CNNT: Continuous Nearest Neighbor Real-Time Microbubble Tracking Algorithm for Ultrasound Localization Microscopy

Shaoxun Wu*, Weiqing Wang*, Qian wang*

***Abstract*— Ultrafast ultrasound localization microscopy has recently gained in popularity as they hold strong potential to obtain high-resolution haemodynamic maps of the vasculature of animals and humans. However, to ensure the microvessel imaging quality, the processing is too slow to be used in real-time applications or 3D applications. In this paper, we present a framework for super-resolution ultrasound microvessel imaging. We also propose an efficient nearest neighbor data structure for continuously tracking bubbles. Our core contribution is a robust novel CPU-based real-time microbubble tracking algorithm based on the new data structure. Moreover, we incorporate our method with state-of-the-art ultrasound microvessel imaging frameworks and validate its benefits.**

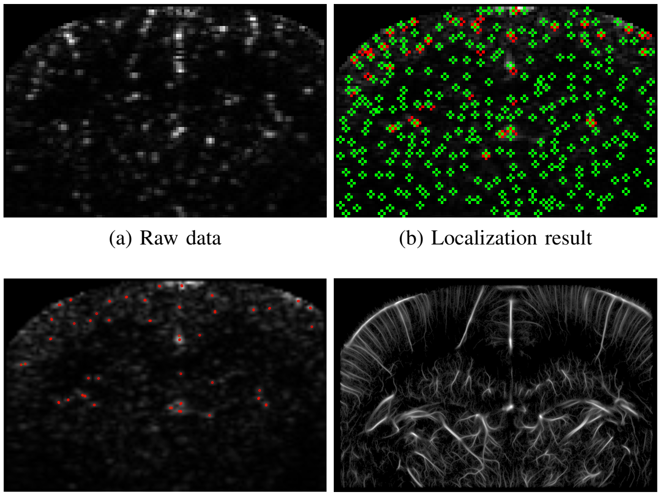
MULTIMEDIA MATERIAL

The algorithms, and the dataset we used, can be found at `https://github.com/erwin-wu-x/CNNT`

I. INTRODUCTION

Ultrasound imaging is one of the most potent tools for screening and diagnosing medical images with safe and fast. However, the quality of traditional ultrasound imaging is affected by internal scattering and acoustic diffraction of tissue, and the resolution is limited to half wavelength, which greatly limits its application in deep tissue imaging and research. *Ultrasound localization microscopy* (ULM) can break through acoustical diffraction limit, super-resolution imaging spatial resolution will ascend ten times and more, produce relevant capillaries and microstructure of structure and function of information. In terms of disease diagnosis and treatment evaluation, super-resolution imaging can break through the diffraction limit and provide rich microstructural information, making this technology a potentially powerful tool for clinical diagnosis and treatment. The current applications in disease diagnosis and treatment evaluation are concentrated in vascular diseases and brain research.

The classical ULM processing and localization algorithms [1] can be divided into several stages: acquiring data, filtering, localizing the microbubbles, pairing, and rendering. As images of a perfused organ with microbubbles are acquired at a high frame rate, real-time ULM can be more challenging than offline processing. Existed methods [1]–[3] provide high-resolution accurate and robust ULM algorithms, with more concerned about data acquiring or filtering velocity. To our current knowledge, we are the first CPU-based ULM algorithm with a tracking part that allowed for real-time processing.


(a) Raw data

(b) Localization result

(c) Tracking result

(d) Reconstructed image

Fig. 1: We present a framework for super-resolution ultrasound microvessel imaging and employ the framework for the in vivo angiography of the rat brain with continuous injection. For fig (b), the red point is localization results with ROI based methods [1], the green point is localization results with nonlocal means (NLM) denoising filter [3]. For fig (c), the red point is NLM based methods located bubbles after tracking algorithms filtering.

CPU-based real-time ULM is considered interesting from a number of perspectives. First, by using plane-wave emissions rather than line-by-line pulse-echo, frame rates were increased from 50 to 20 000 Hz giving access to rapid phenomena [4]. In the last two decades, ultrafast imaging has never been done in real-time because the processing of such ultrafast imaging sequences has always been slower than the acquisition rate. Second, the graphical processing unit (GPU) produced considerable computational power, which has led to the first full software-based clinical systems that perform image construction with software instead of hardware [1], and permitted frame rates of >1000 frames per second, which shows the CPU-based algorithm may also be sufficient for performing image in real-time. Lastly, these ultrafast modes will lead to advancements in prevention, diagnosis, and therapeutic monitoring.

We present novel ULM framework and a real-time tracking algorithm with a new data structure, our main contribution is:

*These three authors contributed equally. The authors are with the School of Information Science and Technology, ShanghaiTech University.

[1] https://www.supersonicimagine.com/

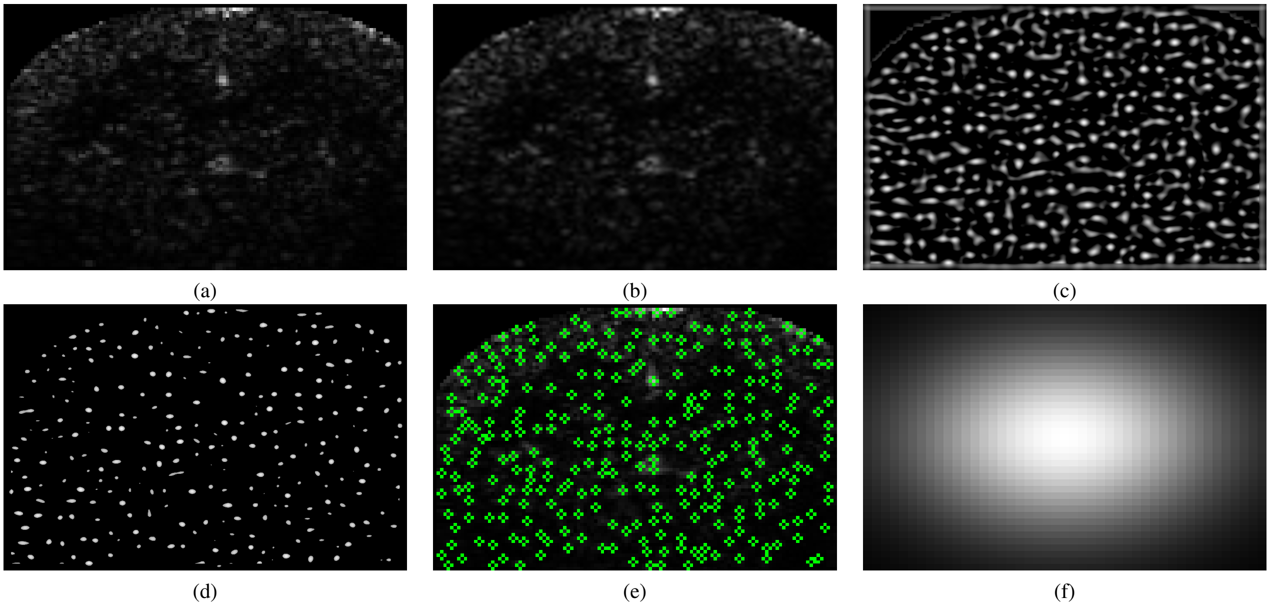


Fig. 2: Processes of microbubble localization. (a) Example of a microbubble data, followed by (b) 2-D spatial interpolation (c) 2-D cross-correlation with the derived system PSF, (d) cross-correlation coefficient thresholding, and finally identifying microbubble center locations by searching local maximum correlation coefficient. (e) Example of the localization indicated by the green points. (f) Derived PSF signal.

- A completa ULM framework with localization, pairing, tracking, and rendering stages.
- Four kinds of pairing algorithms: bipartite graph-based pairing method [3], classic Hungarian algorithm [5], semi-KNN pairing and KNN pairing algorithm [6].
- A novel data structure continuous nearest neighbor field (CNNF), which helps tracking macro-bubble moving.
- A novel tracking algorithm, which supports to real-time ULM framework.

Note that all tracking and the whole ULM algorithm are tested on a 5.0 GHz i7-12700k CPU. Without helping of GPU and considerable hardware acceleration, we hope the following work can be running on computationally constrained hardware like Raspberry pi.

II. RELATED WORK

A. Ultrasound Localization Microscopy (ULM)

In 1977, Bruneel et al. first coined the ultrafast terminology in [7]. Two years later, this same research group developed an ultrasound system able to produce entire frames simultaneously from a single acoustic pulse and reach a frame rate of 1000 images [8]. These are the precursors of ULM. Injection of micro-bubbles to improve the visualization of vessels is used in [9], which has come close to the current ULM frameworks. The time comes to 2022, [1] present a comprehensive data-processing framework for ULM and provide a series of performance metrics to evaluate quantitatively different ULM implementations.

B. Filtering and Localization

High-speed filtering is a conventional significant part of the ULM community, the term 'ultrafast' was introduced to define the velocity of data acquiring and filtering. The simple way of filtering should be high-pass filters like [10] use singular value decomposition (SVD) filter make use of the information of the temporal and spatial directions. Some other filters like nonlocal means denoising filter (NLM) [3] also show outstanding performance. Focusing on localization and tracking, we directly employ the SVD filter [1] results. Similarly, we employ the same localization method in [3] and show its advantages in the method and results section.

C. Pairing and Tracking

The pairing and tracking procedure can be universally divided into three stages [11], for simplicity, we only study the steps between frame k and $k-1$. First, we use the initial motion model to predict how the objects detected in frame $k-1$ will move. Second, we build the association between objects and update the motion model, which will be maintained lifelong. Finally, we keep tracking the objects until the data end. The most widely used algorithm in ULM is simpletracker [12], which could jump the gap between trajectories. However, this method maintained a giant adjacent matrix for each bubble and needs considerable memory and computation power, with the hardware acceleration of MATLAB, simpletracker will be twenty times slower than ours.

Deep learning based pairing methods are also brilliant. [13], [14] both show excellent results since they encoded kinds of prior information like 3D geometry. For the ULM cases, these methods may lead to severe overfitting due to low resolution, and provide worse results than conventional nearest neighbor methods.

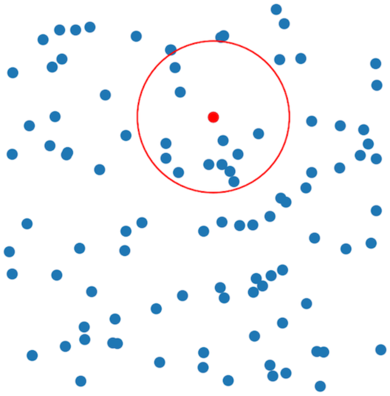


Fig. 3: The red point is the query point, the blue points are used to build a KD-tree in preprocessing stage.

III. Method and Results

This section first reviews all the parts of the proposed ULM project and the pros and cons of different methods. After a clear outline, we can introduce our new data structure and the tracking algorithm.

A. Filtering and Localization

We directly use the singular value decomposition (SVD) filtering approach. By rearranging a continuous set of beamformed images as a Casorati matrix (Nz×Nx, Nt), we could use the new basis given by the singular value decomposition to filter the tissue from the microbubbles according to its spatiotemporal decorrelation.

For the localization part, the filtered data is spatially interpolated to 0.01-mm resolution using 2-D bicubic interpolation, as shown in Fig. 2(b). A system PSF was derived based on typically observed microbubble signal dimensions and shape using a multivariate Gaussian model, which in this project is provided by Professor Cai, as shown in Fig. 2(f). The PSF was then used to perform a 2-D normalized cross-correlation with each frame of the interpolated microbubble signal, resulting in 2-D cross-correlation coefficient maps, as shown in Fig. 2(c). Pixels with a cross-correlation coefficient less than 0.6 were then rejected, creating isolated blobs, as shown in Fig. 2(d). The peaks of each isolated blob, representing the center location of the microbubble, were identified by searching for the regional maximum crosscorrelation coefficient value, as shown in Fig. 2(e).

B. Pairing

Here we compared four pairing methods: bipartite graph-based pairing method [3], classic Hungarian algorithm [5], semi-KNN pairing and KNN pairing algorithm [6].

	Bipartite graph	Hungarian	Semi-KNN	KNN
Processing Time in FPS	3496.54	9.91	1300.98	16058.27
Processing Time in FPS	3132.91	19.22	1134.29	15053.88

TABLE I: Comparision of processing time.

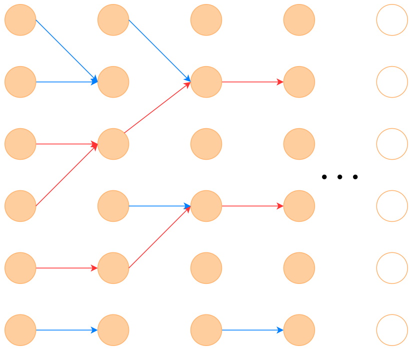


Fig. 4: Each orange point represents one micro bubble, it has properties like coordinates, id, and max trajectory length. Maintaining $P+1$ fields, each field represents one frame. With iterating pairs in the frame, we check if the length of the trajectory is equal to P, and we perform a back-tracking algorithm. If the length of the max trajectory is longer than P, we directly add the pair to the trajectory list. The red line represents the trajectory that will be added, the blue line represents pairs that will not be added.

The most widely used pairing algorithm is the classic Hungarian algorithm (a.k.a. Kuhn-Munkres algorithm). However, standard formulations of the Hungarian algorithm require that all targets in at least one of the frames (the frame with the smaller number of targets) be paired, which will be slow and sub-optimal. Bipartite graph-based pairing algorithm shows impressive performance since it throws out all microbubbles with no matches.

These two algorithms both ask for one microbubble at most has one paired object, which may not tally with conditions in practice. Considering noise, occlusion, and overlap, one microbubble may have multi-paired objects, which leads us to use the K-nearest neighbor algorithm. As Fig. 3 shows, we build the KD-tree in advance and send in the query list, then we will get the nearest neighbor concerning a radius, which is influenced by the microbubble velocity.

The comparision of processing time is in Table. I. First row is tested on a low similarity data, average bubble's number = 72.91875. Second row is tested on a high similarity data, average bubble's number = 79.50625.

C. Tracking

The core idea of our tracking algorithm is the same as [3]: the persistence control requires that the same microbubbles are paired in P consecutive frames (excluding the current frame where the targeted microbubble is in) to be counted as a reliable microbubble signal, which can then be used in final super-resolution accumulation and blood flow speed calculation.

We design a new data structure continuous nearest neighbor field (CNNF) to reduce the cost of memory and processing time. The structure of CNNF is shown in Fig. 4. With the help of CNNF, we can construct the novel tracking algorithm continuous nearest neighbor tracking (CNNT).

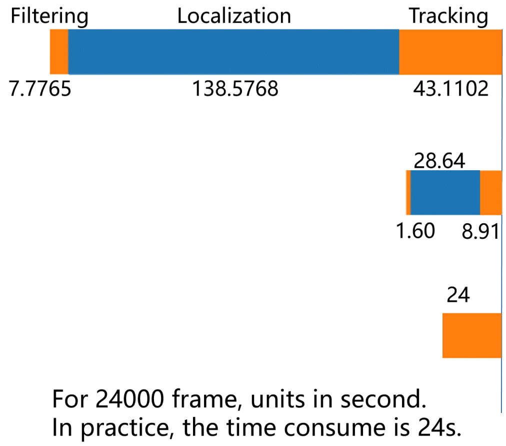


Fig. 5: The first row is the time distribution of ULM algorithm, the second row is the time we used and the correspond processing time, the third row is the data acquiring time.

The comparision of processing time and final rending results is in Table. II. First row is tested on a low similarity data, average bubble's number = 72.91875. Second row is tested on a high similarity data, average bubble's number = 79.50625.

	Bipartite graph	Hungarian	Semi-KNN	KNN
Processing Time in FPS	1931.67	8.56	957.16	2690.68
Processing Time in FPS	1439.71	18.50	805.27	2071.99
SSIM	0.512811	0.509697	0.511937	0.506128
PSNR	23.0911	23.0667	23.0492	22.4454

TABLE II: Comparision of processing time.

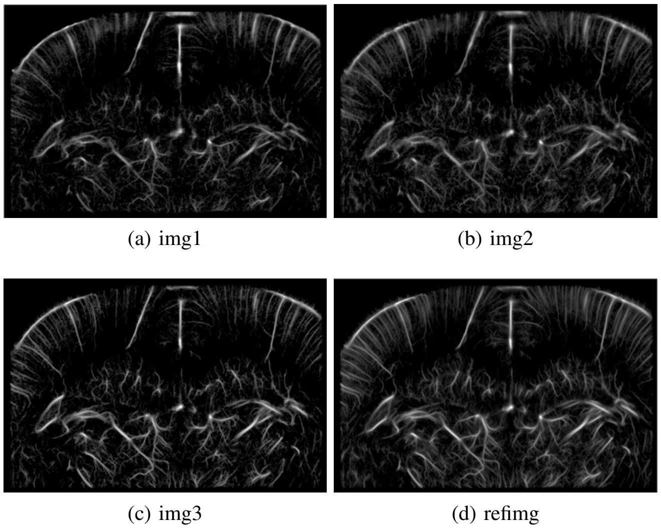
IV. Conclusion

With no hardware acceleration, CNNT has almost come to real-time processing. As shown in Fig. 5, with a approximately five times acceleration, we think we accomplished the real-time target.

V. Discussion

First, the idea to fix gap in simple tracker is reasonable, which can be added to our tracking method. However, simple tracker only using the source and target of two tracks to decide whether they can be linked to one track. We may find a way to fix gap which take the overall information into consideration.

Second, resolution of current ULM is very limited, which make it easy to deal with a large amount of frame in the current method. If the resolution is improved, it will significantly slow down the speed, which we think can be solved using the idea of keyframe. By choosing a subset of representative frames, instead of all the frames, on which to apply the algorithm, we can balance the quality of result and the speed. The key is to find a way to choose keyframe, which need further research.


(a) img1

(b) img2

(c) img3

(d) refimg

Fig. 6: img1: PSF convolution(get one digit result) + biparitite graph based pairing + CNNT. img2: ROI+PSF fitting(get multi digit result)+biparitite graph based paring+CNNT. img3: ROI+PSF fitting(get multi digit result)+biparitite graph based paring+Simpletracker. refimg: More than 30 patches of trajectories added.

Third, we think deep learning, a popular and powerful method, can be applied to improve ULM. We believe that deep learning can help us dig out more hidden features and be used to encode more prior knowledge, such as a standard blood vessels distribution of rate, which can greatly improve the quality and speed of ULM.

Finally, thanks to Professor Cai and all the TAs of this class, giving us so much help this semester, helping us overcome the epidemic situation.

VI. Appendix

The checkpoint images are shown in Fig. 6. The correspond quantative results are shown in Tab. III The worst result is the method of directly using the PSF convolution to obtain the localization and then performing the next steps. It should be noted that these three images and the reference image are all using the bipartite graph-based pairing method. The first two images use the CNNT algorithm we implemented for tracking. But img3 and reference images use simpletracker. So we think the low SSIM we get is acceptable. The reason for the poor localization results obtained by convolution directly with PSF, we think that the impact of precision is relatively large. At this time we introduce img4, img4 is the result of after overlaying ROI, using PSF to perform interpolation and convolution to obtain localization. Overall, img4 and img2 both use ROI, but the difference is not big. But the result is similar to img1, which means that img4 and img3 are only because of the sub-pixel level of localization precision resulting in poor results.

	img1	img2	img3	img4
SSIM	0.310673	0.512811	0.533330	0.302173
PSNR	18.554879	23.0911	22.276433	18.303038

TABLE III: Checkpoints results.

References

[1] B. Heiles, A. Chavignon, V. Hingot, P. Lopez, E. Teston, and O. Couture, "Performance benchmarking of microbubble-localization algorithms for ultrasound localization microscopy," *Nature Biomedical Engineering*, vol. 6, no. 5, pp. 605–616, 2022.

[2] C. Errico, J. Pierre, S. Pezet, Y. Desailly, Z. Lenkei, O. Couture, and M. Tanter, "Ultrafast ultrasound localization microscopy for deep super-resolution vascular imaging," *Nature*, vol. 527, no. 7579, pp. 499–502, 2015.

[3] P. Song, J. D. Trzasko, A. Manduca, R. Huang, R. Kadirvel, D. F. Kallmes, and S. Chen, "Improved super-resolution ultrasound microvessel imaging with spatiotemporal nonlocal means filtering and bipartite graph-based microbubble tracking," *IEEE Transactions on Ultrasonics Ferroelectrics & Frequency Control*, vol. PP, no. 2, pp. 1–1, 2018.

[4] M. Tanter and M. Fink, "Ultrafast imaging in biomedical ultrasound," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 61, pp. 102–119, 2014.

[5] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[6] S. Maneewongvatana and D. M. Mount, "It's okay to be skinny, if your friends are fat," in *Center for geometric computing 4th annual workshop on computational geometry*, vol. 2, 1999, pp. 1–8.

[7] C. Bruneel, R. Torguet, K. M. Rouvaen, E. Bridoux, and B. Nongaillard, "Ultrafast echotomographic system using optical processing of ultrasonic signals," *Applied Physics Letters*, vol. 30, no. 8, pp. 371–373, 1977.

[8] R. Delannoy, B.and Torguet, C. Bruneel, and E. Bridoux, *Ultrafast Electronical Image Reconstruction Device*. Dordrecht: Springer Netherlands, 1979, pp. 447–450.

[9] K. Wei, "Contrast echocardiography: Applications and limitations," *Cardiology in Review*, vol. 20, no. 1, pp. 025–032, 2012.

[10] L. Ledoux, P. J. Brands, and A. Hoeks, "Reduction of the clutter component in doppler ultrasound signals based on singular value decomposition: a simulation study." *Ultrasonic Imaging*, vol. 19, no. 1, pp. 1–18, 1997.

[11] Z. Pang, Z. Li, and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," *arXiv preprint arXiv:2111.09621*, 2021.

[12] J.-Y. Tinevez, "simpletracker - a simple particle tracking algorithm for matlab that can deal with gaps." 2019.

[13] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[14] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao, "Learning two-view correspondences and geometry using order-aware network," *International Conference on Computer Vision (ICCV)*, 2019.