technique across a wide range of vision problems, including pixel labeling problems. The majority of these successes have been on consumer-style images, where there is substantial tolerance for incorrect predictions. In contrast, for problems involving scientific datasets like ice layer finding, there is typically only one “correct” answer, and it is important that the algorithm’s output is as accurate as possible.

Here we propose a technique for combining 3D convolutions and Recurrent Neural Networks (RNNs) to perform segmentation in 3D, borrowing techniques usually used for video analysis to instead characterize sequences of tomographic slice images. In particular, since small pixel value changes only affect a few adjacent images, we apply 3D convolutional neural networks to efficiently capture cross-slice features. We extract these spatial and temporal features for small neighborhoods of slices, and then apply an RNN for detailed structure labeling across the entire 2D image. Finally, layers from multiple images are concatenated to generate a 3D surface estimate. We test our model on extracting 3D ice subsurfaces from sequences of radar topographic images, and achieve the state-of-the-art results in both accuracy and speed.

2. Related Work

A number of methods have been developed for detecting layers or surfaces of material boundaries from sequential noisy radar images. For example, in echograms from Mars, Freeman et al. [13] find layer boundaries by applying band-pass filters and thresholds to find linear subsurface structures, while Ferro and Bruzzone [11] identify subterranean features using iterative region-growing. Crandall et al. [8] detect the ice-air and ice-bed layers in individual radar echograms by combining a pre-trained template model and a smoothness prior in a probabilistic graphical model. In order to achieve more accurate and efficient results, Lee et al. [23] utilize Gibbs sampling from a joint distribution over all candidate layers, while Carrer and Bruzzone [5] reduce the computational complexity with a divide-and-conquer strategy. Xu et al. [35] extend the work to the 3D domain to reconstruct 3D subsurfaces using a Markov Random Field (MRF).

In contrast, we are not aware of any work that has studied this application using deep neural networks. In the case of segmenting single radar echograms, perhaps the closest analogue is segmentation in consumer images [32]. Most of this work differs from the segmentation problem we consider here, however, because our data is much noisier, our “objects” are much harder to characterize (e.g., two layers of ice look virtually identical except for some subtle changes in texture or intensity), our labeling problem has greater structure, and our tolerance for errors in the output is lower.

For segmenting 3D regions, perhaps the closest related work is in deep networks for video analysis, where the frames of video can be viewed as similar to our tomographic slices. Papers that apply deep networks to video applications focus on efficient ways to combine spatial and temporal information, and can be roughly categorized into three classes: (1) combining both RGB frames for spatial features and optical flow images for temporal features in two-stream networks [29], (2) explicitly learning 3D spatiotemporal filters on image spaces through techniques such as C3D [31], and (3) various combinations of both [4]. In order to obtain video representations from per-frame or per-video-segment features, it is a common practice to apply temporal pooling to abstract into fixed-length per-video features [20, 29]. These approaches achieve significantly better classification accuracy on video classification compared to traditional approaches of using hand-crafted features.

Recurrent Neural Networks (RNNs) and the specific version we consider here – Gated Recurrent Units (GRUs) – have been proposed for learning sequential data, such as natural language sentences [10, 14], programming language syntax [19], and video frames [37]. A popular application of RNNs recently [18, 33] is to generate image captions in combination with CNNs. In this case, CNNs are
used to recognize image content while RNNs are used as language models to generate new sentences. Video frames are also naturally sequential data as adjacent frames share similar content while differences reveal motion and other changes over time. A large variety of studies \cite{9, 27, 37} share the common idea of applying RNNs on deep features for each video frame and pooling or summing over each as the final video feature. Other successful applications of RNNs to interesting vision and natural language tasks include recognizing multiple objects by making guided glimpses in different parts of images \cite{3}, answering visual questions \cite{2, 22, 34}, generating new images with variations \cite{15, 36}, reading lips \cite{7}, etc.

We build on this existing work but apply to the novel domain of extracting and reconstructing structured 3D surfaces from sequential data by combining C3D and RNN networks. In particular, we use the C3D network as a robust feature extractor to capture local-scale within-slice and between-slice features in 3D space, and use the RNN to capture longer-range structure both within single slices and across the entire image sequence.

3. Technical Approach

Three-dimensional imaging typically involves sending electromagnetic radiation (e.g., radar, X-ray, etc.) into a region and collecting a sequence of cross-sectional tomographic slices $I = \{I_1, I_2, \cdots, I_D\}$ that characterize returned signals along the searching path. Each slice $I_d$ is a 2D tomographic image of size $H \times W$ pixels. In the particular case of ice segmentation, we are interested in locating $K$ layer surface boundaries between different materials. Our output surfaces are highly structured, since there should be exactly $K$ surface pixels within any column of a given tomographic image. We thus need to estimate the layer boundaries in each individual slice, while incorporating evidence from all slices jointly in order to overcome noise and resolve ambiguities. Layer boundaries within each slice can then be concatenated across slices to produce a 3D surface.

In this section, we describe the two important components of our network framework: our multi-task 3D Convolutional (C3D) Network that captures within-slice features as well as evidence from nearby slices, and our Recurrent Neural Network (RNN) which incorporates longer-range cross-slice constraints. The overall architecture is shown in Figure 2.

\section{3.1. A Multi-task C3D Architecture}

Traditional convolutional networks for tasks like object classification and recognition lack the ability to model spatiotemporal features in 3D space. More importantly, their use of max or average pooling operations makes it impractical to preserve temporal information within the sequential inputs. To address these problems, we use C3D networks that are adept at modeling the spatiotemporal features required for our task.
to capture local spatiotemporal features in our sequence of input images. C3D has typically been used for video, but our dataset has very similar characteristics: we have a sequence of tomographic slices taken in consecutive (discrete) positions along the path of a penetrating wave source (a moving plane, in the case of our ice application). Physical constraints on layer boundaries (e.g., that they should be continuous and generally smooth) mean that integrating information across adjacent images improves accuracy, especially when data within any given slice is particularly noisy or weak.

Figure 3 illustrates details of our C3D architecture, which is based on [31] but with several important modifications. Since the features of these structured layers in tomographic images are typically less complicated than consumer photos, we use a simpler network architecture, as follows. For the input, our model takes \( L \) consecutive images, where \( L \) is a small odd number; we have tried \( L = 1, 3, 5, \ldots, 11 \), and choose 5 as the best empirical balance between running time and accuracy. Then, we use two shared convolutional layers, each of which is followed by rectifier (ReLU) units and max pooling operations, to extract low-level features for all layers. The key idea is that different kinds of layer boundaries usually share similar detailed patterns, although they have individual high-level features, e.g., shapes. Inspired by the template model used in [8, 35], our model uses rectangular convolutional filters with a size of \( 3 \times 5 \times 3 \), since the important features lie along the vertical dimension. Afterwards, the framework is divided into \( K \) branches, each with 6 convolutional layers for modeling features specific to each type of ice layer boundary. The filter size is the same as with the shared layers. Two fully-connected layers are appended to the network for each ice layer, where the \( k \)-th ice layer has \( W \) outputs \( S_d^k = \{ s_{d,1}^k, s_{d,2}^k, \ldots, s_{d,W}^k \} \), each corresponding to a column of the tomographic slice \( I_d \), representing the row coordinate of the \( k \)-th ice layer boundary within that column. All training images have been labeled with ground truth vectors, \( G_d^k = \{ g_{d,1}^k, g_{d,2}^k, \ldots, g_{d,W}^k \} \) to indicate the correct position of these output layers in each image.

We train the C3D network using the L2 Euclidean loss \( L_{elu} \) to encourage the model to predict correct labelings according to human-labeled ground truth,

\[
L_{elu} = \frac{1}{2} \sum_{k=1}^{K} \sum_{w=1}^{W} (s_{d,w}^k - g_{d,w}^k)^2.
\]

We note that this formulation differs from what is typically done with semantic and instance segmentation, which typically use Softmax and Cross-entropy as the target function. This is because we are not assigning each pixel with a categorical label (e.g., dog, cat, etc.), but instead assigning each column of the image with a row index. Since these labels are ordinal, it makes sense to directly compare them and minimize a Euclidean loss.

### 3.2. A Multi-task RNN Architecture

The C3D networks discussed above model features both in the temporal and spatial dimensions, but only in very small neighborhoods. For example, they can model the fact that adjacent pixels within the same layer should have similar grayscale value, but not that the layer boundaries themselves (which are usually separated by dozens of pixels at least) are often roughly parallel to one another. Similarly, CSD models some cross-slice constraints but only in a few slices in either direction. We thus also include an RNN that incorporates longer-range cross-slice evidence. Because of the limited training data, we use Gated Recurrent Units
(GRUs) [6] since they have fewer learnable parameters than another popular networks like Long Short-Term Memories (LSTMs) [16].

**GRU Training and Testing.** The multi-task GRU framework is shown in Figure 2. Our model for each individual slice consists of $K$ GRU cells, each responsible for predicting the $k$-th layer in each image. Each GRU cell takes a tomographic slice $I_d$ and the output of the previous GRU layer as inputs, and produces $W$ real value numbers indicating the predicted positions of the layer within each column of the image. Each GRU also takes as input the output from the GRU corresponding to the same ice layer in the previous slice, since these layer boundaries should be continuous and roughly smooth. In previous work [8, 23, 35], this prior knowledge was explicitly enforced by pairwise interaction potentials, which were manually tuned by human experts. Here we train RNNs to be able to model more general relationships in an fully learnable way.

We split each tomographic input image $I_d$ into separate columns vectors $I_{d,w}$, $w = 1, 2, \cdots, W$, each with width 1 and height $H$. Each column vector is projected to the length of the GRU hidden state with a fully-connected layer. During training time, the $k$-th GRU cell is operated for $W$ iterations, where each iteration $w$ predicts the $k$-th layer position in image column $I_{d,w}$. Then in a given iteration $w$, the $k$-th GRU takes the fused features (e.g., using sum or max fusion) of the (resized) image column $I_{d,w}$ and the hidden state $h_{d,w}^{k-1}$ as the input. It also receives the hidden states $h_{d,w-1}^k$ of itself in iteration $w-1$ as contextual information. More formally, the $k$-th GRU cell outputs a sequence of hidden state $h_{d,1}^k, h_{d,2}^k, \cdots, h_{d,W}^k$ with iteration $w = 1, 2, \cdots, W$, and each hidden state $h_{d,w}^k$ is followed by a fully-connected layer to predict the actual layer position $s_{d,w}^k$ as shown in Figure 4. Since each GRU has the same operation for each 2D image $I_d$, we drop $d$ subscribe for simplicity, and compute,

\[
\begin{align*}
    z_w &= \text{sigmoid}(U_{iz} F(I_w, h_{w-1}^k) + U_{hz} h_{w-1}^k + b_z), \\
    r_w &= \text{sigmoid}(U_{ir} F(I_w, h_{w-1}^k) + U_{hr} h_{w-1}^k + b_r), \\
    n_w &= \tanh(U_{in} F(I_w, h_{w-1}^k) + U_{hn} (r_w \circ h_{w-1}^k) + b_n), \\
    h_w &= z_w \circ h_{w-1}^k + (1 - z_w) \circ n_w, \quad \text{and} \\
    s_w &= U_{y} h_w + b_y,
\end{align*}
\]

where $\circ$ is the Hadamard product, $z_w$, $r_w$, $n_w$, $h_w$, and $s_w$ are the reset, input, new gate, hidden state, and output layer position at time $w$, respectively. We use 512 neurons in the hidden layer of the GRU. We train the GRU network with the same L2 Euclidean loss $L_{elu}$ as discussed in the previous section.

### 3.3. Combination

We combine our proposed C3D model and GRU model for efficiently encoding spatiotemporal information into explicit structured layer predictions. We use the C3D features $C3D_d^b(I_{d,k})$ (where $C3D^b_d$ denotes the features with model parameters $\theta$ for the $k$-th ice layer) to initialize the $k$-th GRU’s hidden state $h_1$, as shown in Figure 2. In the figure, $I_d$ is marked in red; this is the frame currently under consideration, which is divided into columns which are then provided to the GRU cells one at a time.
Figure 6. Results of the extracted ice-air surfaces based on about 330 tomographic images. The x-axis corresponds to distance along the flight path, and the y axis is the width of the tomographic images ($W$), and the color is the height dimension (max height is $H$), which also represents the depth from the radar.

<table>
<thead>
<tr>
<th>Mean Error</th>
<th>Ice-air surface</th>
<th>Ice-bed surface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crandall [8]</td>
<td>-</td>
<td>101.6</td>
</tr>
<tr>
<td>Lee [23]</td>
<td>-</td>
<td>35.6</td>
</tr>
<tr>
<td>Xu et al. (without ice masks) [35]</td>
<td>-</td>
<td>30.7</td>
</tr>
<tr>
<td>Xu et al. [35]</td>
<td>-</td>
<td>11.9</td>
</tr>
<tr>
<td>Ours (RNN)</td>
<td>10.1</td>
<td>21.4</td>
</tr>
<tr>
<td>Ours (C2D)</td>
<td>8.8</td>
<td>15.2</td>
</tr>
<tr>
<td>Ours (C3D)</td>
<td>9.4</td>
<td>13.9</td>
</tr>
<tr>
<td>Ours (C2D + RNN)</td>
<td>8.4</td>
<td>14.3</td>
</tr>
<tr>
<td>Ours (C3D + RNN)</td>
<td>8.1</td>
<td>13.1</td>
</tr>
</tbody>
</table>

Table 1. Error in terms of the mean absolute column-wise difference compared to ground truth, in pixels.

<table>
<thead>
<tr>
<th>Averaged Mean Error (pixels)</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xu et al. [35]</td>
<td>11.9</td>
</tr>
<tr>
<td>Ours (C3D + RNN)</td>
<td>10.6</td>
</tr>
</tbody>
</table>

Table 2. Performance evaluation comparing to the state of the art. The accuracy is computed on the average of the ice-air and ice-bed surfaces. The running time is measured by processing a sequence of 330 tomographic images.

4. Experiments

4.1. Dataset

We use a dataset of the basal topography of the Canadian Arctic Archipelago (CAA) ice sheets, collected by the Multichannel Coherent Radar Depth Sounder (MCoRDS) instrument [28]. It contains a total of 8 tomographic sequences, each with over 3,300 radar images corresponding to about 50km of flight data per sequence. For training and testing, we also have ground truth that identifies the positions of two layers of interest (the ice-air and ice-bed, i.e., $K = 2$). Several examples of these tomographic images and their annotations are shown in Figure 5.

To evaluate our model, we split the data into training and testing sets (60% as training images, 40% as testing images) and learn the model parameters from the training images. More formally, we wish to detect the ice-air and ice-bed layers in each image, then reconstruct their corresponding 3D surfaces from a sequence of tomographic slices. We assume the tomographic sequence has size $C \times D \times H \times W$, where $C$ denotes the number of image channels (which is 1 for our data), $D$ is the number of slices in the sequence, and $W$ and $H$ are the dimensions of each slice. We also parameterize the output surfaces as sequences, $S^k = \{S^k_1, S^k_2, \ldots, S^k_D\}$, and $S^d = \{s^k_{d,1}, s^k_{d,2}, \ldots, s^k_{d,W}\}$, where $s^k_{d,w}$ indicates the row coordinate of the surface position for column $w$ of slice $d$, and $s^k_{d,w} \in [1, H]$ since the boundary can occur anywhere within a column. In our case, $k \in \{0, 1\}$ to represent the ice-air and ice-bed surfaces, respectively.

Normalization. Since images from different sequences have different sizes (from $824 \times 64$ pixels to $2000 \times 64$ pixels), we resize all inputs images to $64 \times 64$ by using bicubic interpolation. For each image, we also normalize their pixels values to the interval $[-1, 1]$ and sub-
Figure 7. Sample results of extracted ice-bed surfaces from a sequence of about 330 tomographic images. The x-axis corresponds to distance along the flight path, and the y axis is the width of the tomographic images ($W$), and the color is the height dimension (max height is $H$), which also represents the depth from the radar.
We project the absolute image coordinates containing 5 consecutive radar images. The training process for evaluation. We repeat this training process (each time divided into 10 sub-sequences on average, and we randomly choose 60% of them as training data and the remaining 40% for evaluation. We repeat this training process (each time from scratch) three times and report the average statistics for evaluation.

For C3D training, we use the Adam [21] optimizer to learn the network parameters with batch size of 128, each containing 5 consecutive radar images. The training process is stopped after 20 epochs, starting with a learning rate of $10^{-4}$ and reducing it in half every 5 epochs. The RNN training is applied with the same update rule and batch size, but uses learning rate $10^{-3}$ multiplied by 0.1 every 10 epochs.

4.3. Evaluation

We evaluate our model on estimating the ice-air and ice-bed surfaces from tomographic sequences of noisy radar images. We run inference on the testing sub-sequences and calculate the pixel-level errors with respect to the human-labeled ground truth. We report the results with two summary statistics: mean deviation errors and running time. As shown in Table 2, the mean error averaged across the two different surfaces is about 10.6 pixels (where the mean ice-air surface error is 8.1 pixels and mean ice-bed surface error is 13.1 pixels), and the running time of processing a topographic sequence with 330 images is about 51.6 seconds. Figure 6 and 7 show some example results of the ice-air and ice-bed surfaces, respectively.

To give some context, we compare our results to previous state of the art techniques as baselines, and results are presented in Table 1. Our first two baselines are Cran-dall et al. [8] that detects the ice-air and ice-bed layers by incorporating a template model with vertical profile and a smoothness prior in a Hidden Markov Model, and Lee et al. [23] that use Markov-Chain Monte Carlo (MCMC) to sample from the joint distribution over all possible layers conditioned on radar images. These techniques were designed for 2D echogram segmentation and do not include cross-slice constraints, so they perform poorly on this problem. Xu et al. [35] does use information between adjacent images and achieves slightly better results than our technique (11.9 vs 13.1 mean pixel error), but that technique also uses more information. In particular, they incorporate additional extra non-visual metadata from external sources, such as the “ice mask” which gives prior weak information about anticipated ice thickness (e.g., derived from satellite maps or other prior data). When we removed the ice mask queue from their technique to make the comparison fair, our technique beat theirs by a significant margin (13.1 vs 30.7 mean pixel error). Our approach has two additional advantages: (2) it is able to jointly estimate both the ice-air and ice-bed surfaces simultaneously, so it can incorporate constraints on the similarity of these boundaries, and (2) it requires less than one minute to process an entire sequence of slices, instead of over 5 minutes for [35].

In addition to published methods, we also implemented several baselines to evaluate each component of our deep architecture. Specifically, we implemented: (a) a basic C2D network using the same architecture with the 3D network but with 2D convolution and pooling operations; (b) the RNN network using the extracted features from the C2D as the initial hidden state; (c) the C3D network alone without the RNN; (d) the RNN network alone without the C3D network. The results of these baselines are also shown in Table 1. The results show that all components of the model are important for achieving good performance, and that the best accuracy is achieved by our full model.

5. Conclusion

We have presented an effective and efficient framework for reconstructing smoothed and structured 3D surfaces from sequences of tomographic images using deep networks. Our approach shows significant improvements over existing techniques in three ways: (1) our model is able to extract and reconstruct different material boundaries, simultaneously; (2) our model only uses the tomographic sequences as input without any extra evidence from other instruments or human experts; (3) our model improves the feasibility of analyzing large-scale datasets by significant decreases in the running time. We also demonstrated the performance of our model by producing convincing ice-air and ice-bed surfaces results against the state-of-the-art.

References


