Liver Ultrasound Tracking using a Learned Distance Metric

Daniel Nouri and Alex Rothberg

4Catalyzer, Inc.,
Guilford CT 06437, USA,
{dnouri,arothberg}@4catalyzer.com

Abstract. We present a method for landmark tracking in long liver ultrasound sequences. We employ metric learning, and train a convolutional neural network to map from pixel intensities of grayscale ultrasound image patches into a low-dimensional embedding space such that patches showing the same landmark at their center have a small L2 distance in the embedding. We then locate landmarks throughout a sequence of ultrasound frames by extracting patches from a search window inside the target frame and finding the patch in the target frame that in the embedding space minimizes the distance to a number of template patches containing the landmark and extracted from previous frames. Our approach had a mean tracking error of 2.83mm, with 38 of 62 tracked points having an error of less than 1.5mm.

Keywords: ultrasound, tracking, medical imaging, learned distance metric, CLUST15

1 Introduction

Ultrasound (US) imaging is a widely used medical imaging technique due to its relatively low-cost components, fast acquisition speed, and safe, non-ionizing radiation. In addition, because it also offers high temporal resolution images in real-time, US is often used for tissue tracking during image-guided intervention and therapy.

Tracking the motion of tissue in an ultrasound sequence is complicated by respiratory motion, image noise, and the relatively long (often more one minute) acquisitions. Tracking is further complicated by large changes in shape of the tracking target, particularly when anatomical targets are not captured in plane. Long acquisitions are particularly difficult due to high likelihood of both patient and operator motion. In many cases, the US capture probe is handheld.

In this paper, we present a new tracking scheme based on a distance metric for US image patches that is learned from data. We use the learned distance metric to compare candidate square image patches with patches extracted from a target reference frame. The algorithm requires a training phase in which the distance metric is learned from raw pixel intensity values, for all device types

simultaneously. No further parameterization is needed when applying the algorithm to previously unseen data. We evaluated this new scheme participating in the MICCAI CLUST15 [2] liver ultrasound tracking challenge.

1.1 Related Work

Several systems that use deep neural networks to learn distance metrics have been proposed for applications such as face verification and signature verification. Training for verification instead of classification has the advantage that fewer labeled examples are needed, and that systems can naturally generalize to categories previously unseen [7,8,20].

The very similar CLUST14 challenge saw a wide range of proposed methods ranging from non-linear image registration, and long-term and short-term template matching, to Bayesian methods. None of the proposed methods incorporated deep neural networks as part of their solution [17]. However, the winner of the *MICCAI 2013 Grand Challenge on Mitosis Detection* was a system using a deep neural network at its core [10].

2 Materials and Methods

2.1 Ultrasound Data

2D B-mode ultrasound data was provided as part of the MICCAI 2015 Challenge on Liver Ultrasound Tracking (CLUST) [2]. The data were cine images of human livers. The data came from a number of patients and institutions (CIL, ETH, ICR, and MED datasets) and were acquired by one of five ultrasound systems (Ultrasonix MDP, Siemens Antares, Elekta Clarity - Ultrasonix, DiPhAs Fraunhofer and Zonare z.one). The data had varying spatial (0.28 – 0.55mm) and temporal resolution (11 – 23Hz) and sequences lasted from 59.4 – 328.6s. The number of annotations per image sequence ranged from one to five liver features.

24 of the 48 datasets, totaling 53 target annotations, were provided with ground truth annotations (of liver blood vessels) throughout the acquisition sequence. Approximately 10% of the frames had the locations of the tracking points annotated. A total of 62 points had to be tracked in the test-set where only the initial position of liver features (blood vessels centers) was given.

Annotations were provided in the following form: frame number, x-pixel (lateral position) and y-pixel (axial position).

2.2 Distance Metric Learning

Given a sequence of ultrasound images $I_{0...N}$, along with an annotated landmark L given by its position $c_0 \in \mathbb{R}^2$ in the first frame I_0, the problem is to locate the center positions $c_{1...N}$ of the given landmark in all subsequent frames $I_{1...n}$. We solve the problem by training a convolutional neural network (ConvNet)

that learns a function $G_W(p)$ to map ultrasound image patches p into to a low-dimensional space such that the distance metric

$$D_W(p_i, p_j) = ||G_W(p_i), G_W(p_j)||_2 \tag{1}$$

is small if p_i and p_j show the same landmark at their center and large otherwise.

The weights W of mapping function G_W are learned using stochastic gradient descent and the following loss function originally proposed in [14],

$$\mathcal{L}(p_i, p_j, s_{ij}, W) = \begin{cases} \frac{1}{2}\, D_W(p_i, p_j)^2 & \text{if } s_{ij} = 1, \\ \frac{1}{2}\, max(0, m - D_W(p_i, p_j))^2 & \text{if } s_{ij} = 0 \end{cases} \tag{2}$$

where $s_{ij} = 1$ for a pair of patches (p_i, p_j) that show the same landmark at their center and $s_{ij} = 0$ otherwise. m is a margin constant used to limit the penalty for dissimilar pairs; it was set to 0.1.

2.3 Training Data

For training, we form pairs of square patches (p_1, p_2) of the same landmark extracted from different frames using the ground-truth annotations. These are the pairs for which the distance $D_W(p_1, p_2)$ is learned to be small.

In addition, we form twice as many pairs of patches for training where p_1 contains the same landmark as p_2 but shifted away from the center by at least 4 and by at most 46 pixels in both dimensions uniformly at random. We also train with some pairs where both patches show different landmarks taken from the same sequence. These are the pairs for which the distance is learned to be big.

Because our mapping function G_W has many learnable parameters (1,865,278 in our best configuration), and thus tends to show high variance, we augment our training data by randomly flipping patches in both vertical and horizontal directions.

We pre-process all ultrasound image frames with a small-size median filter. All extracted patches are of size 46 x 46, which we determined empirically to be optimal. We use all available training data to learn the parameters of G_W.

2.4 Convolutional Neural Network Architecture

A ConvNet is a feed-forward neural network that uses successive pairs of convolutional and max-pooling layers, followed by fully connected layers. The input to our ConvNet is raw pixel intensities, the output is an embedding in low-dimensional space. All weights W of the network are learned from scratch using the contrastive loss function in (2). The weights are randomly initialized using Glorot initialization as described in [12]. Weights are updated during training using Nesterov's Accelerated Gradient [18]. We train a single network that learns the weights of G_W for all sequences and device types simultaneously. Table 1 lists the architecture of our ConvNet.

Our decision to use a ConvNet to implement G_W is motivated by the recent successes of using ConvNets in mitosis detection, and in computer vision tasks in general [10]. Through the use of learning curves in our experiments we've determined that our ConvNet is still well in the regime where using more training data (and possibly more aggressive data augmentation) leads to a linear increase in performance. Another intriguing property of ConvNets is that they can learn from raw pixels directly, and thus eliminate the often tedious task of engineering features and choosing dataset-specific parameters by hand. The max pooling layer calculates the max value of a particular feature over a region of the image. This ensures that the same result is obtained even when images features undergo small translations.

Table 1. 8-layer architecture of our ConvNet with a total of 1,865,278 learnable parameters. Layer type: I - input, C - convolutional, MP - max-pooling, MO - maxout [13], FC - fully-connected.

Layer	Type	Maps and Neurons	Filter size
0	I	1Mx46x46	—
1	C	32Mx46x46	5x5
2	MP	32Mx25x25	2x2
3	C	64Mx23x23	3x3
4	MP	64Mx12x12	2x2
5	FC	200	1x1
6	MO	50	4x1
7	FC	50	1x1

2.5 Template Patches

In a given ultrasound frame I_i, we predict the center $c_i \in \mathbb{R}^2$ position of the tracked landmark L by finding a target patch p that minimizes

$$D_W(p, t_0) + \frac{\sum_{k=i-K}^{i-1} D_W(p, t_k)}{K} \tag{3}$$

for $K + 1$ template patches t extracted from previous frames and showing landmark L. See Figure 1 for examples of the distance map created by the window search.

Template patch t_0 is extracted from the initial frame I_0 with its center position c_0 provided by the human annotation. Patches $t_{i-K} \dots t_{i-1}$ are extracted each from K previous frames $I_{i-K} \dots I_{i-1}$ with their center at the position of the tracking algorithm's previously predicted landmark position $y_i \in \mathbb{R}^2$. Thus, to be able to extract template patches for use in frame I_i, we must first predict the position y of L in frames $I_{i-K}, \dots, I_{i-1}$.

Through our experiments, we determined $K = 10$ to be the optimal number of template patches to use from previous predictions.

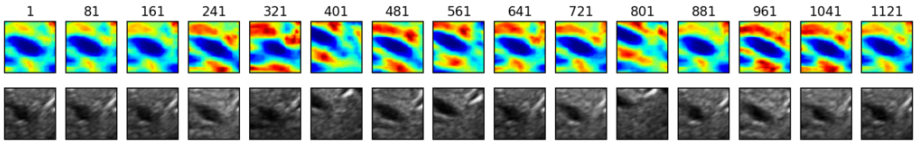

Fig. 1. Distance map created by the window search; the dark blue region represents the most similar match. Performed on the second point in `ICR-01` on frames $1, 81 \ldots 1121$.

2.6 Search Window

When looking for a patch p that has minimum distance to template patches t for landmark L as defined in Section 2.5, we only consider patches in the target frame I_i that have their center pixel within a defined square search window. This search window is searched through exhaustive search and itself centered at the predicted position y_{i-1} of landmark L in the previous frame I_{i-1}, or at the initial annotation c_0 for $i = 1$.

The predicted position y_i is defined by the center pixel of the patch p that's found to have minimum distance. We chose the width of the search window to be 24 pixels, which allows tracking to compensate for errors in previous frames.

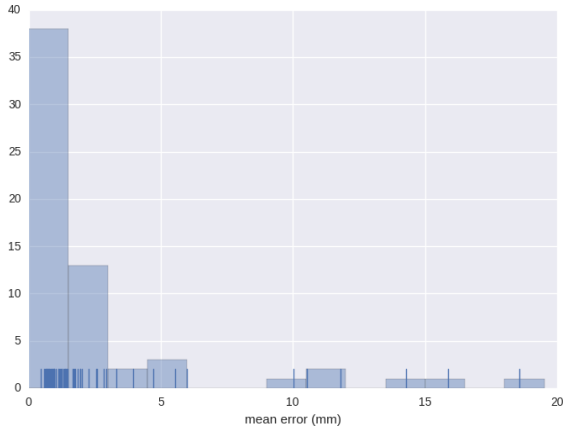

Fig. 2. Histogram of mean tracking errors. 38 of 62 landmarks in the test set have a mean tracking error of $1.5mm$ or less.

3 Experimental Results

We implemented the proposed approaches and methods in the previous section using Nolearn [3] and Lasagne [1] in Python. Lasagne uses Theano [6] for execu-

tion, which allows us to use GPUs for computations. We ran all execution using the Amazon Web Services (AWS) `g2.2xlarge` instances[1].

Each version of the network architecture was trained using a single `g2.2xlarge` instance though various network architectures and hyperparameters settings were often trained in parallel using multiple machines[2]. The sliding window search was performed using a single `g2.2xlarge` instance though each sequence could be made to run in parallel.

The tracking results for each sequence group are shown in Table 2. The results in `CIL`, `MED-1` and `MED-2` were relatively consistent with small standard deviations, 95^{th} percentiles and maximum values. For the `ETH` and `ICR` examples, there were examples where the search "got lost" and the algorithm returned a window very off from the desired target. For example the mean error on `ICR-07_2` was $18.55mm$. Figure 2 visualizes a number of outliers in the mean test error distribution.

The computational time to learn the distance metric was approximately 1.5 hours for the best performing models. Training differs slightly depending on the exact network architectures used and the number of training epochs needed for sufficient convergence. Once the distance metric was learned, the same metric was applied to each sequence group. The processing time for the search was 100msec / annotation / frame. This time was per learned distance metric. Given that we were performing an ensembling where two motion vectors were averaged in order to produce the final result, the actual time was double that: 200msec / annotation / frame though the two estimations can be performed entirely in parallel. Real-time performance of our system is well within reach considering that the GPUs we used in our experiments are about four times slower than the most modern GPUs available.

Table 2. Tracking errors for the 2D point-tracking test data.

	Mean [mm]	Std [mm]	95% [mm]	Min [mm]	Max [mm]
Sequence set					
CIL	1.65	0.97	3.49	0.01	5.13
ETH	2.61	4.33	13.35	0.01	27.70
ICR	5.80	8.86	29.01	0.03	39.39
MED1	2.13	2.25	7.10	0.01	16.83
MED2	1.53	1.03	3.83	0.02	6.41
All sequences	2.83	4.86	13.13	0.01	39.39

[1] `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using_cluster_computing.html`

[2] The cluster of machines was managed using StarCluster.

4 Conclusions

In this paper, we proposed a method of tracking target tissues in long (over one minute) 2D ultrasound sequences of liver. The proposed method uses a ConvNet to learn a distance metric which can then be used in a sliding window fashion to determine the motion vector of the tissue from the previous to the current frame. The experimental results were obtained using 24 sequences of ultrasound with 62 annotated landmarks. The results showed the proposed method has good average accuracy, though there were circumstances where the technique "got lost" and produced results far from the target.

The current implementation is not computationally optimal. The embedding of each window is computed independently, resulting in many redundant convolution operations. Using "fully convolutional networks" instead, we should be able to reach real-time performance easily [16].

Because our ConvNet-based embedding function has many degrees of freedom, it exhibits high variance. In future work, we aim to reduce variance by averaging the outputs of multiple networks trained on the same data but with different random initialization. We're also confident that running the embedding function on patches flipped vertically and horizontally and averaging results would lead to better generalization. These two techniques would both come at the expense of slower runtime performance.

When calculating a distance map inside a given search window, we observe that the map tends to be quite noisy. In future work, we want to look at smoothing functions to be able to more robustly find the correct center pixel.

References

1. Lasagne, `https://github.com/Lasagne/Lasagne`
2. Miccai challenge on liver ultrasound tracking, `http://clust.ethz.ch/`
3. nolearn, `https://github.com/dnouri/nolearn`
4. Banerjee, J., Klink, C., Peters, E., Niessen, W., Moelker, A., van Walsum, T.: 4d liver ultrasound registration. In: Ourselin, S., Modat, M. (eds.) Biomedical Image Registration, Lecture Notes in Computer Science, vol. 8545, pp. 194–202. Springer International Publishing (2014), `http://dx.doi.org/10.1007/978-3-319-08554-8_20`
5. Bell, M.A.L., Byram, B.C., Harris, E.J., Evans, P.M., Bamber, J.C.: In vivo liver tracking with a high volume rate 4d ultrasound scanner and a 2d matrix array probe. Physics in Medicine and Biology 57(5), 1359 (2012), `http://stacks.iop.org/0031-9155/57/i=5/a=1359`
6. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: A cpu and gpu math compiler in python. In: van der Walt, S., Millman, J. (eds.) Proceedings of the 9th Python in Science Conference. pp. 3 – 10 (2010)
7. Bromley, J., Guyon, I., Lecun, Y., Sckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. In: In NIPS Proc (1994)
8. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Computer Vision and Pattern Recognition,

2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 539–546 vol. 1 (June 2005)
9. Cifor, A., Risser, L., Chung, D., Anderson, E., Schnabel, J.: Hybrid feature-based diffeomorphic registration for tumor tracking in 2-d liver ultrasound images. Medical Imaging, IEEE Transactions on 32(9), 1647–1656 (Sept 2013)
10. Cireşan, D., Giusti, A., Gambardella, L., Schmidhuber, J.: Mitosis detection in breast cancer histology images with deep neural networks. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) Medical Image Computing and Computer-Assisted Intervention MICCAI 2013, Lecture Notes in Computer Science, vol. 8150, pp. 411–418. Springer Berlin Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-40763-5_51
11. De Luca, V., Tschannen, M., Szkely, G., Tanner, C.: A learning-based approach for fast and robust vessel tracking in long ultrasound sequences. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) Medical Image Computing and Computer-Assisted Intervention MICCAI 2013, Lecture Notes in Computer Science, vol. 8149, pp. 518–525. Springer Berlin Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-40811-3_65
12. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: International conference on artificial intelligence and statistics. pp. 249–256 (2010)
13. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. arXiv preprint arXiv:1302.4389 (2013)
14. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Computer vision and pattern recognition, 2006 IEEE computer society conference on. vol. 2, pp. 1735–1742. IEEE (2006)
15. Lediju, M., Byram, B., Harris, E., Evans, P., Bamber, J.: 3d liver tracking using a matrix array: Implications for ultrasonic guidance of imrt. In: Ultrasonics Symposium (IUS), 2010 IEEE. pp. 1628–1631 (Oct 2010)
16. Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. ArXiv e-prints (Nov 2014)
17. Luca, V.D., Benz, T., Kondo, S., König, L., Lübke, D., Rothlübbers, S., Somphone, O., Allaire, S., Bell, M.A.L., Chung, D.Y.F., Cifor, A., Grozea, C., Günther, M., Jenne, J., Kipshagen, T., Kowarschik, M., Navab, N., Rühaak, J., Schwaab, J., Tanner, C.: The 2014 liver ultrasound tracking benchmark. Physics in Medicine and Biology 60(14), 5571 (2015), http://stacks.iop.org/0031-9155/60/i=14/a=5571
18. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k2)$. Soviet Mathematics Doklady 27(2), 372–376 (1983)
19. Preiswerk, F., De Luca, V., Arnold, P., Celicanin, Z., Petrusca, L., Tanner, C., Bieri, O., Salomir, R., Cattin, P.C.: Model-guided respiratory organ motion prediction of the liver from 2D ultrasound. Medical Image Analysis 18(5), 740–751 (Jul 2015), http://dx.doi.org/10.1016/j.media.2014.03.006
20. Sun, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. CoRR abs/1406.4773 (2014), http://arxiv.org/abs/1406.4773
21. Vijayan, S., Klein, S., Hofstad, E., Lindseth, F., Ystgaard, B., Lango, T.: Validation of a non-rigid registration method for motion compensation in 4d ultrasound of the liver. In: Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on. pp. 792–795 (April 2013)