The School of Mathematics

THE UNIVERSITY *of* EDINBURGH

# Super-Resolution Ultrasound Imaging via Kalman Filter-Based Microbubble Tracking

**by**

**Matthew Flewelling**

Dissertation Presented for the Degree of
MSc in Computational Applied Mathematics

August 2025

Supervised by
Dr Jonathan Spence and Dr Konstantinos Zygalakis

# Abstract

Microbubble (MB) tracking plays an important role in Super-Resolution Ultrasound Imaging (SRUI) as it reconstructs vascular networks and flow dynamics down to the capillary level, beyond the resolution of conventional ultrasound. Observing blood vessels at this scale can aid clinical intervention and the monitoring of treatment of tumours and other diseases. Reliable MB detection and tracking is essential for vascular flow mapping but remains challenging under noisy or high density conditions. This dissertation focused on the development and evaluation of a Kalman filter-based MB tracking algorithm for SRUI, using a synthetic dataset with ground truth annotations. The tracking pipeline combined Kalman motion modelling, distance-based linking, link rejection criteria, and track post-processing, with each being refined through targeted experiments. Development was guided by quantitative tracking metrics, where each stage was iteratively optimised for performance. Results demonstrated that the Kalman-based tracker outperformed simpler baseline methods, with post-processing strategies improving performance while linking stage rejections had limited effect. The method reconstructed a greater number of shorter tracks than present in the ground truth, and was more effective at capturing slower moving MBs. Tracking performance declined with reduced localisation quality, highlighting the dependence on accurate detections. These findings demonstrate the value of motion modelling for reconstructing vascular flow dynamics and support the broader goals of SRUI.

# Acknowledgements

I would like to thank my supervisors, Dr Jonathan Spence and Dr Konstantinos Zygalakis, for their support and valuable feedback throughout this dissertation. I am also grateful to Dr Georgios Papageorgiou for their expertise and insights on the topic, and to their team for providing code for the localisation, evaluation, and mapping components of this work.

I would also like to acknowledge the organisers of the ULTRA-SR challenge for making available the synthetic dataset that formed the basis of all my experiments and evaluation.

Finally, I would like to thank my family and friends for their encouragement and support throughout the course of this degree.

# Own Work Declaration

I declare that the following work is my own except where otherwise noted.

# Contents

## List of Tables

## List of Figures

## List of Abbreviations

**AED** Average Euclidean Distance

**CEUS** Contrast-Enhanced Ultrasound

**FN** False Negative

**FP** False Positive

**GT** Ground Truth

**MB** Microbubble

**NN** Nearest Neighbour

**PSF** Point Spread Function

**SL** Simulated Localisation

**SRUI** Super-Resolution Ultrasound Imaging

**TP** True Positive

**ULM** Ultrasound Localisation Microscopy

**US** Ultrasound

# 1 Introduction

Ultrasound (US) imaging is widely used in clinical practice due to its affordability, real-time capability, and lack of radiation. The spatial resolution of US, as with other imaging modalities, is limited by diffraction to the scale of its wavelength [9]. However, microvascular structure and the associated flow dynamics are important diagnostic indicators [14], and are not captured at this limited resolution [22]. The formation of new blood vessels in an existing vascular network can be an important predictor in the growth and metastasis of tumours [7]. With early detection, US can provide a non-invasive means of early clinical intervention and the monitoring of treatment for such diseases [46].

Conventional US imaging can be made sensitive to the previously unseen microvascular structure through the introduction of contrast agents [10]. This process is known as Contrast-Enhanced Ultrasound (CEUS), and it aims to achieve a resolution greater than the limit of diffraction, literally named super-resolution. CEUS scans are well established, so extending them into practical diagnostic applications is a possibility [9]. Clinically approved Microbubble (MB) contrast agents are administered intravenously for the CEUS process. MBs are small, typically 3 $\mu$m in diameter, gas filled bubbles which vibrate strongly at US imaging frequencies. These nonlinear oscillations generate make the MBs significantly more reflective than normal body tissues [3]. Their small size allows MBs to traverse into the entire vascular bed [27], including the microvascular structure. This results in a high contrast-to-tissue ratio and enables visualisation of the entire vascular structure. Importantly, MBs can be detected deep within the body making them advantageous as a contrast agent [9].

Despite the advantages of CEUS, it alone can not directly resolve the microvascular structure, as it remains constrained by the diffraction limit. Super-Resolution Ultrasound Imaging (SRUI) refers to a class of techniques that surpass the diffraction limit, of which Ultrasound Localisation Microscopy (ULM) introduced by Errico et al. [16], is one. ULM achieves sub-diffraction imaging by localising MBs over many CEUS frames and reconstructing vascular maps from their accumulated positions [13]. MBs act as point sources where each appears as a blurred spot corresponding to the Point Spread Function (PSF) of the imaging system, and localisation aims to estimate the centroid of the MB with sub-wavelength accuracy [8, 16]. This is a challenging task in practice, particularly in dense MB fields or when noise levels are high, where overlapping or ambiguous PSFs may occur. There exists a trade-off in imaging, where slower acquisition rates could require higher concentration of MBs to maintain sufficient detections per frame, but can complicate the localisation process. The general workflow of ULM is to acquire a sequence of CEUS frames, suppress the tissue background to isolate the MB signal, localise individual MBs, and accumulate these centroid positions across frames to form a high-resolution vascular image [9].

Beyond reconstructing the structure of vascular networks, ULM can also include techniques to gain information on MB speed, direction, and more. By linking localised MB across consecutive frames, trajectories can be established that reflect blood flow within the underlying vasculature. This task, known as tracking, is an example of a multiple-particle tracking problem [17, 39], in which a large number of particles (i.e. MBs) must be correctly identified and followed through time. With so many particles to detect over many frames, manual annotation of images is not feasible and robust computational algorithms are required to perform localisation and tracking reliably. The SRUI steps utilised in this dissertation is presented in Figure 1, where synthetic CEUS images undergo localisation and tracking before being compared to the Ground Truth (GT). Reconstruction maps are created to visualise the flow and direction of MBs.

Through the use of SRUI, the generation of microvascular flow maps with high spatial and temporal resolution could improve the diagnostic potential in clinical practice. The techniques behind ULM have been applied in the preclinical imaging of tumours [31, 37] and lymph nodes [49] for example, and demonstrated in humans in the breast [12, 37], brain [11], and liver [23] among others.

The primary aim of this dissertation is to develop, implement, and evaluate a Kalman Filter-based MB tracking algorithm as part of a SRUI pipeline. This work will take inspiration of similar methods [43, 44], and build upon the exploration of motion-model based tracking methods to improve robustness and accuracy in MB trajectory reconstruction. A synthetic data approach which draws from existing real-life data enables the testing and assessment of tracking algorithms. These synthetic datasets provide GT annotations for both the localisation and tracking processes, allowing for quantitative comparison. By systematically increasing the amount of noise to GT data, thus creating datasets with varying levels of localisation quality, the tracking performance of our proposed method can be evaluated, assessing robustness. The performance of the Kalman filter-based approach is to be compared against baseline tracking methods.

The findings of this work aim to provide insight into the role of motion modelling in MB tracking and informing the design of more reliable SRUI pipelines. Ultimately, this may support the development of clinically viable CEUS-based imaging systems capable of visualising microvascular structure and flow dynamics.

All processing was completed on a Windows 11 laptop equipped with 16 GB RAM, an Intel Core i7-10710U CPU @ 1.10GHz (Intel, Santa Clara, California, USA). All procedures were implemented in MATLAB R2025a (MathWorks, Natick, Massachusetts, USA), utilising the Image Processing and Sensor Fusion and Tracking Toolboxes.

The remainder of this paper is structured as follows. Section 2 presents the background necessary for this work, beginning with an overview of localisation and tracking and the existing methods. Section 3 introduces the synthetic dataset used in this dissertation. The localisation pipeline used is detailed in Section 4, while Section 5 outlines the proposed tracking pipeline. Section 6 presents the efforts of developing the tracking pipeline and the performance of the proposed method in comparison to baseline tracking methods across various localisation inputs. Finally, Section 7 summarises the findings and ideas for potential future work are explored.

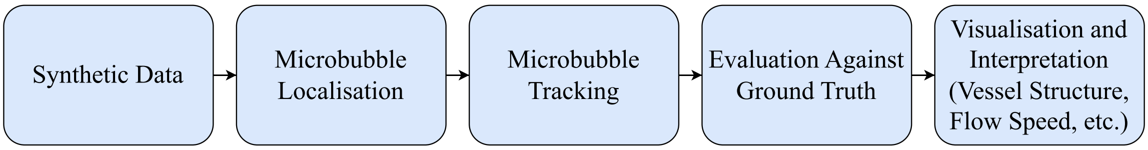



Figure 1: Overview of the Super-Resolution Ultrasound Imaging (SRUI) pipeline implemented in this dissertation.

## 2 Background

This background section introduces key stages of SRUI, localisation and tracking, and a brief introduction to Kalman filters and their application in MB tracking.

### 2.1 Microbubble Localisation Methods

Localisation is the first processing step in the SRUI pipeline, as displayed in Figure 1, with the aim of accurately identifying the spatial positions of MB in each frame of a CEUS sequence. Localisation methods in SRUI aim to surpass the diffraction resolution limit by estimating the centroid positions of individual MBs with sub-wavelength accuracies.

The localised positions, also known as detections, serve as the input of the subsequent tracking step. Therefore, errors in the localisation process, whether that be missed detections, incorrect positions, or incorrectly identifying noise as a MB, directly impacts the quality of reconstructed tracks and the final super-resolution images. Precise localisation is therefore essential to the success of the entire SRUI process.

In the frames, each MB appears as a blurred shape, which can be represented as a PSF. The PSF describes how a single point (i.e. MB) is represented in the image due to the system's resolution limits. In general, localisation methods aim to estimate the true centre of each MB by fitting or analysing its PSF. The localisation process contains many challenges in practice. High-density MB fields, excessive noise, and overlapping PSFs can all contribute to limiting the quality of MB detection and fitting. An example of a single frame is shown in Figure 4, which displays a synthetic frame from the ULTRA-SR challenge dataset (see Section 3). The variability of signal intensity which represent the MB positions are evident among the noisy background in the frame. As such, localisation methods are typically preceded by a series of processing steps to isolate MBs from the noise. This highlights the need for robust localisation algorithms that can perform under a range of imaging conditions, where the true positions of MBs are not immediately evident.

Several strategies have been proposed for MB localisation, of which a selection are reviewed here. Centroiding is one of the simplest approaches, where MBs are localised by the intensity weighted centre of mass of their isolated PSFs [8]. Gaussian fitting is another widely used alternative, where a two-dimensional Gaussian function is fitted to isolated PSFs and the peak is used to localise the MB [21]. The normalised cross-correlation method compares the observed signal with a template PSF, generating a correlation map where the peaks are the centre of MBs [41]. Other approaches, such as interpolation, deep learning, and sparsity-based methods [45, 48], have also been explored. In this dissertation, the primary focus is on evaluating MB tracking, therefore a centroid-based approach is implemented and is described in Section 4 and Appendix A.

### 2.2 Microbubble Tracking Algorithms

The purpose of tracking in SRUI is to link individual MB detections obtained in the localisation process across consecutive frames to form continuous trajectories, or tracks. These tracks represent the motion of individual MBs through the vascular structure and are fundamental in reconstructing flow maps and estimating velocities.

At the core of tracking algorithms is a frame-to-frame pairing, or linking, process, which attempts to match MBs detected in one frame to the next. Repeating this process over many frames produces full-length tracks for each MB, capturing their dynamics over time. Figure 2 demonstrates how tracking is inherently an ambiguous task. Each panel shows the same set of MBs detected across four consecutive frames, represented by the filled circles. Despite having the same input detections, many plausible tracking outcomes are possible. Two such results are shown, each having a distinct way of linking detections across frames. Additionally, it is shown how some tracks may begin or end in any given frame, highlighting how MBs can enter or leave the field-of-view, and how not all MBs are linked in the process. These factors contribute to the complexity of the task and underline the need for robust tracking algorithms which are capable of reconstructing the true tracks.

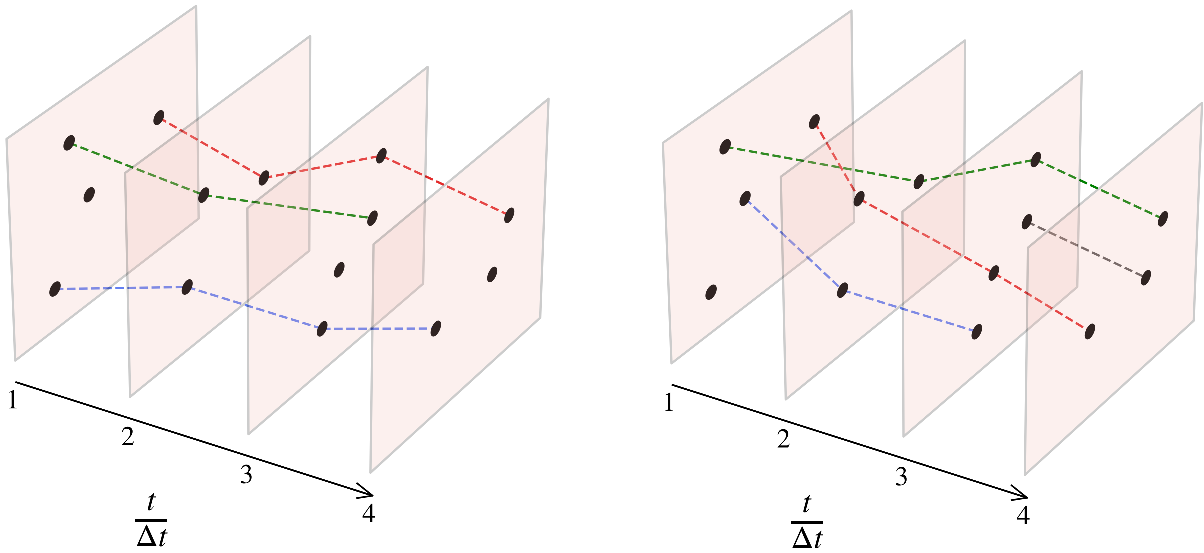
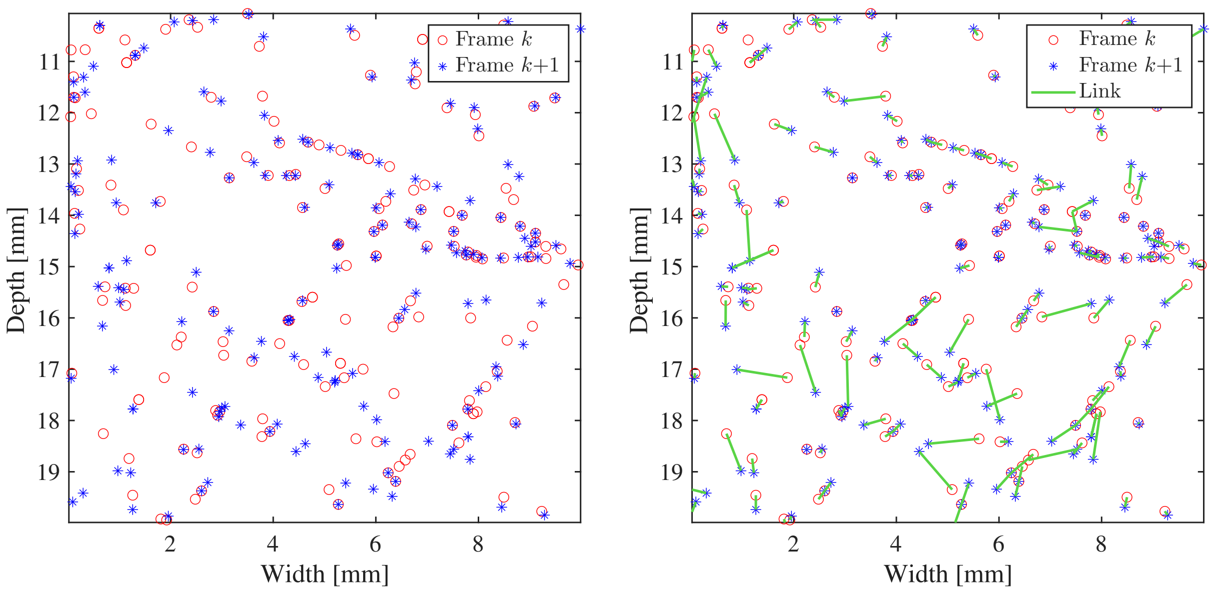



Figure 2: Visualisation of two different tracking possibilities applied to the same set of detected MBs across four consecutive frames. The filled circles mark the positions of detected MBs and dashed lines represent the tracks. Each diagram demonstrates a distinct way of linking the same detections over time, highlighting the ambiguity in the tracking process. Figure inspired by [37].

Tracking performance is highly sensitive to the the quality of the detected localisations. Errors or missing detections propagate directly into the tracking results, as they can lead to incorrect links, broken tracks, or false trajectories. In high-density scenarios, where many MBs are in close physical proximity of each other, it becomes additionally challenging to reliably form the correct links across frames. Additional complications come from the presence of regions with varying blood flow velocities. Recall that MBs are an indirect method to visualise the vascular structure, and there will be vessels of different physical characteristics. Adjacent vessels could have significantly different speeds and directions. A robust tracking algorithm should be able to reject potential links which violate the physical reality of the problem.

While Figure 2 presents a small number of detections per frame, the problem is typically much more complex than that example. Using synthetic data (further detailed in Section 3), Figure 3 illustrates ground truth linking between MBs in two consecutive frames, shown for a zoomed-in region of the field of view. It is clear that this linking process is not straightforward. In panel (a) we observe a mix of dense clusters of MBs along with some in isolation. If one were to assign the links based on the closest MB between detections across frames, there would be many incorrect links. Panel (b) demonstrates how there is a great variety in the directions and lengths of the links, with longer links indicating faster moving MBs between frames. In the dense sections the linking becomes even more ambiguous, where links are close to each other and may appear to cross. Further, in the top of panel (b) we can see MBs that do not have a true link to the next frame, indicating that track has ended, and adding complexity to the task. Similarly not all detections in frame $k+1$ will have link to the previous frame. Lastly, some MBs align in paths, which demonstrates the presence of the underlying vascular structure.

A wide range of methods have been developed for MB tracking, of which a selection are discussed here. At the core of most tracking algorithms lies an assumption that MBs move continuously and are therefore expected to be close to their previous position. A basic approach to MB tracking is the Nearest Neighbour (NN) method, where each detection in the current frame is linked to the closest detection in the following frame [18, 33]. This method is computationally efficient and is known to perform well in sparse MB fields [27]. However, in high MB density scenarios where trajectories cross or PSFs are overlapping it is known to be prone to errors [37]. To remedy these shortcomings, several extensions have been proposed with the objective




(a) MB detections in frames $k$ and $k$+1, without links.

(b) MB detections in frames $k$ and $k$+1, with true links.

Figure 3: Ground truth MB linking example in a zoomed-in region of two consecutive frames from the ULTRA-SR dataset. (a) shows the MB positions and (b) includes the true links between MBs illustrating motion across frames.

to apply restrictions and constraints such that the links created with the Nearest Neighbour (NN) algorithm are more realistic. Examples of this include rejecting backwards motion or the incorporation of velocity-based constraints, each showing an improvement over the baseline NN implementation [5, 34].

Beyond a NN-based approach, other approaches have been used to perform the matching of detections across frames. The Hungarian algorithm [28] finds the globally optimal set of pairings between frames based on minimising the distance between pairs and has been used in SRUI [29, 43]. Similarly, bipartite graph-based pairing methods have been explored [41, 44]. These methods come at the expense of higher computational cost when compared to a NN approach.

Markov Chain Monte Carlo (MCMC) methods have also been applied to MB tracking [1, 36]. These take a probabilistic approach, where from localisations a set of possible tracks between detections are defined according predetermined rules (such as, start new track, extend track, split track, etc.). A posterior distribution of possible tracks are explored using MCMC sampling and the most likely set of tracks are selected. This allows for the modelling of uncertainty but at the expense of higher computational requirements.

The multiple hypothesis tracking (MHT) algorithm [38] is a popular data association technique [2] and has been applied towards SRUI [40]. The key idea in MHT is to produce potential hypotheses of links from a given detection. The likelihood of each potential link is calculated upon receiving new measurements and the most likely track is selected.

### 2.3 Kalman Filters

The Kalman filter was introduced by Rudolf E. Kalman in 1960 as an estimator of a dynamic system's state in the presence of noise [26]. It has since been found widespread use in control engineering, the aerospace industry, navigation systems, and signal processing [20], where its ability to provide real-time state estimation from noisy data and simplicity makes it a powerful tool. The key objective of the Kalman filter is to obtain the best estimate of a system's true (but hidden) state by optimally combining a dynamic motion model with uncertain and noisy measurements.

The Kalman filter operates in two stages that are repeated sequentially: The prediction stage and the update stage. In the prediction stage, a motion model estimates a system state's future state based on its previous history. This estimated state is accompanied by a measure of its uncertainty. Given a measurement of the system, the update stage refines the estimated state by optimally combining it with the observation with consideration of their respective uncertainties. The result is an improved state estimate and an updated measure of uncertainty. In its simplest form, the Kalman filter assumes linear dynamics, but non-linear extensions have been presented in the extended Kalman filter [47], and the unscented Kalman filter [25], for example. Many further extensions of have since been developed. The Kalman filter model and its implementation for the purpose of this dissertation are described in Section 5.4.

As previously mentioned, localisations are noisy, potentially ambiguous, and imperfect. Kalman filters address these issues by exploiting motion history to smooth trajectories, rejecting implausible track formations, and has the ability to fill in for missing detections. Kalman filtering was first applied to MB tracking in US imaging by Ackermann and Schmitz [1] and Solomon et al. [40], where it was used as just one component of a larger tracking pipeline. Tang et al. applied Kalman filters as a post-processing step to refine the output of other tracking methods and improve accuracy [44]. Taghavi et al. used Kalman filtering to predict future MB positions and link them to detections in subsequent frames [43]. The latter approach forms the basis of the method developed in this dissertation and is described in Section 5.

# 3 Synthetic Data

The development and evaluation of MB localisation and tracking algorithms for SRUI require datasets with known ground truth. While the ultimate goal of SRUI is to apply these techniques to *in vivo* data, such data lack a ground truth making quantitative assessment challenging. Synthetic datasets provide a controlled environment in which parameters such as MB density, motion, frame rate, and noise can be varied systematically. This provides the capability to perform reliable and robust benchmarking of different algorithms. Further, synthetic datasets could allow for the exploration of algorithm behaviour in edge-case scenarios, and support repeatable comparisons across different studies.

In this section we describe the synthetic datas used for development and assessment in this dissertation, the ULTRA-SR challenge dataset. For completeness, we also note that a very sparse secondary dataset, containing only one or two MBs per frame, was used in the development to verify basic algorithm functionality before progressing to the more challenging ULTRA-SR data.

## 3.1 ULTRA-SR Challenge Dataset

The data used in this dissertation was sourced from the Ultrasound Localization and TRacking Algorithms for Super-Resolution (ULTRA-SR) challenge, an initiative to evaluate and benchmark current MB localisation and tracking methods. Challenge organisers provided synthetic datasets at varying ultrasound frequencies, defined evaluation metrics for both the localisation and tracking processes, and issued an open call for algorithm submissions. Further, the submitted algorithms were tested on *in-vivo* datasets. Full results and methodology of the challenge are presented in the accompanying post-competition paper [29].

This dissertation focuses on the high-frequency synthetic dataset, labelled net_003 or High Freq, and is one of the datasets used for the challenge's evaluation. For clarity, this dataset will simply be referred to as the ULTRA-SR challenge dataset, and its accompanied ground truth annotations as the GT dataset. The data was selected at the recommendation of our supervisor as it contains a MB density higher than one would typically see, making it an interesting and challenging test case for the development and evaluation of SRUI pipelines. The dataset was generated using the BUbble Flow Field (BUFF) simulation framework [30], which randomly generates vascular trees and flowing MBs. The resulting data satisfy Poiseuille's law (via a Hagen-Poisuille flow model), creates a 3D vascular tree generator, and US signal simulator with non-linear dynamics. The data is available at https://www.ultra-sr.com.

A summary of the synthetic dataset's simulation parameters is provided in Table 1. The dataset is comprised of 500 synthetic B-mode US frames of size $512 \times 512$ pixels. Each image represents a 2D slice through a 3D simulated vascular network. The physical field-of-view (image limits) spans 20 mm in both width and depth. The acquisition frame rate $f_r$ is 50 Hz, and all spatial coordinates are expressed in SI units. For simplicity in processing, the spatial coordinates are converted to pixels using the parameters in Table 1, and are considered temporally by frame.

Each frame includes approximately 550 MBs, which appear as bright, blurry, bubble-like features against a noisy background. MBs move according to physiologically realistic flow velocities, according to the BUFF simulation framework, and may enter or exit the imaging plane over time. This means that we expect MBs to not persist across all frames in the dataset. An example frame, shown in Figure 4, illustrates the visual appearance of the data with and without MB position overlays. This figure demonstrates how the MBs are difficult to distinguish from the noisy background. Table 2 further summarises key dataset statistics, including total MB count, mean MB density, and track length distributions.

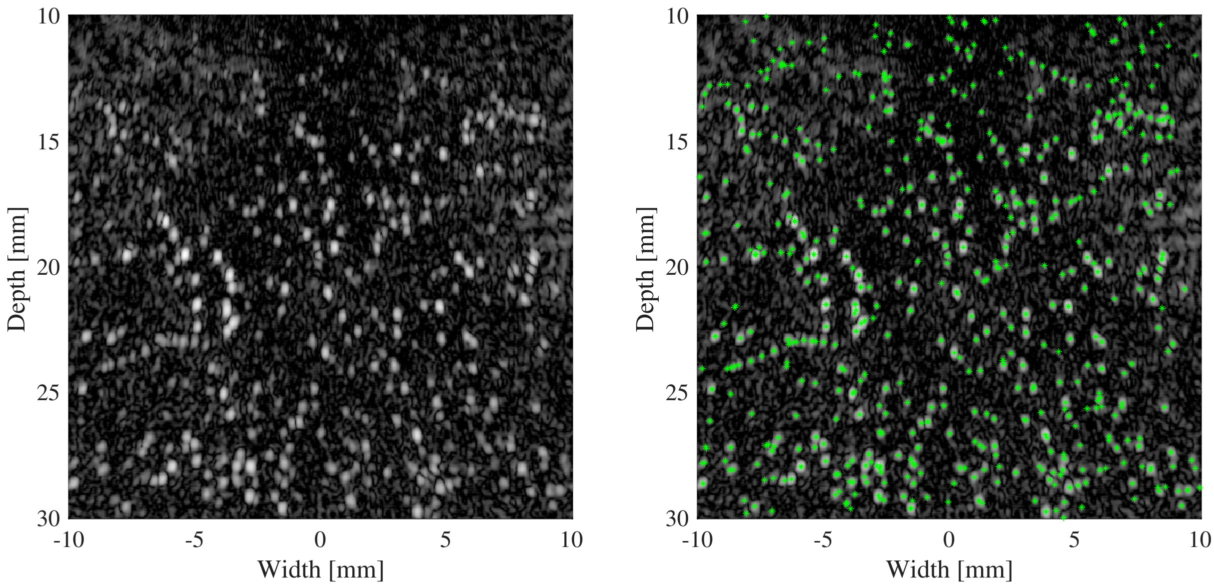
GT data is provided as a plain text file, where each row represents a localisation event in the format: [frame number, MB ID, $x$, $y$, $z$]. Since the image data is 2D, only the $x$ and $z$ coordinates are used, representing the width and depth dimensions respectively. Each MB retains the same ID across all frames in which it appears, allowing its positions over time to be collected into a track. These trajectories can then be analysed to compute velocities, direction of motion, and other flow characteristics.

| Simulation Settings | |
|---|---|
| Dataset ID | GT |
| Number of frames | 500 |
| Dimensionality | 2D + time |
| Image size (pixels) | $512 \times 512$ |
| Frame rate (Hz) | 50 |
| Signal-to-noise ratio (dB) | 19 |
| Sampling Frequency (MHz) | 80 |
| Transmitted frequency (MHz) | 7.24 |
| **Field of View** | |
| Pixel size (mm/pixel) | 0.3914 |
| Width range (mm) | $[-10, 10]$ |
| Depth range (mm) | $[10, 30]$ |

Table 1: Summary of imaging and simulation parameters of the ULTRA-SR synthetic dataset [29].

| | |
|---|---|
| **Dataset Info** | |
| Dataset ID | GT |
| **Microbubble Statistics** | |
| Total number of MBs | 275,163 |
| Mean number of MBs per frame | 550.33 |
| Mean MB density | 137.58 MB/cm$^2$ |
| **Track Statistics** | |
| Number of tracks | 7,802 |
| Mean track length | 35.27 frames |
| Minimum track length | 1 frame |
| Maximum track length | 500 frames |

Table 2: Summary of MB and track statistics for the ULTRA-SR ground truth dataset.




(a) Frame without annotations.

(b) Ground truth MB positions overlaid in green.

Figure 4: Example frame (frame 1) from the ULTRA-SR synthetic dataset, with and without true locations of MBs overlaid. This frame is representative of the characteristics across the dataset and highlights the difficulty of determining MB positions from the noisy background.

# 4 Localisation

The goal of the localisation stage is to detect and accurately estimate the positions of individual MBs within each US frame. Reliable localisation is essential for establishing links between detections across frames and consequently for reconstructing vascular maps and estimating blood flow dynamics. In addition to the position, the localisation process also determines signal intensity and size for the MBs.

Each detection represents a discrete sampling of the vascular structure and flow. Missed detections therefore lead to information loss and can reduce the quality of estimation of flow. Conversely, incorrect detections can lead to misrepresentations of the true structure. Localisation is a non-trivial task due to several practical challenges: MBs can appear faint or obscured within a noisy background and they may also overlap each other in regions of high cluster density, making it difficult to distinguish individual bubbles. Figure 4 demonstrates how background noise complicates the determination of MB position.

## 4.1 Localisation Pipeline

The localisation process implemented in this dissertation follows the pipeline illustrated in Figure 5. The first stage applies a series of pre-processing steps designed surpress background noise isolate the MB signals. Following this, PSFs of candidate MB are localised using an intensity-weighted centre of mass calculation. Lastly, detections are filtered using tuned size and intensity thresholds to discard candidates that do not meet the selection criteria.

All methods presented in the pipeline follow the procedure described by Kanoulas et al. [27]. The output is a set of MB positions for each frame. Localised MBs are assigned arbitrary IDs at this stage, which are independent of their corresponding IDs in the GT.

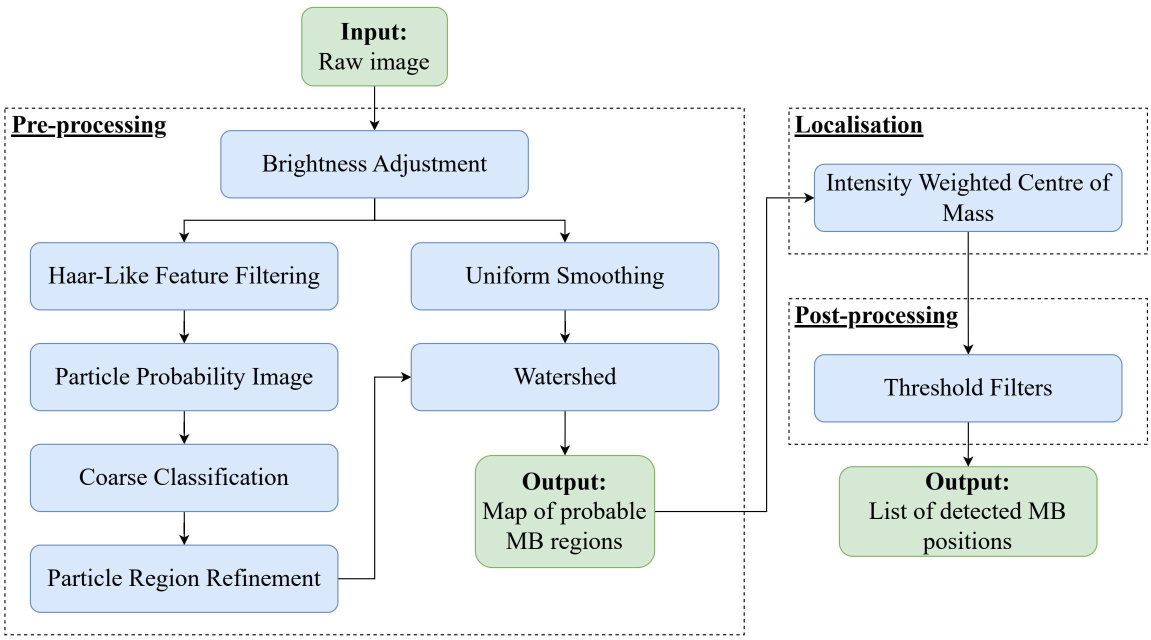



Figure 5: Data processing pipeline for MB localisation. Pre-processing steps transform raw ultrasound images to isolate and determine the positions of MBs from the background and noise.

As the primary focus of this dissertation is the development and evaluation of a tracking algorithm, implementation notes of the localisation pipeline are provided in Appendix A. Table 3 summarises a selection of tuned localisation parameters used, the minimum and maximum MB area (in pixels), and the minimum average pixel intensity of a MB to be distinguished from the background. These values were optimised empirically using the ULTRA-SR synthetic dataset by iteratively adjusting parameters until no further improvement was observed to the evaluation metrics described in Section 4.2.

| **Parameter** | **Value** |
|---|---:|
| Minimum particle size | 20 (pixels) |
| Maximum particle size | 400 (pixels) |
| Intensity threshold | 60 (grayscale units, [0-255]) |

Table 3: Subset of optimised parameters (LOC-OPT) used during the image pre-processing and MB localisation stages.

## 4.2 Localisation Evaluation Metrics

The objective of localisation evaluation is to define quantitative metrics to compare MB localisations with GT data when they are available. These metrics present the fractions of correct detections and the estimated error. We follow the methodology presented by Lerendegui et al. [30] in the ULTRA-SR challenge. Since detected MBs are assigned arbitrary identifiers in the localisation process, a naive direct comparison to the ground truth MB ID is not meaningful. Instead performance is assessed spatially, where localisations are matched to GT MBs based on proximity using a defined search radius. The search radius is defined as being half of the transmitted wavelength since the spatial resolution of ultrasound systems is limited by diffraction. We therefore define the search radius as $\lambda/2$ and centred around a ground truth MB, where the transmitted wavelength $\lambda$ is given by:

$$\lambda = \frac{c_{\text{soft tissue}}}{f_{\text{transmitted}}}, \tag{4.1}$$

where $f_{\text{transmitted}} = 7.24$ MHz [29] and $c_{\text{soft tissue}} = 1540$ m/s [32] is the standard approximation of speed of sound through soft tissue [42]. This yields

$$\frac{\lambda}{2} = \frac{1}{2}\frac{1540 \text{ m/s}}{7.24 \times 10^6 \text{ Hz}} \approx 1.06 \times 10^{-4} \text{ m}. \tag{4.2}$$

Using the spatial resolution defined in Table 1, this corresponds to approximately 2.72 pixels. This search radius can be considered the uncertainty in evaluating localisation accuracy.

To perform this spatial matching, we use MATLAB's matchpairs.m function, which solves a bipartite assignment problem to find an optimal pairing between localisations and ground truth positions. This method is described in more detail in Section 5.5.1, where its use in the tracking process is explained.

A detection is considered a True Positive (TP) if a localisation lies within the radius of a ground truth MB position. A False Positive (FP) is a detected MB which is not matched to any ground truth MB. Lastly, a False Negative (FN) is a ground truth MB without a matched detection. From these definitions, we compute the following metrics:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4.3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4.4}$$

$$\text{AED} = \frac{\sum_{i \in \text{TP}} ||B_i - L_i||}{N_{\text{TP}}},$$

where $||B_i - L_i||$ is the Euclidean distance between paired TP MBs with ground truth position $B_i$ and localisation position $L_i$, and $N_{\text{TP}}$ is the number of TPs. Precision and recall inform on the quality and completeness of localisation, where the Average Euclidean Distance (AED) is a measure of spatial accuracy. Simply put, a localisation is considered better with higher precision and recall and lower AED. These metrics provide a framework to assess localisation performance under synthetic and experimental conditions where ground truth positions are known. Of course, for *in vivo* data where ground truth is unknown, these metrics do not apply.

### 4.3 Localisation Results

Based on the evaluation metrics detailed in the previous section, an iterative testing of different thresholding, filtering, and fitting configurations were applied in the localisation pipeline to optimise performance. The output MB positions (detections) were compared against the GT, and the resulting metrics were examined to select a configuration that offered the most balanced performance. The optimised parameters are summarised in Table 3.

In practice, a trade-off appeared where parameters that yielded high precision came at the cost of low recall (missing many true detections) and conversely maximising recall led to low precision (more FPs detections). Table 4 summarises the metrics from a selection of trials during the optimisation process. LOC-2 and LOC-3 demonstrate the extremes of prioritising precision or recall, whereas LOC-4 demonstrates a more balanced outcome. The final optimised configuration, LOC-OPT, achieved a high precision with a modest decrease in recall, favouring reliable detections even if some true MBs were missed.

| | **Localisation Metric** | | |
|---|---|---|---|
| **Result ID** | **Precision (%)** | **Recall (%)** | **AED (mm)** |
| LOC-OPT | 79.17 | 54.89 | 0.0443 |
| LOC-2 | 88.95 | 49.77 | 0.0423 |
| LOC-3 | 47.75 | 62.65 | 0.0441 |
| LOC-4 | 62.89 | 61.18 | 0.0439 |

Table 4: Localisation evaluation metrics for the optimised parameter set (LOC-OPT) compared with selected alternative configurations (LOC-2 to LOC-4), illustrating the trade-off between precision and recall. AED is the Average Euclidean Distance between detections and ground truth.

Figure 6 shows a zoomed-in region of an example frame comparing the GT position and estimated localisations for the optimised parameters. The figure highlights the typical localisation performance, with some detected MBs falling in or outside of the search radius as was defined in (4.2). This image demonstrates the difficulty of the localisation problem, where the FNs (missed GT positions) are sometimes indistinguishable from the background noise.

Despite the tuning efforts, the final performance fell short of the empirical target set by our advisor, who noted precision and recall localisation metrics below 80% is generally not suitable for reliable MB tracking. Since the focus of this dissertation is the implementation of tracking algorithms, this limitation increased motivation for the creation of simulated localisation results. This process adds a controlled level of noise to the GT position data to achieve a desired localisation metrics, as described in the following section.

### 4.4 Simulated Localisation Datasets

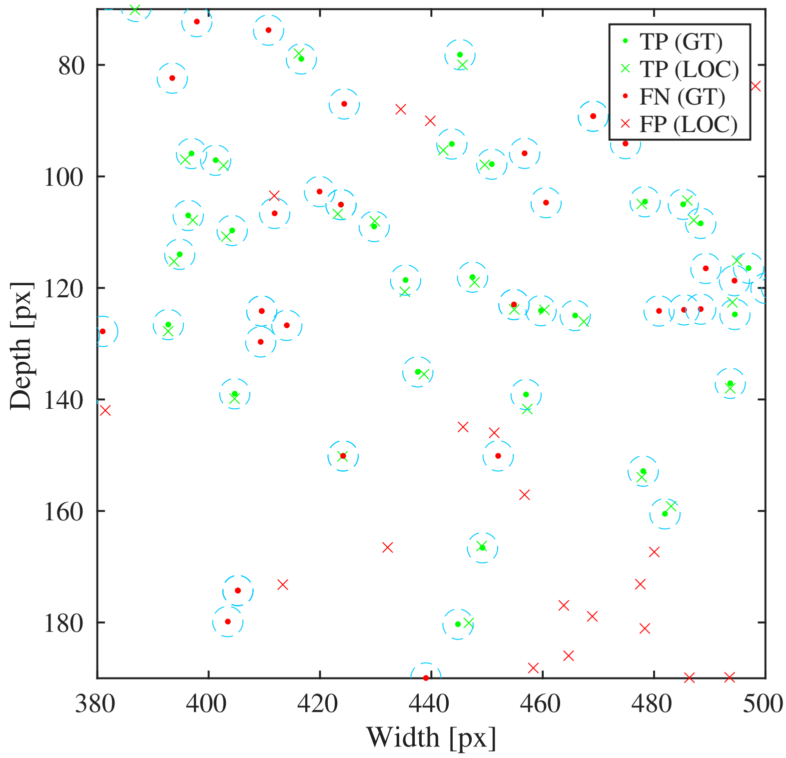
To evaluate the performance of the tracking algorithm independently of the localisation process, we generated a series of simulated localisation results based on the actual GT positions from the ULTRA-SR challenge dataset. These datasets were created to mimic realistic localisation imperfections, allowing the testing of tracking robustness under controlled degradation of localisation quality.

Each variant of the GT position data was created through three modifications. For each frame, 2D Gaussian noise was added to the $(x, z)$ positions of each MB. This noise was added with standard deviation $\sigma$ and was represented as a fraction of the transmitted wavelength $\lambda$ as was defined in (4.1). Next, random drop out of MBs was applied, where 2.5% of the MBs in each frame are removed to simulate missed detections (FN). Lastly, 2.5% of the number of MB in the frame were added at random positions to simulate FP detections. The simulated localisations were then evaluated against the GT dataset using the metrics described in Section 4.2. Table 5 summarises the parameter used to create each dataset variant and the corresponding localisation evaluation metrics.

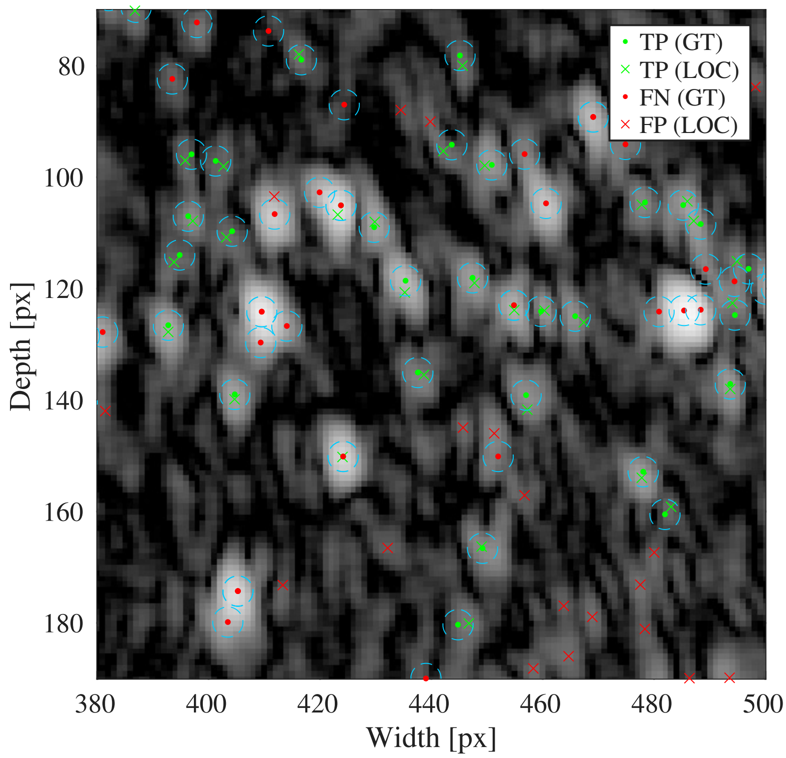
| **Dataset ID** | **Parameter** | **Localisation Metric** | | |
|---|---|---|---|---|
| | **Noise $\sigma$ (mm)** | **Precision (%)** | **Recall (%)** | **AED (mm)** |
| SL-95 | 0.0372 | 94.97 | 94.91 | 0.0447 |
| SL-90 | 0.0447 | 89.80 | 89.75 | 0.0504 |
| SL-85 | 0.0500 | 84.77 | 84.72 | 0.0531 |
| SL-80 | 0.0542 | 80.19 | 80.15 | 0.0552 |
| SL-75 | 0.0590 | 75.13 | 75.08 | 0.0569 |
| SL-70 | 0.0638 | 69.91 | 69.86 | 0.0581 |

Table 5: Simulated localisation dataset variants used to evaluate tracking performance. Noise represents standard deviation of added Gaussian noise.

The datasets are listed from SL-95 to SL-70 to represent their localisation quality via their precision and recall metrics, ranging from approximately 95% to 70% respectively. The drop and add rates in each dataset is fixed at 2.5% such that the total number of events in each dataset remains roughly the same. Further, since the drop and rates are the same, within each dataset the precision and recall are nearly identical. This is not necessarily an outcome that is observed in actual localisation results, as seen in Table 4. The original GT dataset contains 275,163 detections across 500 frames, while each variant contains 274,996 detections due to small rounding effects during processing. This framework provides a systematic analysis of how tracking performance is impacted by varying levels of localisation quality.




(a) Localisation results showing ground truth positions (dots) and detected localisations (crosses). Blue circles indicate the search radius used for pairing during metric evaluation.




(b) Same localisation results as in (a), overlaid on the corresponding raw image frame.

Figure 6: Example zoomed-in region of a single frame in the dataset, illustrating the localisation output compared to ground truth. Panel (a) shows the localisations and ground truth positions in isolation, while panel (b) overlays them on the original image. The blue circles denote the search radius used to match ground truth positions with detections when computing localisation metrics.

# 5 Tracking

Through the localisation process, a set of detected MB positions is obtained for each frame. The objective of tracking is to associate these detections across frames to form continuous MB tracks. We wish to establish tracks which truthfully occur in the GT data. This task is complicated by factors such as high MB density, the presence of both correct and incorrect localisations, and the fact that MBs can enter or leave the field of view at any time. Consequently, in a given frame, a track may not always have a true continuation in the next frame and not all detections will have an associated position in the previous frame. We make an important distinction between links and tracks: A link refers to the pairing of two localisations between consecutive frames, whereas tracks is a sequence of such links representing the motion of a single MB from its first to last detection.

In this section, we describe our approach to the tracking problem, illustrated in Figure 7. We first outline the evaluation metrics used to assess tracking performance, followed by the baseline tracking methods used for benchmarking. We then present our Kalman filter-based pipeline in full detail, including linking, rejection criteria, and post-processing.

## 5.1 Tracking Evaluation Metrics

Following the methodology presented by Lerendegui et al. [30], tracking performance is assessed by measuring the accuracy with which localised MBs are associated across consecutive frames compared to the ground truth. The purpose of tracking is to correctly identify MB trajectories over time and avoid the generation of false tracks, enabling the construction of valid vascular structures.

As previously mentioned, tracking performance is inherently dependent on localisation quality. The presence of many FP and FN MB detections will lead to incorrectly identified pairs or missing pairs in the tracking process. An incorrectly localised MB should not be paired, and a MB which was not detected can not be paired. If all localisations, correct and incorrect, were considered in tracking evaluation there would be a double penalty for FP and FN localisations. Therefore, concerning the evaluation of tracks, only TP localisations are considered. That is, only links of correctly localised MBs are used. This will reduce the overall number of pairings to work with and introduces some measure of bias into the evaluation. However, with high localisation quality (in terms of metrics as described in Section 4.2) and as the total number of localisations increase this bias will decrease.

Similarly to the localisation process, in tracking paired MBs are assigned to tracks with arbitrary IDs. This allows MB trajectories to be reconstructed by following consecutive localisations that share the same track ID. However, this means the track IDs cannot be simply compared to the ground truth. To obtain metrics, we consider a True Positive (TP) pairing corresponds to a correctly linked pair of localisations belonging to the same MB as defined in the ground truth. A False Positive (FP) pairing occurs when localisations of two different MBs are given the same ID. Lastly, a False Negative (FN) pairing is when localisations from the same MB are incorrectly assigned different IDs. Precision and recall as defined in (4.3) and (4.4) respectively were used to indicate the correctness of tracking.

To evaluate performance and quantify the distance error for tracking, the distance between pairs can be used. A weighted Jaccard index with the weights being the distances of each pair is defined:

$$J = \frac{\mathrm{TP}_d}{\mathrm{TP}_d + \mathrm{FP}_d + \mathrm{FN}_d},$$

where:

- $\mathrm{TP}_d = \sum_{i\in\mathrm{TP}} ||\mathrm{pair}_i||$ is the total distance of correctly linked MB pairs,
- $\mathrm{FP}_d = \sum_{i\in\mathrm{FP}} ||\mathrm{pair}_i||$ is the total distance of incorrectly linked MBs,
- $\mathrm{FN}_d = \sum_{i\in\mathrm{FN}} ||\mathrm{pair}_i||$ is the total distance from missed pairs.

The Jaccard index is a measure of similarity between two datasets, in our case that is correctly localised tracking results and the respective ground truth. A score of 0 is considered the worst is 1 is the best. Making the index weighted by distance reflects the difficulty of correctly pairing over long distances, as pairing over long distances is susceptible to error but when done correctly yields a greater score.

To improve interpretability of the metric a remapping of the range $[0, 1] \to [-1, 1]$ is performed:

$$J_{\text{map}} = \frac{\text{TP}_d - \text{FP}_d - \text{FN}_d}{\text{TP}_d + \text{FP}_d + \text{FN}_d}.$$

In this form, $J_{\text{map}} = 1$ indicates perfect tracking, $J_{\text{map}} = 0$ indicates as much correct pairing distance as incorrect/missing, and $J_{\text{map}} < 0$ indicates more error paired distance than correct. As with localisation, these metrics rely on synthetic ground truth data and are only applicable in synthetic or experimental datasets. For *in vivo* data, where true tracks are unknown, alternative evaluation methods must be considered.

These tracking metrics serve two purposes in this dissertation. First, they guide the development process, enabling us to evaluate implementation choices with the goal of maximising the presented tracking precision, recall, and mapped Jaccard index. Second, they provide the ability to compore our proposed method against baseline tracking methods and validated using synthetic localisation data. All results based on these metrics, from both the development and method performance evaluation stages, are presented in Section 6.

## 5.2 Baseline Methods

To reasonably compare our proposed Kalman filter-based method, we also present two baseline approaches to the MB tracking problem: Nearest Neighbour (NN) and the Hungarian algorithm. For implementation details and further information on these methods, see Appendix B. Fundamentally, both methods are developed under the rationale that MBs in subsequent frames that have the shortest distance are most likely to be linked.

For both methods a cost matrix is computed which is the pairwise distance between localisations in frames $k-1$ and $k$ with size nTracks $\times$ nDetections, where nTracks is the number of tracks in frame $k-1$ and nDetections is the number of detections in frame $k$. The baseline methods present ways to assign pairs between tracks and detections given the cost matrix. These pairing assignments are then used to either append detected MBs to tracks, end tracks in frame $k-1$ with no assigned detections in frame $k$, or create a new track in frame $k$ if a detection is not assigned to a track in frame $k-1$.

The NN implementation is a greedy local assignment, where the track IDs in frame $k-1$ are looped through arbitrarily and selects the detection in frame $k$ which is the closest in terms of distance. Used detections are not considered for future iterations of the looping of tracks, thus making it a local minimum and not a global minimum assignment. These assignments do not consider flow direction or motion of MBs.

The Hungarian algorithm is first introduced by Kuhn [28] and returns the global optimum minimum distance assignment of a square cost matrix. The algorithm was later improved by Munkres [35] to reduce the complexity. The Munkres varation is typically what is implemented when discussing the Hungarian algorithm. Further adaptations, such as [4], extends the methods to rectangular matrices, thus enabling partial assignments when the number of rows and columns differ. Our method utilises an efficient MATLAB implementation of the Munkres algorithm by Yi Cao [6], found on the MATLAB File Exchange.

In short, the Hungarian algorithm performs iterative transformations to the cost matrix through row and column reductions, leading to a set of zero-cost entries. It the identifies the minimal set of lines needed to cover all zeroes in the matrix. Any entries not covered by these lines are further manipulated to create additional zeros. This process is repeated until the optimal one-to-one assignment is found while ensuring a global minimum cost.

## 5.3 Proposed Method Overview

Figure 7 illustrates the workflow of the proposed MB tracking method. At the core of the approach is a Kalman filter-based motion model, which predicts the likely position of each MB in the next frame based on its previously estimated state. These predicted states are then linked to detections in the same frame. This process is repeated across all frames to build MB trajectories.

We refer to active tracks as those that have been successfully linked to a detection in the most recent frame, and are therefore in consideration for continuation in the current frame. The synthetic data is presented in units of time in seconds and position in metres. For ease of implementation, the data is considered on a per frame basis, with time scaled such that one unit corresponds to the time interval between frames, and space is scaled such that one pixel corresponds to one spatial unit. The following sections describe the tracking pipeline in detail, explaining each of the steps as outlined in Figure 7.

## 5.4 Kalman Filtering

For clarity, this part of the pipeline is divided into two components: the Kalman filter motion model [26], which defines the assumed dynamics of MB motion, and the algorithm implementation used to update their states over time.

### 5.4.1 Kalman Filter Model

The motion of each MB is modelled using a discrete-time linear state-space formulation. The state vector of a moving MB at frame k is defined as:

$$x(k) = [x_1(k),\ x_2(k),\ dx_1(k),\ dx_2(k)]^T \in \mathbb{R}^4, \tag{5.1}$$

where $x_1(k)$ and $x_2(k)$ represent the axial and lateral positions respectively, and $dx_1(k)$ and $dx_2(k)$ are their corresponding velocities.

The state motion is predicted according to a constant velocity motion model:

$$x(k) = \mathbf{F}x(k-1) + \epsilon(k), \quad \epsilon(k) \sim \mathcal{N}(0, \mathbf{Q}), \tag{5.2}$$

where $\epsilon(k) \in \mathbb{R}^4$ represents zero mean Gaussian noise that influences MB locations with covariance $\mathbf{Q}$. The state transition matrix $\mathbf{F}$ models the constant velocity dynamics and is given by:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.3}$$

where $\Delta t = 1/f_r$ is the time between frames and $f_r$ is the frame rate. Since the system is modelled on a per-frame basis, we set $\Delta t = 1$. This matrix is applied to the previous state vector $x(k-1)$ to compute the predicted state at frame $k$.

The observation model assumes that the detected position of each MB is a 2D localisation:

$$y(k) = \mathbf{H}x(k) + \nu(k), \quad \nu(k) \sim \mathcal{N}(0, \mathbf{R}), \tag{5.4}$$

where $y(k) \in \mathbb{R}^2$ is the observed axial and lateral position. The localisation process introduces noise into the detections, therefore $\nu(k) \in \mathbb{R}^2$ represents the zero mean Gaussian measurement noise with covariance $\mathbf{R}$.

The observation matrix $\mathbf{H}$ extracts the positions components of the state vector $x(k)$ and is given by:

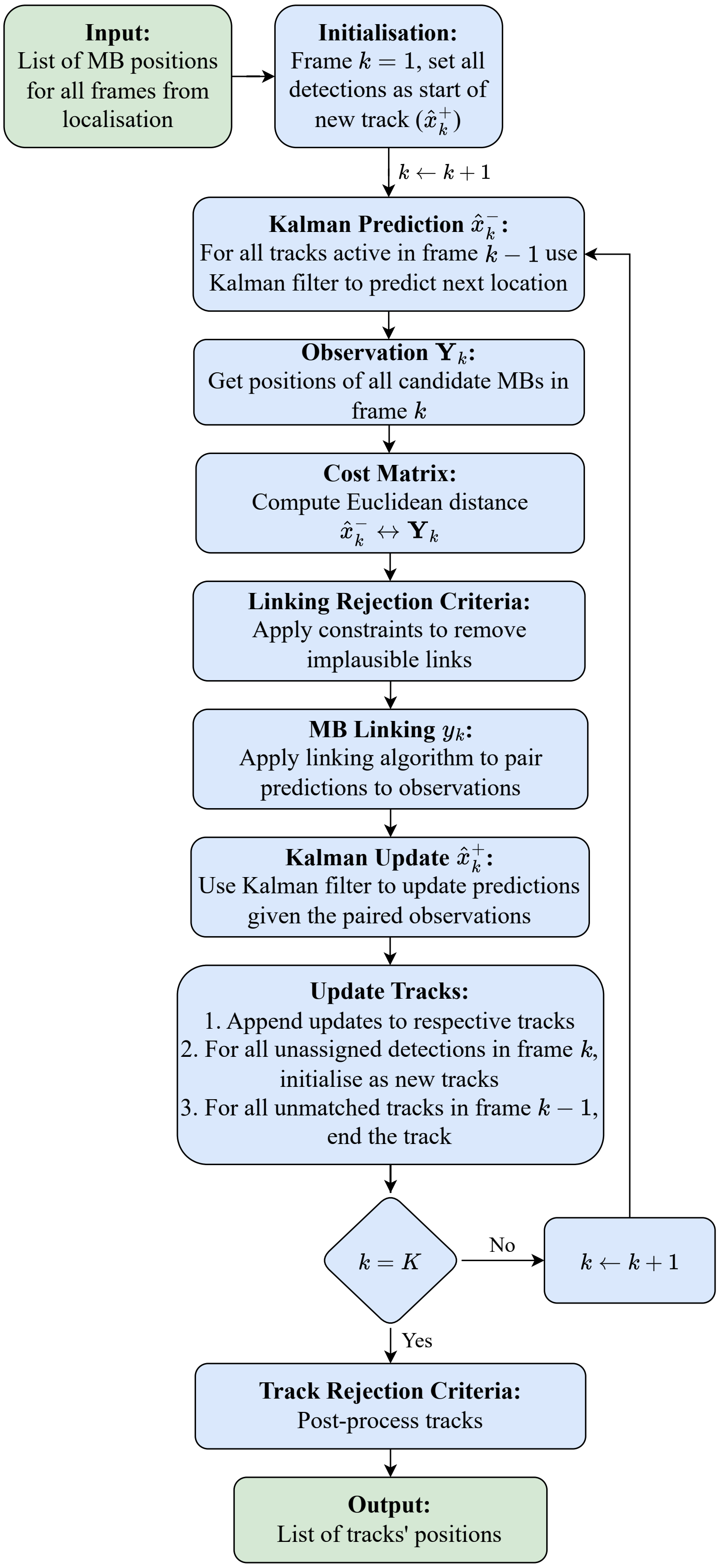



Figure 7: The proposed Kalman-filter based workflow for MB tracking. Predicted positions are linked to detections using a cost matrix with rejection criteria. Tracks are updated, initialised, or ended accordingly to the linking assignment.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5.5}$$

The third and fourth columns of $\mathbf{H}$ are set to zero since the velocity components of $x(k)$ are not directly observable in the localisation process. The noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ are tuned parameters, which are further discussed in Section 6.1.1.

This linear model provides a computationally efficient model to estimate MB motion, though it simplifies the true dynamics of MB flow. One would expect non-linear motion of MBs due to blood flow perturbations, vascular structure curvature, among other things. To represent more complex dynamics, variants such as the Extended or Unscented Kalman filters could be applied [25, 47].

### 5.4.2 Kalman Estimation Algorithm

Active MB tracks are propagated over frames using a Kalman estimation algorithm based on the model presented in the previous section. The algorithm proceeds in three stages: initialisation, prediction, and update, as illustrated in Figure 8.

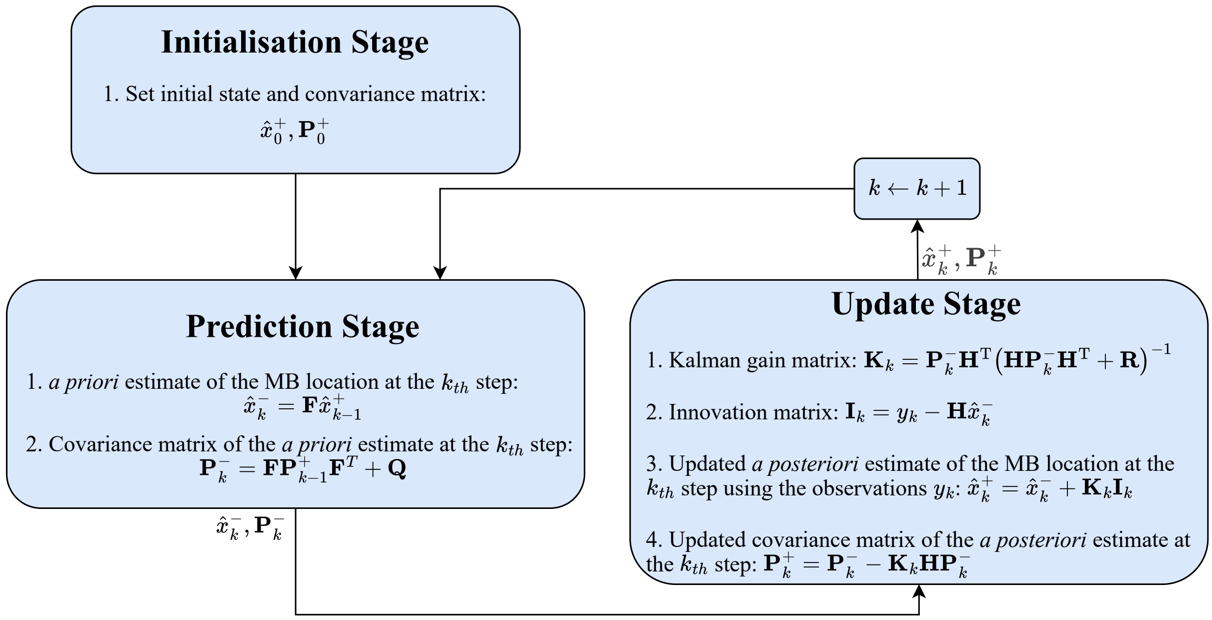



Figure 8: Flow diagram of the Kalman filtering estimation algorithm, as a component of the overall tracking workflow. Figure adapted from [19, 44].

**Initialisation**

When a new track is started from an unlinked detection, the initial state vector is constructed using the observed position and a zero-velocity assumption:

$$\hat{x}_k^+ = [x_1(0),\ x_2(0),\ 0,\ 0]^T .$$

The hat operator, $\hat{}$ , indicates an estimate of a variable. That is, $\hat{x}$ is an estimate of the state $x$. This initialisation is a simplification of the true dynamics of a MB, as a MB entering the field of view of the frame is being moved by blood flow and thus would not have zero velocity. However, since we can not assume a direction of this velocity, we assume it is reasonable to initialise the velocity component as zero.

The covariance matrix $\mathbf{P}_k^\pm \in \mathbb{R}^{4\times 4}$ encodes the uncertainty in each state dimension. During the prediction and update stages, this covariance matrix evolves over time as more observations are incorporated into the track and the uncertainty of the estimates therefore changes. The initial covariance matrix is defined as:

$$\mathbf{P}_{\mathbf{0}}^{+} = \begin{bmatrix} p_{1,1} & 0 & 0 & 0 \\ 0 & p_{2,2} & 0 & 0 \\ 0 & 0 & p_{3,3} & 0 \\ 0 & 0 & 0 & p_{4,4} \end{bmatrix},$$

where $p_{1,1}$ through $p_{4,4}$ represent the variances associated with each component of the state vector. One could make the assumption that in early track history, velocity is less certain than position due to the lack of motion information and could set $p_{1,1} = p_{2,2}$, $p_{3,3} = p_{4,4}$, and $p_{1,1} < p_{3,3}$ to reflect this. Such parameter settings would imply equal uncertainty in both the axial and lateral directions for position and velocity. These values are treated as tunable parameters and are further discussed in Section 6.1.1.

**Prediction**

The state of a MB is predicted from the previous estimate using the constant-velocity model, as was described in Section 5.4.1:

$$\hat{x}_k^- = \mathbf{F}\hat{x}_{k-1}^+,$$

where $\mathbf{F}$ is as defined in (5.3). Given this estimation, the uncertainty associated with the predicted state is adjusted accordingly via the covariance matrix:

$$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}^+\mathbf{F}^T + \mathbf{Q},$$

where $\mathbf{Q} \in \mathbb{R}^{4\times4}$ is the process noise covariance matrix, representing the uncertainty in the prediction step. This corresponds to the Gaussian noise term $\epsilon(k)$ as was defined in (5.2). It is chosen to be diagonal:

$$\mathbf{Q} = \begin{bmatrix} \sigma_\epsilon^2 & 0 & 0 & 0 \\ 0 & \sigma_\epsilon^2 & 0 & 0 \\ 0 & 0 & \sigma_\epsilon^2 & 0 \\ 0 & 0 & 0 & \sigma_\epsilon^2 \end{bmatrix}.$$

The value $\sigma_\epsilon$ is treated as a tunable parameter and is further discussed in Section 6.1.1.

**Update**

If the predicted state $\hat{x}_k^-$ is linked to a detection $y_k$ in frame $k$, an update stage is applied to refine the estimate of the true MB position given both the prediction and the observation. First, the Kalman gain matrix is computed:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right)^{-1},$$

where $\mathbf{H}$ is as defined in (5.5). The Kalman gain acts as a weighting factor, determining the relative influence of the prediction and the observation in the state update. $\mathbf{R} \in \mathbb{R}^{2\times2}$ is the measurement noise covariance matrix, representing the uncertainty in the localisation step. This corresponds to the Gaussian noise term $\nu(k)$ as was defined in (5.4). It is chosen to be diagonal:

$$\mathbf{R} = \begin{bmatrix} \sigma_\nu^2 & 0 \\ 0 & \sigma_\nu^2 \end{bmatrix}.$$

The value $\sigma_\nu$ is treated as a tunable parameter and is further discussed in Section 6.1.1.

The innovation matrix is calculated as:

$$\mathbf{I}_k = y_k - \mathbf{H}\hat{x}_k^-.$$

This can be considered as the residual between the actual measurement and the predicted observation based on the prior state.

The updated position, $\hat{x}_k^+$ is computed given the predicted location, the Kalman gain matrix, and the innovation matrix:

$$\hat{x}_k^+ = \hat{x}_k^- + \mathbf{K}_k \mathbf{I}_k.$$

The updated state is taken as the best estimate of the true MB position at frame $k$, and is what is used as the state vector. Lastly, the covariance matrix is adjusted to reflect the uncertainty in this updated state:

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H} \mathbf{P}_k^-.$$

For a visual interpretation of the prediction and update stages, Figure 9 demonstrates how an updated estimated state is predicted to the next frame and is then updated given its paired detection. The solid lines represent actual track movement between the updated positions, with dashed lines presenting the path to the predicted position.

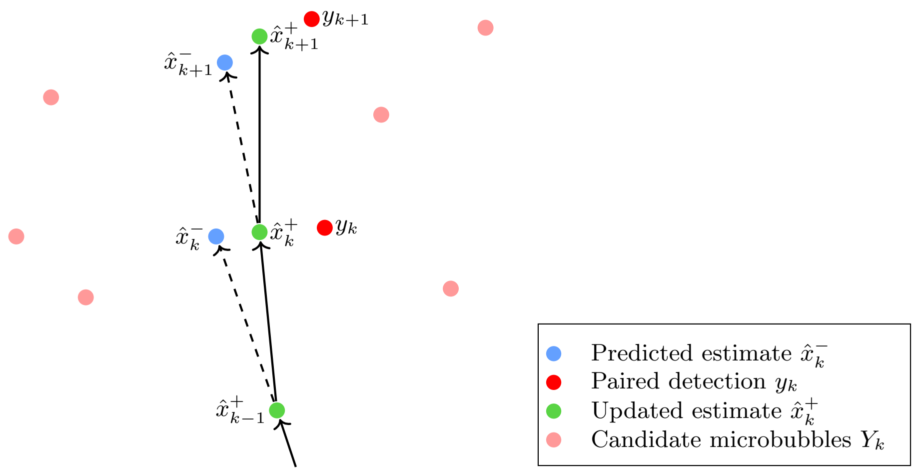



Figure 9: Kalman filter tracking process illustrated over three frames. At each frame, a predicted MB position $\hat{x}_k^-$ (blue) is linked to a detection $y_k$ (red) to produce an updated estimate $\hat{x}_k^+$ (green). Dashed arrows indicate the prediction steps, while solid lines show the updated track. Faded red markers show candidate MB detections in the field of view, $Y_k$, which could have been linked to the predictions.

These prediction and update stages are repeated for each active track across all frames in the dataset. The following section will explain how a given prediction $\hat{x}_k^-$ is paired to a detection $y_k$.

### 5.5 Microbubble Linking

At this stage, we aim to find the optimal one-to-one assignments of predicted MBs locations to the detections in a given frame. The combination of many such links form a track. The absence of a link between predictions and detections can indicate either the end of a track or the start of a new one. In our notation, we link the predicted locations of active tracks, $\hat{x}_k^-$, to all detections in frame $k$, $Y_k$, yielding a linked subset $y_k \subseteq Y_k$ containing only the detections assigned to $\hat{x}_k^-$.

The linear assignment problem seeks the optimal matching between $m$ row items and $n$ column items, given a cost matrix $\mathbf{C} \in \mathbb{R}^{m \times n}$, where $\mathbf{C}_{ij}$ is the cost of assigning row $i$ to column $j$. This corresponds to the objective:

$$\min_{\mathbf{M}} \sum_{(i,j) \in \mathbf{M}} \mathbf{C}_{ij},$$

where $\mathbf{M}$ is a matrix of non-overlapping pairs, also known as matchings. A matching provides a one-to-one pairing where no two pairs share the same row or same column. When $m = n$ and every item is matched exactly once, it is known as a perfect matching. When $m \neq n$, some items will remain unmatched and the result is known as a partial matching.

In our application, the cost matrix $\mathbf{C}$ is the pairwise Euclidean distance between all active tracks $\hat{x}_k^-$ to all detections in the same frame $Y_k$. We do not expect every predicted MB position (row item in $\mathbf{C}$) to have a true match with a detection (column item in $\mathbf{C}$). Therefore, we wish to obtain the optimal partial assignment solution to the linear assignment problem. Even if $m = n$ for a given frame, a perfect matching is not expected because MB tracks may begin or end in any frame (i.e. tracks do not continue forever). The linking rejection criteria in Section 5.6 are implemented by setting the corresponding distance in the cost matrix $\mathbf{C}$ to infinity, thus removing implausible links from matching in the minimisation process.

One could also consider this problem in graph terms, where the equivalent set-up is to find the minimum-weight partial matching of a weighted bipartite graph. In such a graph, the left nodes represent the predicted positions of MBs, the right nodes represent the detected MB positions, and the edges correspond to possible links weighted by their cost (i.e. distance). Setting a given cost to infinity to apply linking rejections is equivalent to removing the corresponding edge between those nodes.

### 5.5.1 Solving the Linear Assignment Problem

To solve the linear assignment problem, the matchpairs.m function in MATLAB was implemented as it allows for partial matching. This function applies the method described by Duff and Koster [15]. It minimises the total cost of assigning rows to columns based on a given cost matrix. The total cost of an assignment is computed as:

$$\text{Total Cost} = \sum_{i=1}^{P} \mathbf{C}\left(\mathbf{M}(i,1), \mathbf{M}(i,2)\right) + \text{costUnmatched} \cdot (m + n - 2p)\,.$$

Here, $\mathbf{C}$ is the cost matrix as previously defined. The matrix $\mathbf{M}$ contains $P$ matched pairs, with $\mathbf{M}(i,1)$ and $\mathbf{M}(i,2)$ corresponding to the row and column indices of the $i$-th assignment. The term $(m + n - 2p)$ counts the total number of unmatched rows and columns. Each unmatched row or column incurs a penalty of costUnmatched, allowing for partial assignments. We set costUnmatched empirically to $L_{\max}$ as it is defined in the following section. This cost penalises links that exceed expected physical movement, effectively discouraging implausible assignments.

The presented method is not the only way to solve the linear assignment problem. The Hungarian (Munkres) Algorithm, discussed in Section 5.2 and Appendix B, is a common approach. In the development process it was found the Duff and Koster method achieved an average matching time of 0.08 seconds per frame, representing a 35.5x speed-up compared to the 2.52 second per frame of the Hungarian method. Given the many trials and parameter tuning iterations described in future sections, we chose the Duff and Koster method for computational efficiency. In our implementation, switching between these two methods is as simple as changing a single function call to perform the matching, allowing for the straightforward comparison of linking algorithms.

## 5.6 Linking Rejection Criteria

To remove implausible links between predicted estimates, $\hat{x}_k^-$, to observations in the frame, $Y_k$, a series of rejection criteria are applied prior to the MB linking stage. Each criterion is representative of a physical constraint of the problem and aims to eliminate unrealistic associations between localisations. The rejection workflow is designed to be modular, allowing each criteria to be enabled or disabled independently. This provides flexibility in tuning of the tracking pipeline, and can inspect the individual impact of each criterion. Velocity (Section 5.6.1) and short link (Section 5.6.2) rejections are always applied while angle (Section 5.6.3) and acceleration (Section 5.6.4) rejections are optional.

In addition to improving the physical plausibility of the associations, the rejection criteria help reduce the number of candidate links, leading to a sparser cost matrix and simplifying the optimisation during tracking. Further, by filtering out the implausible links these criteria help reduce incorrect tracks with respect to the ground truth information and improve the objective quality of the tracking. The rejection criteria applied in the proposed workflow are explained in the following sections.

### 5.6.1 Velocity Rejection

The velocity-based rejection criterion restricts the maximum speed that a MB can travel between consecutive frames. Any proposed link between two localisations that would imply a velocity greater than what would be physically plausible, it gets rejected. Intuitively, it follows that it is implausible for a MB detected in one corner of an image to truthfully appear in the opposite corner in the next frame.

According to the ULTRA-SR GT data [29], the maximum velocity magnitude of a link is $v_{\max} = 80$ mm/s. Converting this pixel and frame units using the simulations parameters in Table 1 gives $v_{\max} = 40$ pixels/frame. Since the tracking pipelines operated on a per frame basis, the maximum linking distance between two frames is therefore 40 pixels, which we denote $L_{\max}$. This constraint is visualised in Figure 10, where candidate MBs in the next frame are rejected when they are outside of the $L_{\max}$ radial distance from the current MB's predicted position.

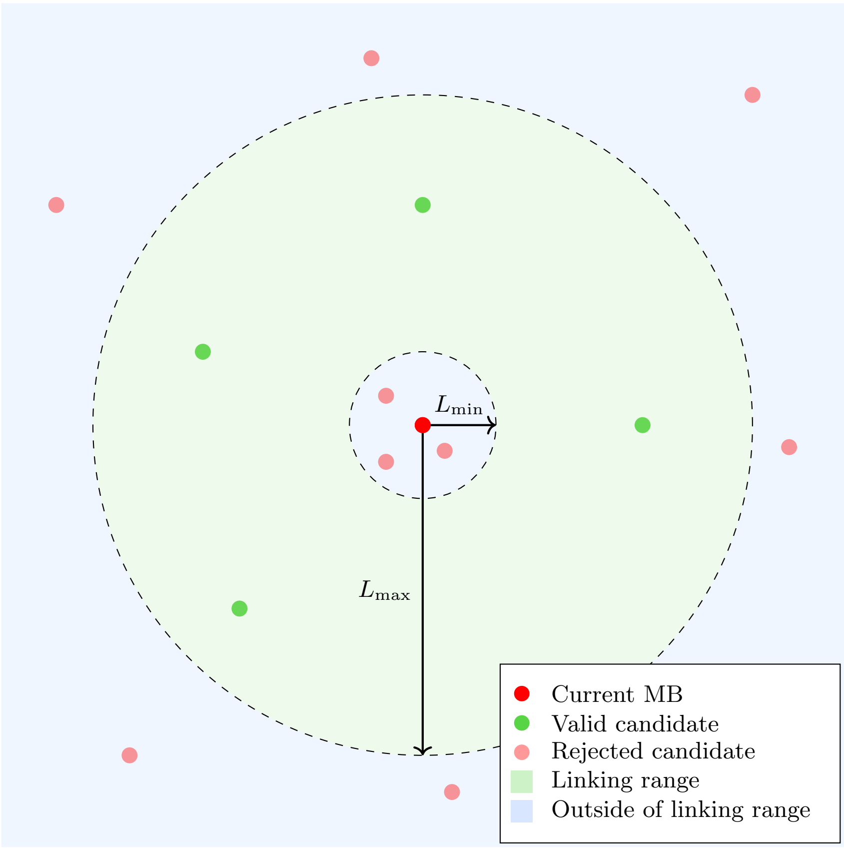
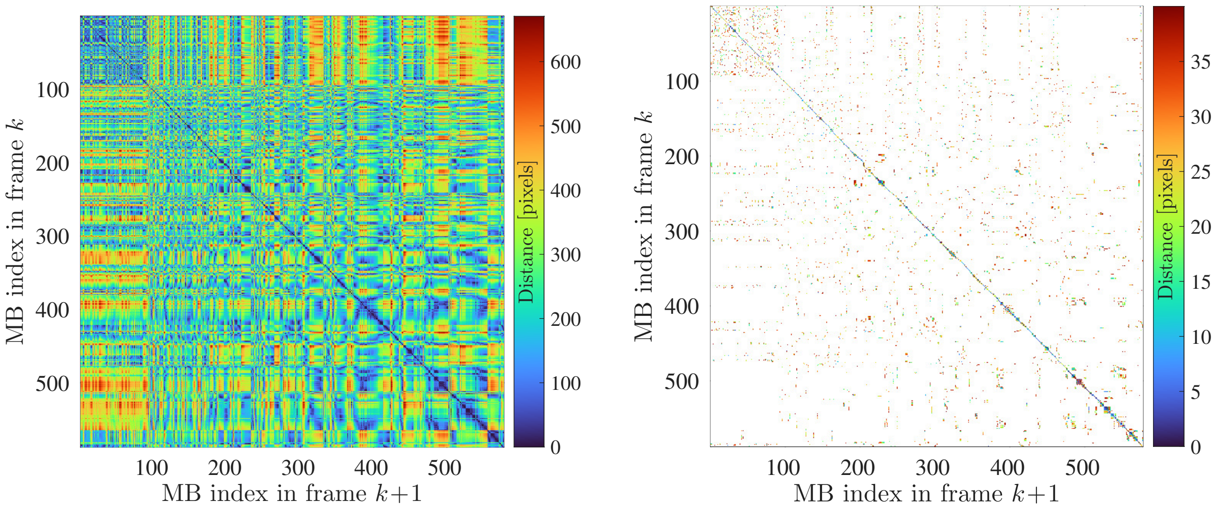



Figure 10: Illustration of the velocity and short link-based rejection criteria. Candidate MBs must lie in the annular region defined by the minimum and maximum distances, $L_{\min}$ and $L_{\max}$ respectively, to be considered valid. These bounds represent the physical constraints on velocity and displacement of MB motion per frame. Distances are exaggerated for visualisation purposes.

To implement this rejection criterion, we create a second Euclidean distance cost matrix between the most recent updated position of the previous frame's active tracks, $\hat{x}^{+}_{k-1}$, and the list of candidate MBs in the current frame, $Y_k$. Proposed links in this secondary matrix with distances greater than the $L_{\max}$ threshold are identified and used to mask the main cost matrix, setting the corresponding entries to infinity and rejecting those potential associations in the linking assignment.

The main distance cost matrix $\mathbf{C}$ before and after the application of the the velocity-based constraint is visualised in Figure 11. In this figure the MBs detections in frames $k$ and $k+1$

are arbitrarily indexed and the pairwise distance is calculated. Figure 11 (a) shows the full cost matrix without thresholding, and contains many associations which are over $L_{\max}$. The threshold is applied in Figure 11 (b), and we observe how the cost matrix becomes significantly more sparse and rejects a majority of the possible links. This step will simplify the later linking assignment. Figure 11 represents the associations between frames 1 and 2 in the ULTRA-SR dataset. Note that the cost matrix is not required to be square as the number of active tracks in frame $k$ and observations in frame $k+1$ does not need to be equal.




(a) Full cost matrix showing all pairwise distances between MB positions in consecutive frames.

(b) Thresholded cost matrix after applying a maximum velocity constraint, removing implausible pairings.

Figure 11: Pairwise distance cost matrices used for MB tracking. (a) shows all pairwise distances between candidate localisations in frames $k$ and $k+1$. (b) displays the matrix after applying a maximum velocity threshold, removing implausible pairings and significantly reducing the complexity of optimisation. White entries in (b) correspond to infinite costs (i.e. disallowed pairings).

#### 5.6.2 Short Link Rejection

The short link-based rejection criteria implements the physical assumption that MBs should not remain stationary in consecutive frames. Recall that the localisations represent blood flow within a vascular network, therefore it is expected that MBs have at least some motion between frames.

Analysis of the ground truth data shows that the minimum observed linking distance between consecutive frames is approximately 2.55 pixels (or 1 mm). However, due to localisation uncertainty and the inherent noise in the detection process, we relax this constraint empirically. The minimum linking distance parameter, $L_{\min}$, represents the minimum allowable displacement and is set to 5-10% of the pixel size. Since all processing is being computed in pixel units, this threshold is directly set as a value in the range $L_{\min} = 0.05$–$0.1$. The short link-based rejection criteria is visualised in Figure 10, where candidate MBs in the next frame are rejected when they are within the $L_{\min}$ radial distance to the current MB's predicted position.

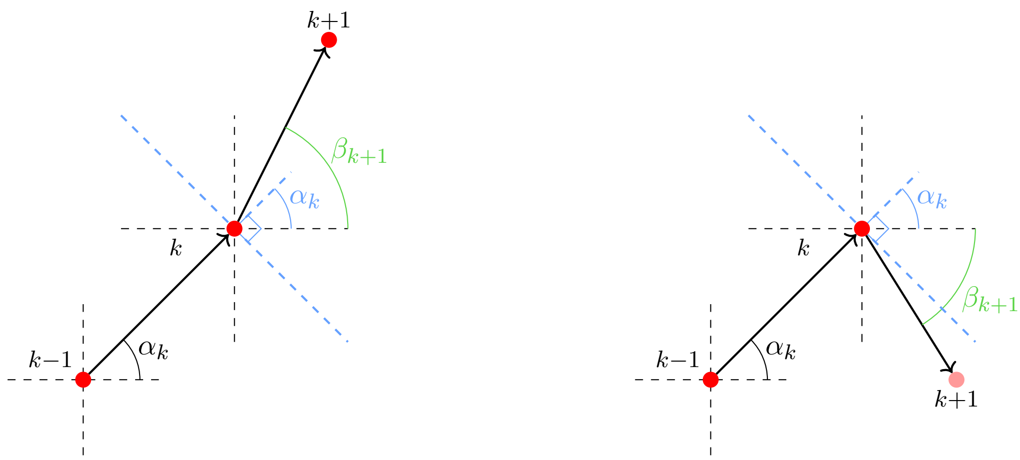
In practice, this is implemented similarly to the velocity-based rejection criterion as explained in the previous section. A second Euclidean distance matrix is computed between the corrected positions of active tracks in the previous frame $\hat{x}^+_{k-1}$ and the set of all candidate detections in the current frame $Y_k$. Links in the secondary matrix with distances below the $L_{\min}$ threshold are identified and used to mask the main cost matrix $\mathbf{C}$, setting the corresponding entries to infinity, thus rejecting those associations in the linking optimisation.

### 5.6.3 Angle Rejection

The angle-based rejection criteria restricts the allowable change in direction of a MB's trajectory in consecutive frames. The angle between the current candidate motion and the track's history is limited such that it prevents abrupt or physically implausible changes in direction. In this implementation, the maximum allowed deviation in angle is set to $\theta_{\text{thresh}} = \pi/2$, which constrains the direction of motion within $[-\pi/2, \pi/2]$ relative to the previous velocity vector. This effectively ensures that MBs do not exhibit backwards motion.

This criterion is illustrated in Figure 12, which shows valid and invalid candidate detections relative to the track's history. Let $\alpha_k$ denote the angle of motion of a MB from frame $k-1$ to $k_1$, with respect to the position in frame $k-1$ as the origin. Similarly, $\beta_{k+1}$ is the angle to a candidate in frame $k+1$, measured from the position in frame $k$ as the origin. A proposed link is accepted if the change in angle $\Delta\theta_k$ satisfies:

$$\Delta\theta_k = |\beta_{k+1} - \alpha_k| < \theta_{\text{thresh}}. \tag{5.6}$$




(a) The MB detection at frame $k{+}1$ is within the angular threshold, making the proposed track originating from frame $k{-}1$ valid.

(b) The MB detection at frame $k{+}1$ is outside the maximum threshold. This detection would not be a valid continuation of the track originating from frame $k{-}1$.

Figure 12: Angle-based rejection criterion for MB tracks where the threshold angle, $\theta_{\text{thresh}}$, is equal to $\pi/2$. Figure inspired by [44]. a) Valid MB candidate. b) Invalid MB candidate.

The angle-based rejection is implemented by evaluating the tracks with respect to candidate detections. For each active track, the most recent two positions are used to compute $\alpha_k$. From the last position, angles $\beta_{k+1}$ are computed to all localisations in the current frame, $Y_k$. Any proposed links where $\Delta\theta_k$ exceeds the threshold as defined in (5.6) is considered invalid. These links are masked out by setting the corresponding indices in the cost matrix $\mathbf{C}$ to infinity, therefore excluding them from the linking assignment. In practice, this criterion can only applied when a track has at least two prior positions in its history. To compute angles $\beta_{k+1}$ and $\alpha_k$, there must be at least two positions already. For newly initialised tracks which only contain a single detection, the angle-based rejection is skipped.

### 5.6.4 Acceleration Rejection

The movement of MBs is governed by the underlying blood flow, which imposes physiological constraints of how quickly velocity can change. In particular, we do not expect a given MB to have significant accelerations or decelerations in consecutive frames. Within the spatial and temporal resolution of a dataset, the velocity of a MB travelling through a single vessel is expected to remain relatively consistent.

To enforce these assumptions, an adaptive acceleration-based rejection criterion is applied. For each candidate link, the change in speed relative to the most recent velocity in the track's history is computed. A potential link is rejected if this change exceeds a threshold. The absolute value of acceleration is calculated as

$$a = \frac{|\,\|\vec{v}_{k+1}\| - \|\vec{v}_k\|\,|}{1/f_r} < a_{\text{thresh}}, \tag{5.7}$$

where $||\vec{v}_k||$ is most recent magnitude of velocity in a track's history, $||\vec{v}_{k+1}||$ is the magnitude of velocity between the most recent position and a candidate detection, and $f_r$ is the frame rate. A proposed link is accepted only if $a < a_{\text{thresh}}$, where $a_{\text{thresh}}$ is an adaptive threshold computed as

$$a_{\text{thresh}} = \frac{0.5 \times \bar{v}}{1/f_r}, \tag{5.8}$$

with $\bar{v}$ defined as the 20% trimmed mean values of the velocities measured in each frame of a the track [24], implemented with the MATLAB function trimmean.m. This follows the methodology presented by Tang et al. [44], and removes the influence of outlier velocities.

Since the magnitudes of velocities are being used, this criterion will eliminate potential links that would imply motion that is either too fast or too slow given prior track information. This concept is visualised in Figure 13, where candidate MBs are either valid or invalid depending on the track's velocity history and implied relative speed change.

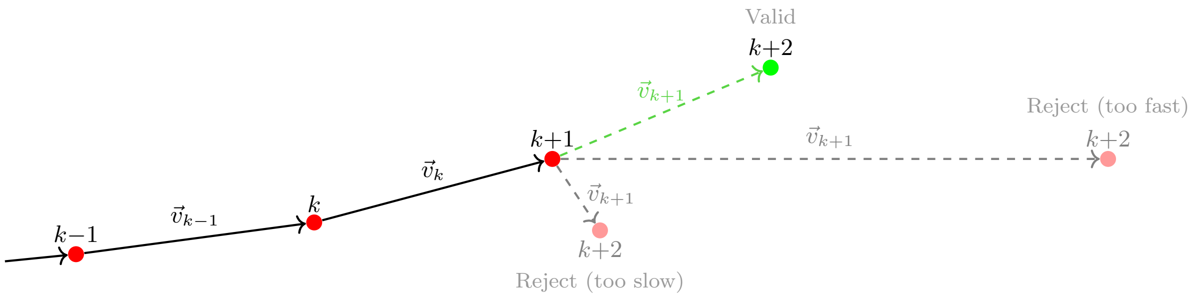



Figure 13: Acceleration-based rejection criterion applied to candidate MB detections in frame $k{+}2$. The velocity of the track up to frame $k{+}1$ informs the adaptive threshold for acceptable motion. Detections that induce excessive acceleration or deceleration relative to prior track dynamics are rejected.

As with the angle-based rejection criterion, the acceleration constraint is implemented by evaluating active tracks with respect to all candidate detections in the current frame. For each track, the two most recent positions are used to compute $\vec{v}_k$, and candidate velocities $\vec{v}_{k+1}$ are computed with respect to each candidate detection. Any proposed link for which the acceleration exceeds $a_{\text{thresh}}$ is rejected by setting the corresponding entry in the cost matrix $\mathbf{C}$ to infinity.

To ensure reliability, the acceleration constraint is only applied to tracks with sufficient motion history. We empirically set the minimum track length to at least five positions before the constraint is implemented. This avoids being overly restrictive on newly initialised tracks whose Kalman filter estimates may still be stabilising due to the uncertainty in their state covariance.

## 5.7 Track Management

For completeness, this section will explain how links are compiled into MB tracks, informing on the MATLAB implementation. This corresponds to the Update Tracks stage in Figure 7.

A track refers to the sequence of links which through the proposed method are believed to be of the same MB over time. With a slight abuse of notation, we denote the full set of all tracks by:

$$\mathbf{\Gamma} = \left\{ \gamma^{(1)}, \gamma^{(2)}, \ldots, \gamma^{(D)} \right\},$$

where $D$ is the total number of tracks. Each track $\gamma^{(d)}$ consists of a sequence of updated state vectors for that MB:

$$\gamma^{(d)} = \left\{ \gamma_1^{(d)}, \gamma_2^{(d)}, \ldots, \gamma_{N_d}^{(d)} \right\},$$

where $\gamma_i^{(d)}$ denotes the $i$-th updated state vector in the track. Here, $N_d$ is the total number of detections assigned to track $\gamma^{(d)}$.

It is important to consider that each track has its own local indexing. We denote $\gamma_i^{(d)} = \hat{x}^+_{k_i^d}$, where $\hat{x}^+_{k_i^d}$ is the updated Kalman state vector at global frame index $k_i^d$, and $i$ is the local index within track $\gamma_i^{(d)}$.

For each track $\gamma^{(d)}$, the position and instantaneous velocity vectors at the $i$-th position in the track can be extracted from the full state vector $\gamma_i^{(d)}$:

$$\vec{r}_i = [x_{1_i}, x_{2_i}], \quad \vec{v}_i = [dx_{1_i}, dx_{2_i}],$$

where the state vector components are as described in (5.1) and are appropriately associated with that track. The mean velocity of a track $\gamma^{(d)}$ can be expressed as:

$$\bar{\vec{v}} = \frac{\vec{r}_{N_d} - \vec{r}_i}{N_d},$$

where $\vec{r}$ corresponds to the appropriate position vectors for that track. These quantities are applied in post-processing steps as described in Section 5.8.

New tracks are initialised as described in Section 5.4.2. In the first frame, all detections are initialised as state vectors with zero velocity and are added to the set of all tracks $\mathbf{\Gamma}$. For all subsequent frames, new tracks are similarly initialised when a detection in $Y_k$ is not paired with a prediction $\hat{x}_k^-$. These unpaired detections are initialised as new tracks which are appended to $\mathbf{\Gamma}$, and are marked as active for consideration in the following frame. If a prediction $\hat{x}_k^-$ is linked to a detection $y_k$, the resulting updated state $\hat{x}_k^+$ is appended to the corresponding track $\gamma^{(d)}$. This extends the track and is considered active for the next frame. If a prediction $\hat{x}_k^-$ is not paired to a detection in $Y_k$, the track is considered to have ended in frame $k-1$. The prediction is not appended into the associated track and the track is marked inactive, meaning it will not be considered for linking in future frames.

## 5.8 Post-Processing Track Rejection

After the Kalman filtering and linking workflow is completed for all frames, a set of tracks $\mathbf{\Gamma}$ is obtained. However, after this process false MB locations and trajectories may still exist due to imperfect linking or noisy detections. As displayed in Figure 7, the final step in the pipeline involves applying the following track post-processing rejection criteria can be applied to improve the robustness and accuracy of the resulting tracks. Angle (Section 5.8.1) and acceleration (Section 5.8.2) post-processing are considered optional, while short track rejection (Section 5.8.3) is always applied. Importantly, angle and acceleration rejections are not applied redundantly throughout the pipeline. If they are used during the linking stage (Section 5.6), they would not be used as a post-processing step and vice-versa.

### 5.8.1 Angle Rejection

This criterion is similar in principle to the linking angle-based rejection presented in Section 5.6.3, but is applied as a post-processing to individual tracks rather than frame-to-frame linking candidates. In post-processing, each track $\gamma^{(d)}$ is iterated through, and the angle between consecutive segments is computed. If the angle at index $i$ of the track exceeds the threshold as defined in (5.6), the track is split at this index.

The first track segment $\left\{\gamma_1^{(d)}, \ldots, \gamma_i^{(d)}\right\}$ remains under the original track index $d$, while the second segment $\left\{\gamma_{i+1}^{(d)}, \ldots, \gamma_{N_d}^{(d)}\right\}$ is added to the set $\mathbf{\Gamma}$ as a new track. This new segment is then re-evaluated by the same rejection criterion as it may contain further angle violations.

### 5.8.2 Acceleration Rejection

This criterion is the similar in principle to the linking acceleration-based rejection presented in Seciton 5.6.4, but again is applied as a post-processing to individual tracks rather in the linking process. As in the previous section, each track $\gamma^{(d)}$ is iterated through and if the acceleration constraint defined in (5.7) is exceeded at index $i$ the track is split. The resulting segments are added to the set $\mathbf{\Gamma}$, following the same procedure as in the track angle-based rejection.

The acceleration threshold is computed using the trimmed mean as described in (5.8) using the entire track $\gamma^{(d)}$. It is assumed that tracks shorter than length of 10 do not have sufficient motion history for this criterion to be meaningful and are skipped in the iteration process.

In practice, it was found that this rejection criterion was overly sensitive even under relaxed thresholds. To mitigate over-segmentation and excessive computing time, a counter was implemented such that each track may be split at most three times. Once this limit is reached, the remaining segments are retained in $\mathbf{\Gamma}$ and the algorithm proceeds to the next track.

### 5.8.3 Short Track Rejection

The simplest tracking rejection criterion involves discarding tracks which are considered too short to be meaningful. The minimum track length is empirically set to three, meaning that any tracks that are simply a single link is removed from the set of all tracks:

$$\mathbf{\Gamma} \leftarrow \mathbf{\Gamma} \setminus \left\{\gamma^{(d)} \mid N_d < 3\right\}.$$

It is assumed that any tracks shorter than this threshold are the result of noise or represent an unreliable partial trajectory of a MB. It is applied to eliminate false positive links that likely do not correspond to the true MB motion. This short track rejection is applied after the angle and acceleration post-processing steps as those criteria are likely to create small tracks.

# 6 Results and Discussion

This section presents the results of the our proposed Kalman filter tracking approach. First the development process is explored, where parameter tuning and design decisions are discussed. Secondly, the final model is evaluated, beginning with the simulated localisation data before applying it to actual localisation outputs.

## 6.1 Development Process

The development process focuses on maximising the tracking metrics described in Section 5.1. This section outlines the process of building and refining the tracking method, rather than the final performance evaluation, which is reported separately in Section 6.2. All experiments conducted in the development process were performed using the GT raw data from the synthetic dataset (Section 3). This was to ensure that tracking performance could be assessed independently of localisation errors. A series of trials were carried out, each testing parameter choices or algorithm modifications, with the goal of developing the most robust tracker possible. Of note, the velocity rejection (Section 5.6.1), short link rejection (Section 5.6.2), and track post-processing rejection short track rejection (Section 5.8.3), were always applied, as these are simple constraints assumed to always hold. The angle and acceleration-based criteria were treated as optional or additional refinements and are evaluated separately in Sections 6.1.4 and 6.1.5.

The subsections that follow present these experiments in the order in which they were conducted, as each stage informs the next. In the tables that follow, the highlighted row indicates the optimal configuration derived from each trial, representing the design choices carried forward into the next stage of development.

### 6.1.1 Tuning the Kalman Filter

The parameters of the Kalman filter are described in Section 5.4.2. Recall that the model contains three covariance matrices: $\mathbf{P}_\mathbf{0}^+$ (initial state uncertainty), $\mathbf{Q}$ (process noise), and $\mathbf{R}$ (measurement noise). For simplicity, each matrix was set to be diagonal with identical values along the diagonal. Therefore, the tuning process simplifies to selecting three parameters corresponding to these diagonals. The objective of this stage was to identify the set of parameters that maximise the tracking metrics.

A grid search approach was applied, chosen for its simplicity in implementation and interpretable results. Parameter values were varied across a logarithmic grid spanning $10^{-4}$ to $10^2$, which were assumed to be the extremes of potential tuning values. Every parameter combination was evaluated, and the results were compared according to the tracking metrics.

Selected results are presented in Table 6. In this and all subsequent tables, $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{R}$ represent the values along their respective diagonals. It is important to emphasise that the localisation metrics here do not indicate changes to the underlying dataset, as the localisation quality of the input data remains unchanged in the tracking process. Instead, these values in the table represent the effective localisation metrics, which are computed only on the subset of detections that remain in the tracking results after the application of the Kalman filter and track rejection. Effective localisation values decrease either because updated state estimates are shifted outside the evaluation search radius (4.2), or because tracks are discarded during short track rejection (Section 5.8.3). Therefore, these metrics reflect the proportion of ground truth information retained in the tracking output rather than the input data.

When $\mathbf{R}$ is very small, as seen in Trials 1 and 2, the filter places almost all confidence in the measurements rather than the predictions in the update stage of the model. These trials show that with a small value of $\mathbf{R}$, the choices for $\mathbf{P}$ and $\mathbf{Q}$ have a lesser impact on the metrics. Such parameters yielded results with $J_{\text{map}} < 0$, which indicates there is more erroneous pairing distance than correct, which is of course something that should be avoided for a tracking method.

Trial 3 illustrates a more balanced result, where with a more moderate value for $\mathbf{R}$ all tracking metrics are increased, with a positive $J_{\text{map}}$. We do however, start to see a decrease

| Trial | Parameter | | | Tracking Metric | | | Effective Localisation Metric | |
|---|---|---|---|---|---|---|---|---|
| | **P** | **Q** | **R** | **Precision (%)** | **Recall (%)** | $J_{\mathbf{map}}$ | **Precision (%)** | **Recall (%)** |
| 1 | 100 | 100 | 0.001 | 73.31 | 73.97 | -0.1450 | 100 | 99.93 |
| 2 | 1 | 1 | 0.001 | 73.20 | 73.85 | -0.1470 | 100 | 99.93 |
| 3 | 10 | 1 | 1 | 79.06 | 79.51 | 0.0345 | 88.69 | 88.62 |
| 4 | 1 | 1 | 10 | 87.07 | 87.72 | 0.2893 | 54.74 | 54.68 |

Table 6: Selected results of the grid search tuning of the Kalman filter. The highlighted row indicates the parameter set used for subsequent analyses, as it achieved the best tracking metrics. Localisation values represent effective localisation metrics, i.e., the subset of detections retained in the final tracking results.

in effective localisation metrics as a result of increasing the tracking metrics. This is likely a result of the Kalman update stage placing more weight on the predictions and shifting the final positions being outside of the evaluation radius.

Trial 4 achieves the strongest overall tracking performance, with all tracking metrics maximised and is therefore selected as the best tuned parameter set. This comes at a the cost of a significant reduction in effective localisation metrics. This trade-off is not ideal, but for the purposes of developing a robust tracking method, priority was given to maximising the tracking metrics. For subsequent experiments, the tuned values selected were $\mathbf{P}_{\mathbf{0}}^{+} = \text{diag}(1)$, $\mathbf{Q} = \text{diag}(1)$, and $\mathbf{R} = \text{diag}(10)$ and will be called the baseline Kalman filter model. These values are consistent in order of magnitude with those presented by Tang et al. [44], though no value for $\mathbf{P}_{\mathbf{0}}^{+}$ was reported in their work.

### 6.1.2 Using Updated or Observed Position

This investigation was performed as a validation of the Kalman filter implementation itself, designed to consider whether one should simply use the observed position (i.e. the localisation) rather than using the updated state. In the implementation described up to now, the updated state estimate $\hat{x}_k^+$ is used to record position and is included in the relevant track information $\gamma_k$. To test this choice, a variant method was implemented in which the linked detections $y_k$ were saved directly to the tracks. The Kalman filter still performed predictions $\hat{x}_k^-$, but these were only used in the linking assignment. A grid search procedure, as described in Section 6.1.1, was applied to both versions to ensure a fair comparison.

The results are summarised in Table 7. Unsurprisingly, since tracker development was performed on GT data, using the observed positions yielded higher effective localisation metrics than using the updated state. However, the corresponding $J_{\text{map}}$ is negative, indicating that more incorrect error distance is obtained than correct links. Conversely, the updated position method has a positive $J_{\text{map}}$ metric and considerably higher tracking precision and recall, making it the better choice. This confirms that Kalman updating improves trajectory reconstruction over simply using Kalman predictions as part of a linking strategy, and one would expect that its advantage to be even greater when working with noisy and imperfect localisations.

| Model Version | Parameter | | | Tracking Metric | | | Effective Localisation Metric | |
|---|---|---|---|---|---|---|---|---|
| | **P** | **Q** | **R** | **Precision (%)** | **Recall (%)** | $J_{\mathbf{map}}$ | **Precision (%)** | **Recall (%)** |
| Updated Position ($\hat{x}_k^+$) | 1 | 1 | 10 | 87.07 | 87.72 | 0.2893 | 54.74 | 54.68 |
| Observed Position ($y_k$) | 10 | 0.01 | 0.0001 | 75.57 | 76.24 | -0.0903 | 100 | 99.93 |

Table 7: Tracking and localisation performance when saving the updated state estimate or the observed measurement position. The highlighted row corresponds to the model version with superior tracking performance.

### 6.1.3 Comparing Against Baseline Methods

As a further validation, the tuned Kalman filter was compared against two baseline methods: Nearest Neighbour (NN) and Hungarian algorithm, as described in Section 5.2 and Appendix B. To ensure a fair comparison, only the minimal set of rejection criteria were applied to all three methods: Velocity and short link-based rejections during linking, and short track rejection in post-processing. Since identical rejection criteria and perfect localisations were used, the differences in performance should reflect the tracking model itself.

Table 8 presents the results. The Kalman filter clearly outperforms both baseline methods across all tracking metrics, and is the only method which achieves a positive $J_{\text{map}}$. The NN and Hungarian algorithm methods yield significantly lower tracking precision and recall, with negative $J_{\text{map}}$ values indicating more incorrect link distance than correct. The baseline methods retain nearly all detections, reflected in their high effective localisation recall. However, despite retaining more information in their tracking outputs, the negative $J_{\text{map}}$ values indicate these methods fail to use it successfully in tracking. By comparison, the Kalman filter sacrifices some detections but produces substantially higher quality links.

| | **Tracking Metric** | | | **Effective Localisation Metric** | |
|---|---|---|---|---|---|
| **Method** | **Precision (%)** | **Recall (%)** | $J_{\text{map}}$ | **Precision (%)** | **Recall (%)** |
| Baseline Kalman Filter | 87.07 | 87.72 | 0.2893 | 54.74 | 54.68 |
| Nearest Neighbour | 61.73 | 62.76 | -0.4932 | 100 | 98.07 |
| Hungarian Algorithm | 57.19 | 58.49 | -0.5349 | 100 | 99.97 |

Table 8: Tracking performance of the Kalman filter compared with baseline methods (Nearest Neighbour and Hungarian algorithm). All methods were evaluated using identical rejection criteria and ground truth localisations.

Interestingly, the simple NN implementation outperforms the Hungarian algorithm, despite the latter producing a global minimum of the cost matrix while the NN does not. As stated in Section 5.2, these two methods minimise the distances between detections across frames $k-1$ to $k$ without consideration of motion dynamics. This means that minimising the spatial distance across frames does not guarantee true trajectories are recovered, and the global minimum assignment causes more incorrect pairings since the closest detection is not necessarily the correct continuation of a track.

In contrast, the Kalman filter-based approach incorporates motion prediction and creates links between estimated positions $\hat{x}_k^-$ and detections $y_k$ within the same frame. The superior tracking metrics suggest that incorporating a motion model better captures the underlying MB trajectories than methods that simply rely on spatial proximity. It should be noted that this outcome reflects the characteristics in this synthetic dataset and performance comparison between the methods could vary depending on differing localisation quality, MB density, and imaging parameters of the input data.

### 6.1.4 Linking Rejections

The purpose of this experiment was to investigate whether applying additional rejection criteria during the linking process would improve overall tracking performance. This follows the intuition that if implausible links are prevented from being formed in the first place, the resulting tracks should be more accurate and require less post-processing. This strategy aims to create tracks which are more reliably built from the outset. In all trials, the velocity (Section 5.6.1) and short link-based (Section 5.6.2) rejections were applied as defaults. Further trials tested the angle and acceleration-based rejections, as they are described in Section 5.6.3 and Section 5.6.4 respectively, both individually and in combination.

Interestingly, as shown in Table 9, applying additional rejection constraints in the linking process consistently reduced the tracking metrics relative to the baseline configuration. This outcome was contrary to initial expectations. Trials in which the acceleration-based rejection criteria was applied showed a significant drop in recall and resulted in negative $J_{\text{map}}$ scores. While Trial 4 (combined angle and acceleration rejection) did show an improvement in precision over the baseline, this gain is outweighed when considering the substantial reduction in $J_{\text{map}}$, indicating worse overall tracking quality.

| Trial | Linking Rejection Criteria | | Tracking Metric | | |
|---|---|---|---|---|---|
| | **Angle** | **Acceleration** | **Precision (%)** | **Recall (%)** | $\boldsymbol{J_{\text{map}}}$ |
| 1 | False | False | 87.07 | 87.72 | 0.2893 |
| 2 | True | False | 81.69 | 80.69 | 0.0747 |
| 3 | False | True | 82.86 | 68.49 | -0.4157 |
| 4 | True | True | 87.46 | 73.65 | -0.1886 |

Table 9: Tracking performance with combinations of angle and acceleration-based rejections applied as part of the linking process. The highlighted row corresponds to the baseline configuration, where only velocity and short-link rejections were applied, which yields the best results.

To investigate whether the reduction in tracking performance was due to overly strict thresholds, additional tests were performed with relaxed parameters. That is, in (5.6) the $\theta_{\text{thresh}}$ was increased and in (5.8) the factor of 0.5 in $a_{\text{thresh}}$ was increased. However, even under these less restrictive conditions, none of the configurations improved upon the baseline.

These results suggest that, while conceptually appealing, such rejections are not well suited to be applied during the linking process. It is assumed that they are too restrictive at this stage and true continuations of tracks may get rejected early in the process, which propagates errors into the following iterations of the algorithm. Further, for an acceleration-based constraint to be valuable, a track should be of sufficient length to have a significant amount of motion history. Short tracks contain less velocity information and could yield ineffective thresholds. This interpretation is supported by the observed increase in the number of shorter, fragmented tracks when these criteria were applied. Based on these findings, only velocity and short link-based rejections were retained going forward. Angle and acceleration-based rejections applied as post-processing steps are discussed in the following section.

### 6.1.5 Track Post-Processing Rejections

This section details the results of applying additional rejection criteria as a post-processing step after the tracks have been formed, as described in Section 5.8. The goal of this step in the pipeline is to improve tracking quality by splitting tracks at links which are deemed implausible by the rejection criteria. Since post-processing operates on complete tracks, these methods can make use of longer motion history to enforce more informed rejections than are available during frame-to-frame linking rejections as seen in Section 6.1.4.

First, the angle-based criterion, as described in Section 5.8.1, was investigated. To assess the effect of the angle threshold $\theta_{\text{thresh}}$ on tracking performance, a series of trials were conducted across a range of values from 160° to 30°. Smaller $\theta_{\text{thresh}}$ indicates a more restrictive constraint. The results of this tuning effort are presented in Table 10.

These results demonstrate that progressively lowering the angle threshold leads to a consistent improvement of tracking metrics. Precision, recall, and $J_{\text{map}}$ are all shown to be maximised when the threshold is at its most restrictive. Broadly speaking, this supports the hypothesis that angle-based rejection as post-processing technique improves overall tracking results. However, this improvement comes at a cost. The first row of Table 10 simply lists the ground truth data statistics, where 267,219 true links are present. As $\theta_{\text{thresh}}$ gets smaller, we observe greater fragmentation of tracks and as a result lowers the total number expected pairs and increases the percentage of missed GT pairs. Additionally, more aggressive splitting increases the computational time to of post-processing, as every time a track is split it gets re-evaluated recursively.

To balance performance and practical considerations of computational efficiency, an angle threshold of 90° was selected for the final configuration. This threshold presents a consider-

| Trial | $\theta_{\text{thresh}}$ (deg) | Tracking Metric | | | | |
|---|---|---|---|---|---|---|
| | | Precision (%) | Recall (%) | $J_{\text{map}}$ | Expected Pairs | GT Pairs Missing (%) |
| GT (Raw Data) | - | 100 | 100 | 1 | 267,219 | 0 |
| No Angle Rejection | - | 86.90 | 87.47 | 0.2893 | 114,624 | 57.10 |
| 1 | 160 | 89.44 | 89.94 | 0.3631 | 114,172 | 57.28 |
| 2 | 130 | 89.69 | 90.17 | 0.3761 | 113,755 | 57.43 |
| 3 | 110 | 89.84 | 90.35 | 0.3863 | 113,368 | 57.57 |
| 4 | 90 | 90.00 | 90.54 | 0.3961 | 112,918 | 57.74 |
| 5 | 60 | 90.43 | 91.11 | 0.4270 | 111,863 | 58.14 |
| 6 | 45 | 90.81 | 91.56 | 0.4509 | 110,974 | 58.47 |
| 7 | 30 | 91.54 | 92.32 | 0.4947 | 109,438 | 59.05 |

Table 10: Tracking performance under different angle thresholds, with the highlighted row showing the angle chosen for the proposed method as it enforces a realistic no backwards motion constraint.

able improvement in tracking metrics over the baseline, while avoiding excessive fragmentation associated with smaller thresholds which can significantly increase post-processing time. This value is also conceptually practical, as it effectively enforces a no backwards motion constraint. This will prevent unrealistic trajectories while still allowing for some vessel curvature which one would expect in a vascular structure. Additionally, it is also consistent with the work presented by Tang et al. [44].

The effects of including different post-processing was explored by testing different combinations of rejections. The results, summarised in Table 11, show that applying all three of track post-processing rejections yields the best tracking performance. The final configuration, combining angle, acceleration, and short track-based rejections, achieves the highest precision and $J_{\text{map}}$, and only a marginally lower recall than some of the other trials. This configuration considerably outperforms the baseline, showing a $\sim 0.15$ increase $J_{\text{map}}$. While it results in a slight increase in the percentage of missing GT pairs, this is increase is small and is acceptable given the improvement in tracking quality.

Comparing Trials 2 and 3 supports the value of each criterion. Both angle only and acceleration demonstrate improvement over the baseline, with relative small difference between their respective tracking metrics. When combining the rejections in Trial 4, the effects of the two rejections appear additive rather than overlapping as this was the best overall performance. It is worth noting that in the implementation of Trial 4, the rejection criteria were applied in the following order: Angle, acceleration, and then short track removal. Since each criterion splits tracks at the index at which it violates the threshold adds the segments to the set of all tracks, the ordering of these rejections directly influences the motion history available to subsequent rejections. This is particularly relevant for acceleration-based rejection, as $a_{\text{thresh}}$ is computed adaptively based on the track history. It is likely that if the rejection order were changed, the final tracking results would be different.

| Trial | Track Rejection Criteria | | | Tracking Metric | | | | |
|---|---|---|---|---|---|---|---|---|
| | Angle | Acceleration | Short Track | Precision (%) | Recall (%) | $J_{\text{map}}$ | Expected Pairs | GT Pairs Missing (%) |
| 1 | False | False | True | 87.07 | 87.72 | 0.2893 | 114,624 | 57.10 |
| 2 | True | False | True | 90.00 | 90.54 | 0.3961 | 112,918 | 57.74 |
| 3 | False | True | True | 89.54 | 90.51 | 0.4038 | 112,730 | 57.81 |
| 4 | True | True | True | 90.43 | 90.31 | 0.4380 | 110,331 | 58.45 |

Table 11: Tracking performance under different combinations of track post-processing rejection criteria, showing that applying all the criteria optimises the metrics.

### 6.1.6 Proposed Method Summary

This section summarises the final Kalman filter-based tracking pipeline used in the subsequent model performance evaluation. All design and parameter choices were selected based on the experimental results presented in the previous subsections. These efforts led to a final configuration aimed to maximise tracking performance.

The proposed tracking pipeline consists of:

- A constant-velocity Kalman filter for motion modelling (Section 5.4),
- A linear assignment for detection to track association (Section 5.5),
- Linking rejection using:
  - Velocity-based rejection (Section 5.6.1),
  - Short link rejection (Section 5.6.2),
- Track post-processing using:
  - Angle-based rejection (Section 5.8.1),
  - Acceleration-based rejection (Section 5.8.2),
  - Short track rejection (Section 5.8.3).

The Kalman filter was tuned via grid search (Section 6.1.1), resulting in the following parameter values:

$$\mathbf{P}_{\mathbf{0}}^{+} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; \mathbf{R} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}.$$

## 6.2 Model Performance

This section evaluates the tracking performance of the final proposed method across two scenarios. First, we assess performance on simulated datasets with increasing levels of localisation noise described in Section 4.4. Second, we evaluate on the actual localisation results obtained from the localisation pipeline described in Section 4.1. In both cases, the final proposed model is compared against the baseline Kalman filter implementation without angle and acceleration-based post-processing track rejections to quantify the improvements achieved through tuning and rejection strategies.

### 6.2.1 Simulated Localisation Results

To assess the robustness of the proposed tracking method, we evaluate its performance on the collection of simulated localisation datasets with increasing levels of Gaussian noise, which are summarised in Table 5. These datasets simulate progressively decreasing quality of localisation input to the tracking pipeline, allowing a controlled evaluation of tracking performance under different input conditions. After running the final proposed model on all of the Simulated Localisation (SL) datasets, an analysis of the resulting tracking output was conducted to explore the effect of increasing localisation noise on the resulting MB statistics and track structures, compared against the GT raw data.

These findings are summarised in Table 12. Across all simulated datasets, the proposed method produces a significantly higher number of tracks, which are on average shorter than those present in the raw data. As localisation noise increases, we observe an increasing number of tracks in the output. The mean track length does not vary greatly between the results in the SL datasets. This highlights the difficulty of maintaining longer trajectories under noisy localisation conditions.

Recall that the tracking method includes the short track rejection criterion described in Section 5.8.3. As more fragmented and shorter tracks are formed, this criterion will lead to a higher number of rejections, removing tracks that are deemed too short to be meaningful. Consequently, we observe a decrease in the MB statistics that the total and per frame number of MBs and mean MB density per frame in the reconstructed tracking results compared to the

| | Raw Data | Tracking Results | | | | | |
|---|---|---|---|---|---|---|---|
| **Dataset Info** | | | | | | | |
| Dataset ID | GT | SL-95 | SL-90 | SL-85 | SL-80 | SL-75 | SL-70 |
| **Microbubble Statistics** | | | | | | | |
| Total number of MBs | 275,163 | 232,769 | 231,025 | 230,989 | 231,059 | 230,698 | 230,175 |
| Mean number of MBs per frame | 550.33 | 465.54 | 462.05 | 461.98 | 462.12 | 461.40 | 460.35 |
| Mean MB density (MB/cm$^2$) | 137.58 | 116.38 | 115.51 | 115.49 | 115.53 | 115.35 | 115.09 |
| **Track Statistics** | | | | | | | |
| Number of tracks | 7,802 | 34,255 | 34,483 | 34,656 | 34,783 | 35,009 | 35,448 |
| Mean track length (frames) | 35.27 | 6.80 | 6.70 | 6.67 | 6.64 | 6.59 | 6.49 |
| Minimum track length (frames) | 1 | 3 | 3 | 3 | 3 | 3 | 3 |
| Maximum track length (frames) | 500 | 115 | 97 | 93 | 81 | 94 | 74 |

Table 12: Summary of MB and track statistics for the GT raw data and for the tracking results of the final proposed model applied to synthetic localisation datasets (SL-95 to SL-70).

raw data. Despite this, the MB statistics remain relatively consistent across all the SL datasets, suggesting robustness of the tracking as input quality degrades.

Table 13 compares the tracking metrics obtained by the baseline and the final proposed Kalman filter models across the synthetic datasets. As previously noted, localisation error propagates through to tracking results, so as expected both models demonstrate a degradation of tracking performance as localisation quality decreases from SL-95 to SL-70. This is reflected through the decrease of each precision, recall, and $J_{\text{map}}$. Localisation noise introduces additional complexity to the tracking problem, making it more difficult to reliably link MBs between frames and leading to more incorrect pairings.

It is shown the proposed model consistently outperforms the baseline model, with particularly strong improvements in the $J_{\text{map}}$ scores. The relative improvement of this metric in Table 13 becomes less significant with increased localisation noise. Overall, this highlights how the additional post-processing rejection criteria are important to remove implausible trajectories that would otherwise be hindering the tracking performance.

The recall metric across all datasets is shown to be slightly higher for the baseline model. This indicates that the method links more MB detections overall. However, increased recall may also reflect a higher amount of FPs, which reduces precision and the overall quality of the reconstructed tracks. This interpretation aligns with the baseline method's lower $J_{\text{map}}$ scores, as a higher proportion of incorrect or missed links results in a greater error distance. The proposed model could be considered more selective, where at the cost of a slightly lower recall, it rejects low quality links resulting in increased precision and $J_{\text{map}}$.

| | Tracking Metrics - Baseline Model | | | Tracking Metrics - Proposed Model | | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | **Precision (%)** | **Recall (%)** | $J_{\text{map}}$ | **Precision (%)** | **Recall (%)** | $J_{\text{map}}$ | $\Delta J_{\text{map}}$ **(%)** |
| SL-95 | 81.07 | 82.40 | 0.0952 | 82.66 | 81.15 | 0.1765 | +7.42 |
| SL-90 | 80.78 | 82.26 | 0.0906 | 81.97 | 80.52 | 0.1651 | +6.83 |
| SL-85 | 80.20 | 81.74 | 0.0681 | 81.42 | 79.94 | 0.1386 | +6.60 |
| SL-80 | 79.84 | 81.42 | 0.0561 | 81.11 | 79.54 | 0.1258 | +6.60 |
| SL-75 | 79.40 | 81.22 | 0.0487 | 80.35 | 78.87 | 0.1086 | +5.71 |
| SL-70 | 78.45 | 80.42 | 0.0135 | 79.49 | 77.98 | 0.0727 | +5.84 |

Table 13: Tracking performance comparison between the baseline Kalman filter model (no angle or acceleration-based post-processing rejections) and the final proposed model across synthetic localisation datasets. The last column shows the relative improvement in $J_{\text{map}}$.

For visualisation of the MB trajectory reconstructions, two maps are presented: A track density map, which counts how many tracks pass through each pixel, and a mean MB speed map, which reflects the average velocity of MBs observed in each pixel. To generate these maps the results across all frames in the dataset are compiled, meaning they represent all of the trajectory information rather than a single frame. These maps provide insight into the underlying vascular structure and blood flow dynamics captured by the tracking algorithm.

The maps generated from the GT raw data are presented in Figure 14, where the true nature of the synthetic dataset is displayed. The track density map highlights regions where tracks are highly concentrated, which indicates the presence of larger or more frequently used vessels. The mean speed map shows the wide variety of MB speed in the synthetic dataset. Tracks with high, moderate, and slow speeds are overlapping and travel in various directions. The variability displayed highlights the complexity of reconstructing accurate tracks in such environments, as the algorithm must distinguish between many crossing paths and varying flow speeds. These maps provide a baseline for which the tracking results of proposed model can be compared.

For brevity, the maps for the tracking results of the proposed model using only the SL-95 dataset as input are presented in Figure 15. This figure illustrates the tracking results of the model when applied to a dataset with high quality localisations. As previously discussed and summarised in Table 12, tracking results were observed to output a lower MB density than in the GT data. This is reflected in the track density map, where the maximum value of tracks per pixel is notably reduced compared to the GT. Additionally, tracks are observed in regions where none exist in the GT, likely due to the Kalman filter updates adjusting the positions of detections and the presence of incorrect pairings. These inaccuracies contribute to the increased visual noise in all maps produced by the tracking results. Despite this, the underlying vascular structure is visually recognisable, as some of the dominant vessels exhibit comparatively higher track densities than their surrounding regions.

The speed map for the SL-95 results appear substantially noisier than the corresponding map for the GT data. The highest magnitude speed appear to be mostly derived from the false or incorrect track pairings rather than true MB motion. This suggests that even with relatively accurate localisations as input, tracking errors can highly impact the resultant velocity profile. This aligns with the $J_{\mathrm{map}}$ for SL-95 in Table 12. With $J_{\mathrm{map}} = 0.1765$, this indicates there is only slightly more correctly paired distance over erroneous distance. There are however some consistencies with the GT speed map, particularly the slower moving MBs that branch from the top right of the maps.

For completeness, the SL-95 dataset was also evaluated using the NN and Hungarian algorithm tracking methods, with results provided in Appendix B.

### 6.2.2 Actual Localisation Results

The final dataset examined is the LOC-OPT dataset, which contains localisations results generated from the pre-processing and detection pipeline described in Section 4. This dataset was selected as it represents a realistic set of localisations obtained through the proposed localisation pipeline. As summarised in Table 4, these localisations have what can be considered an acceptable level of precision but a reduced recall. Evaluating tracking results on this dataset provides insight on how the proposed method performs when applied to practical yet imperfect localisations.

A summary of the MB and track statistics of the tracking output for the LOC-OPT dataset is provided in Table 14. Compared to the GT raw data, the tracking results show a substantial reduction in the total number of MBs and MB density. The number of tracks is also considerably higher, with the average track being much shorter compared to the GT. These trends are consistent with what is observed for the simulated localisation results in Table 12, where degrading input quality produces more short tracks. Compared to the SL results however, the LOC-OPT results retains the least amount of MB detection information and produces the shortest tracks. This is likely reflective of the LOC-OPT have the lowest localisation recall metric out of any of the evaluated datasets.

The tracking performance on the LOC-OPT dataset is summarised in Table 15. As with the trials on the simulated localisation datasets, the proposed model incorporating post-processing outperforms the baseline implementation overall. Notable improvements are observed in tracking precision and $J_{\mathrm{map}}$, with recall remaining consistent. This further supports that the post-processing choices are crucial in improving tracking performance across all levels of localisation quality.

| | Raw Data | Tracking Results |
|---|---|---|
| **Dataset Info** | | |
| Dataset ID | GT | LOC-OPT |
| **Microbubble Statistics** | | |
| Total number of MBs | 275,163 | 160,763 |
| Mean number of MBs per frame | 550.33 | 321.53 |
| Mean MB density (MB/cm$^2$) | 137.58 | 80.38 |
| **Track Statistics** | | |
| Number of tracks | 7,802 | 28,053 |
| Mean track length (frames) | 35.27 | 5.73 |
| Minimum track length (frames) | 1 | 3 |
| Maximum track length (frames) | 500 | 67 |

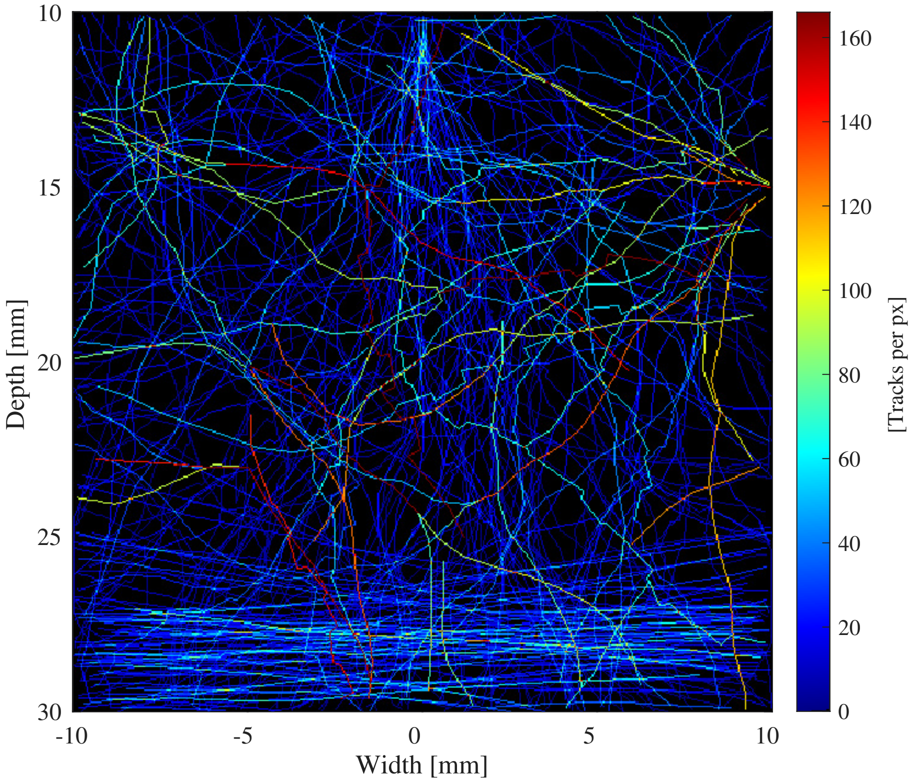
Table 14: Summary of MB and track statistics for the GT raw data and for the tracking results of the final proposed model applied to an actual set of localisations (LOC-OPT).

| | Tracking Metrics - Baseline Model | | | Tracking Metrics - Proposed Model | | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | **Precision (%)** | **Recall (%)** | $J_{\mathbf{map}}$ | **Precision (%)** | **Recall (%)** | $J_{\mathbf{map}}$ | $\Delta J_{\mathbf{map}}$ **(%)** |
| LOC-OPT | 78.98 | 88.61 | 0.0848 | 81.04 | 86.88 | 0.1779 | +8.58 |

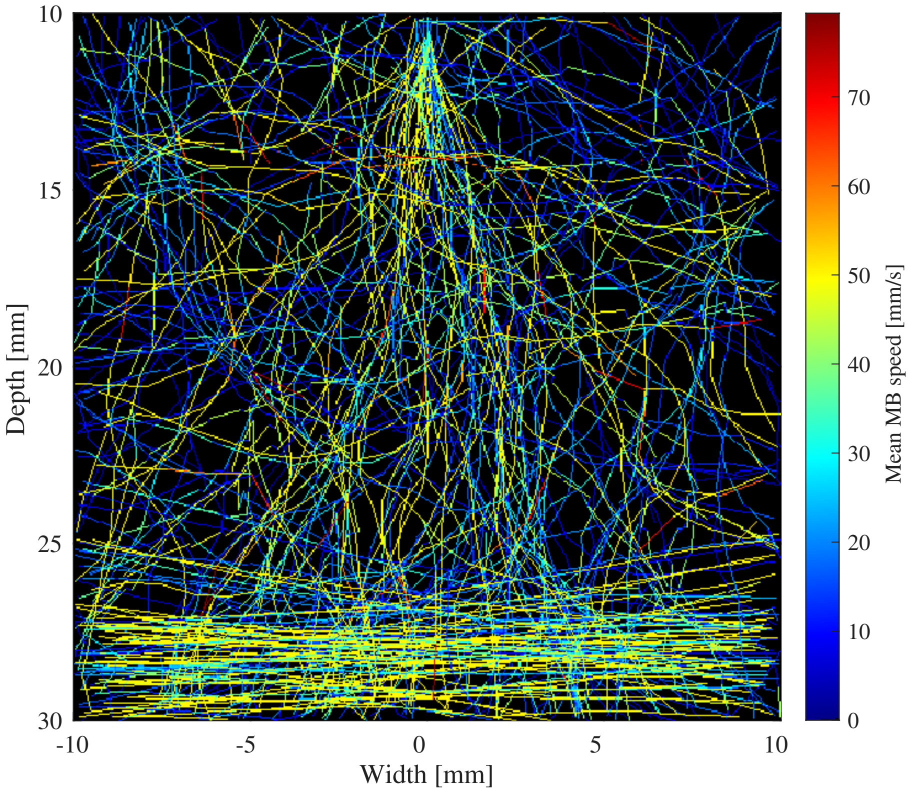
Table 15: Tracking performance comparison between the baseline Kalman filter model (no angle or acceleration-based post-processing rejections) and the final proposed model on the LOC-OPT dataset. The last column shows the relative improvement in $J_{\text{map}}$.

The track density and mean MB speed maps from the LOC-OPT dataset using the proposed tracking model are presented in Figure 16. Similar to the SL-95 results, the maximum pixel count in the track density map is significantly lower than that in the GT, reflecting the reduced number of detections in the tracking results. The maps again display a high level of noise, or links which do not appear in the GT maps. The level of noise seems to be increased from the SL-95 map, particularly in the top left section where there are dense overlapping tracks. This aligns with the reduction in track metrics we observe when evaluating the two datasets, and supports the conclusion that lower quality localisations degrade the tracking performance.

The associated speed map continues the trends observed in the SL-95 results. While the overall magnitude of mean MB speeds are relatively consistent with the GT map, the highest speeds in the LOC-OPT results are achieved in the noisy and erroneous links. In contrast, tracks that visually resemble the GT in shape and path typically correspond to slower speeds. Although the broader vascular structure remains visually apparent in these maps, the noise displays a limitation in the current tracking method. It suggests that further refinements throughout the tracking pipeline may be needed to improve the overall tracking results across the varying quality localisation inputs.

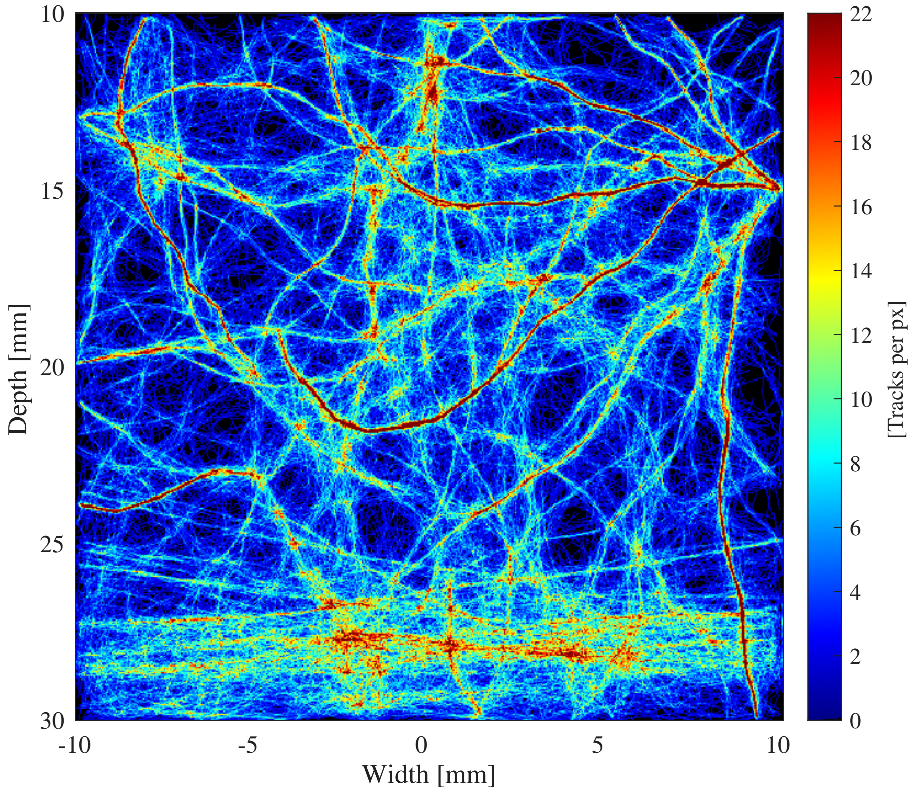



(a) Track density map.

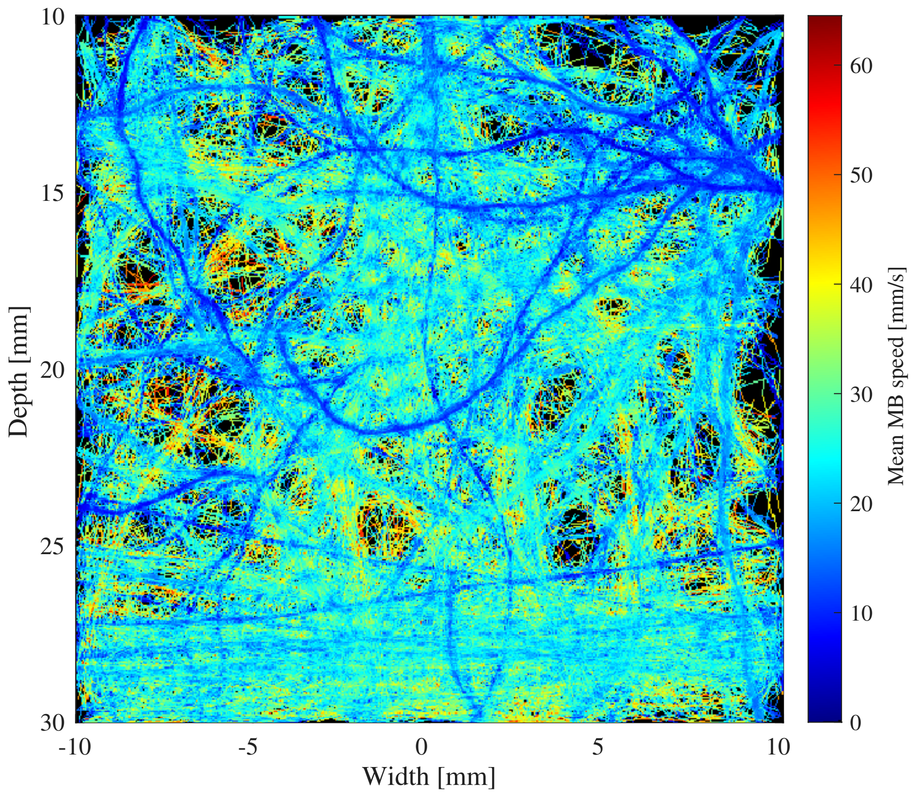



(b) Mean MB speed map.

Figure 14: Track density and velocity maps generated from GT raw data. (a) Number of tracks per pixel. (b) Mean MB speed per pixel.

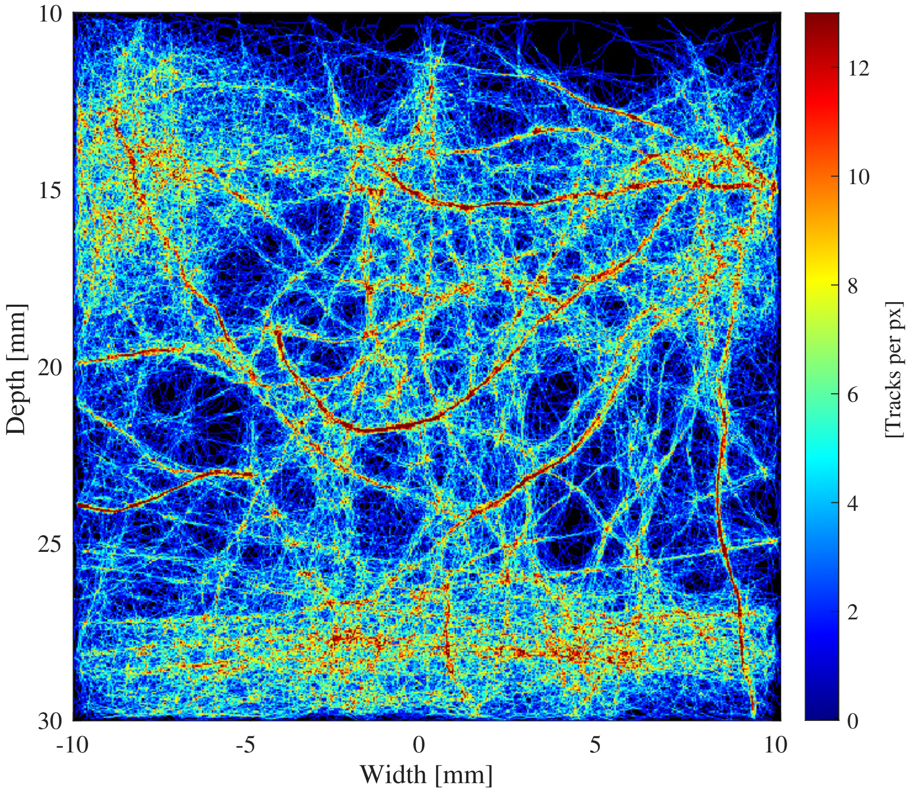



(a) Track density map.

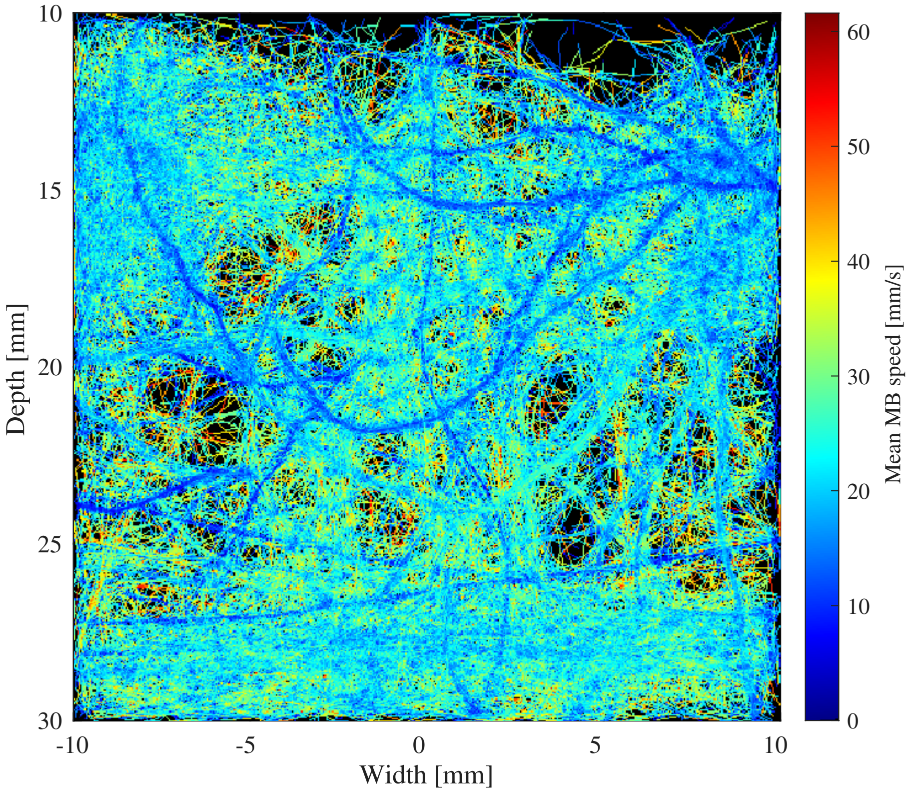



(b) Mean MB speed map.

Figure 15: Track density and velocity maps generated from tracking results of the Kalman filter method on the SL-95 dataset. (a) Number of tracks per pixel. (b) Mean MB speed per pixel.




(a) Track density map.




(b) Mean MB speed map.

Figure 16: Track density and velocity maps generated from tracking results of the Kalman filter method on the LOC-OPT dataset. (a) Number of tracks per pixel. (b) Mean MB speed per pixel.

# 7 Conclusions

This dissertation aimed to develop and evaluate a Kalman filter-based MB tracking algorithm for SRUI. The goal was to assess the effects of motion modelling, linking strategies, and track post-processing in the reconstruction of MB trajectories using synthetic data.

In the development phase of the tracking implementation, several key findings were observed. First, the use of a Kalman motion model significantly outperformed simpler baseline approaches such as NN and the Hungarian algorithm in terms of tracking performance. Second, applying additional angle and acceleration restrictions during the frame-to-frame linking process was found to not improve tracking results. It is assumed that these rejections at this stage are overly restrictive and prematurely rejecting valid continuations of tracks. Lastly, track post-processing rejections based on motion continuity (angle and acceleration) showed improvement in tracking. These methods resulted in improved tracking precision and distance based metrics by splitting implausible trajectories into separate tracks after their initial formation.

Localisations obtained from the localisation pipeline (Section 4.1) in addition to a series of simulated localisation datasets of varying quality were assessed. As localisation quality decreased, tracking performance gradually degraded, but the inclusion of track post-processing was shown to outperform a baseline implementation in which they were not included. The model produced a larger number of short tracks than what was present in the raw data. With the rejection of very small tracks, this resulted in a model which prioritised making high quality links at the expense of fewer GT pairs retained. The results showed that the model had difficulty reconstructing long trajectories and instead resulted in smaller fragments of said tracks. Visualisations of the tracking results showed that while the vascular structure was preserved in the broader sense, high speed erroneous links were common under noisy localisation input. It suggests that the proposed model may be better at capturing slower moving MB trajectories.

Overall, this dissertation contributes to ongoing efforts to achieve more reliable reconstruction of vascular flow dynamics, therefore advancing the broader goals of SRUI.

## 7.1 Limitations and Future Work

This work has a number of limitations. The evaluation was conducted on a single synthetic dataset, which limits the generalisability of the findings. Applying the same model with the tuned parameters to datasets with different MB densities, noise levels, and imaging characteristics would provide stronger evidence for robustness. It is likely that parameter retuning would be required for satisfactory results on other datasets. The tuning efforts presented were conducted on the set of perfect localisations which is not a realistic scenario for a full SRUI pipeline. Better tracking performance in the noisy localisation input trials may have been observed if the Kalman filter had been tuned with noisy data. In addition, the current implementation assumes continuous detections and does not permit temporal gaps within tracks. In practice, imperfect localisations are expected and means that MBs belonging to a track may be missing in one or more frames. The Kalman filter could allow tracks without a detection to remain active for a set number of frames by propagating predictions without updates. Further, the implemented track rejection strategy simply removes tracks below a length threshold. While this improved tracking precision, it resulted in removing a large number of short tracks and consequently a non-negligible loss of information.

Several algorithmic assumptions further constrain the approach. The Kalman filter is based on a linear motion model, which may not reflect true MB dynamics as they travel throughout bending and branching vascular structure. The Kalman filter implementation applies a fixed amount of Gaussian noise to the prediction and measurement processes which may not be an accurate reflection of the true error.

A number of future research directions could be explored. Additional rejection criteria, such as MB density or size continuity constraints, may further increase the accuracy of the tracks which are reconstructed. Further post-processing steps such as re-tracking or linking tracks together could also better recreate the true vascular structure from shorter segments. In addition, alternative evaluation metrics could be considered where the accuracy of full tracks are assessed rather than just frame-to-frame pairings. Instead of implementing a grid search approach for tuning the Kalman filter parameters, an adaptive or learning-based method where an algorithm determines the optimal parameters from the data could be explored. This could make the process more applicable to other datasets. The robustness of the method could be tested by running the dataset in reverse order (i.e. from the last frame to first). In theory the tracking results obtained should be the same, but it would be interesting to compare how the results differ. Finally, future work should extend beyond synthetic data to *in vivo* datasets. Three-dimensional tracking and incorporating estimations of uncertainty could make the method more suitable for clinical applications. Exploring these avenues would help the robustness, generalisability, and applicability of Kalman filter-based tracking for SRUI.

# References


[1] D. Ackermann and G. Schmitz. Detection and tracking of multiple microbubbles in ultrasound B-mode images. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 63(1):72–82, Jan. 2016.

[2] Y. Bar-Shalom and X.-R. Li. *Multitarget-multisensor tracking: principles and techniques*, volume 19. YBS publishing Storrs, CT, 1995.

[3] M. J. K. Blomley, J. C. Cooke, E. C. Unger, M. J. Monaghan, and D. O. Cosgrove. Microbubble contrast agents: A new era in ultrasound. *BMJ : British Medical Journal*, 322(7296):1222–1225, May 2001.

[4] F. Bourgeois and J.-C. Lassalle. An extension of the Munkres algorithm for the assignment problem to rectangular matrices. *Commun. ACM*, 14(12):802–804, Dec. 1971.

[5] M. B. Butler, G. Papageorgiou, E. D. Kanoulas, V. Voulgaridou, H. Wijkstra, M. Mischi, C. K. Mannaerts, S. McDougall, W. C. Duncan, W. Lu, and V. Sboros. Mapping of prostate cancer microvascular patterns using super-resolution ultrasound imaging. *European Radiology Experimental*, 9(1):25, Feb. 2025.

[6] Y. Cao. Munkres assignment algorithm. Available at `https://www.mathworks.com/matlabcentral/fileexchange/20328-munkres-assignment-algorithm`, June 2008.

[7] P. Carmeliet and R. K. Jain. Angiogenesis in cancer and other diseases. *Nature*, 407(6801):249–257, Sept. 2000.

[8] K. Christensen-Jeffries, R. J. Browning, M.-X. Tang, C. Dunsby, and R. J. Eckersley. In vivo acoustic super-resolution and super-resolved velocity mapping using microbubbles. *IEEE transactions on medical imaging*, 34(2):433–440, Feb. 2015.

[9] K. Christensen-Jeffries, O. Couture, P. A. Dayton, Y. C. Eldar, K. Hynynen, F. Kiessling, M. O'Reilly, G. F. Pinton, G. Schmitz, M.-X. Tang, M. Tanter, and R. J. G. van Sloun. Super-resolution ultrasound imaging. *Ultrasound in Medicine & Biology*, 46(4):865–891, Apr. 2020.

[10] D. Cosgrove and N. Lassau. Imaging of perfusion using ultrasound. *European Journal of Nuclear Medicine and Molecular Imaging*, 37(1):65–85, Aug. 2010.

[11] C. Demené, J. Robin, A. Dizeux, B. Heiles, M. Pernot, M. Tanter, and F. Perren. Transcranial ultrafast ultrasound localization microscopy of brain vasculature in patients. *Nature Biomedical Engineering*, 5(3):219–228, Mar. 2021. Publisher: Nature Publishing Group.

[12] S. Dencks, M. Piepenbrock, T. Opacic, B. Krauspe, E. Stickeler, F. Kiessling, and G. Schmitz. Clinical pilot application of super-resolution US imaging in breast cancer. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 66(3):517–526, Mar. 2019.

[13] S. Dencks and G. Schmitz. Ultrasound localization microscopy. *Zeitschrift für Medizinische Physik*, 33(3):292–308, Aug. 2023.

[14] M. Dewey, M. Siebes, M. Kachelrieß, K. F. Kofoed, P. Maurovich-Horvat, K. Nikolaou, W. Bai, A. Kofler, R. Manka, S. Kozerke, A. Chiribiri, T. Schaeffter, F. Michallek, F. Bengel, S. Nekolla, P. Knaapen, M. Lubberink, R. Senior, M.-X. Tang, J. J. Piek, T. van de Hoef, J. Martens, and L. Schreiber. Clinical quantitative cardiac imaging for the assessment of myocardial ischaemia. *Nature Reviews Cardiology*, 17(7):427–450, July 2020. Publisher: Nature Publishing Group.

[15] I. S. Duff and J. Koster. On algorithms for permuting large entries to the diagonal of a sparse matrix. *SIAM Journal on Matrix Analysis and Applications*, 22(4):973–996, Jan. 2001. Publisher: Society for Industrial and Applied Mathematics.

[16] 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*, 527(7579):499–502, Nov. 2015. Publisher: Nature Publishing Group.

[17] A. Genovesio, T. Liedl, V. Emiliani, W. J. Parak, M. Coppey-Moisan, and J.-C. Olivo-Marin. Multiple particle tracking in 3-D+t microscopy: Method and application to the tracking of endocytosed quantum dots. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 15(5):1062–1070, May 2006.

[18] W. J. Godinez, M. Lampe, S. Wörz, B. Müller, R. Eils, and K. Rohr. Deterministic and probabilistic approaches for tracking virus particles in time-lapse fluorescence microscopy image sequences. *Medical Image Analysis*, 13(2):325–342, Apr. 2009.

[19] J. Gomez-Gil, R. Ruiz-Gonzalez, S. Alonso-Garcia, and F. J. Gomez-Gil. A Kalman filter implementation for precision improvement in low-cost GPS positioning of tractors. *Sensors*, 13(11):15307–15323, Nov. 2013. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[20] M. S. Grewal and A. P. Andrews. Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Systems Magazine*, 30(3):69–78, June 2010.

[21] 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*, 6(5):605–616, May 2022. Publisher: Nature Publishing Group.

[22] V. Hingot, C. Errico, B. Heiles, L. Rahal, M. Tanter, and O. Couture. Microvascular flow dictates the compromise between spatial resolution and acquisition time in ultrasound localization microscopy. *Scientific Reports*, 9(1):2456, Feb. 2019. Publisher: Nature Publishing Group.

[23] C. Huang, W. Zhang, P. Gong, U.-W. Lok, S. Tang, T. Yin, X. Zhang, L. Zhu, M. Sang, P. Song, R. Zheng, and S. Chen. Super-resolution ultrasound localization microscopy based on a high frame-rate clinical ultrasound scanner: an in-human feasibility study. *Physics in Medicine & Biology*, 66(8):08NT01, Apr. 2021. Publisher: IOP Publishing.

[24] P. J. Huber. Robust statistics: A review. *Ann. Math. Statist*, 43:1041–1067, 1972.

[25] S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. page 182, Orlando, FL, USA, July 1997.

[26] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, Mar. 1960.

[27] E. Kanoulas, M. Butler, C. Rowley, V. Voulgaridou, K. Diamantis, W. C. Duncan, A. McNeilly, M. Averkiou, H. Wijkstra, M. Mischi, R. S. Wilson, W. Lu, and V. Sboros. Super-resolution contrast-enhanced ultrasound methodology for the identification of in vivo vascular dynamics in 2D. *Investigative Radiology*, 54(8):500, Aug. 2019.

[28] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1):83–97, 1955. e-print: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.20053`.

[29] M. Lerendegui, K. Riemer, G. Papageorgiou, B. Wang, L. Arthur, A. Chavignon, T. Zhang, O. Couture, P. Huang, M. Ashikuzzaman, S. Dencks, C. Dunsby, B. Helfield, J. A. Jensen, T. Lisson, M. R. Lowerison, H. Rivaz, A. E. Samir, G. Schmitz, S. Schoen, R. van Sloun, P. Song, T. Stevens, J. Yan, V. Sboros, and M.-X. Tang. ULTRA-SR challenge: Assessment of ultrasound localization and tracking algorithms for super-resolution imaging. *IEEE Transactions on Medical Imaging*, 43(8):2970–2987, Aug. 2024.

[30] M. Lerendegui, K. Riemer, B. Wang, C. Dunsby, and M.-X. Tang. BUbble Flow Field: a simulation framework for evaluating ultrasound localization microscopy algorithms, Nov. 2022. arXiv:2211.00754 [eess].

[31] F. Lin, S. E. Shelton, D. Espíndola, J. D. Rojas, G. Pinton, and P. A. Dayton. 3-D ultrasound localization microscopy for identifying microvascular morphology features of tumor angiogenesis at a resolution beyond the diffraction limit of conventional ultrasound. *Theranostics*, 7(1):196–204, Jan. 2017. Publisher: Ivyspring International Publisher.

[32] G. D. Ludwig. The velocity of sound through tissues and the acoustic impedance of tissues. *The Journal of the Acoustical Society of America*, 22(6):862–866, Nov. 1950.

[33] J. Mazzaferri, J. Roy, S. Lefrancois, and S. Costantino. Adaptive settings for the nearest-neighbor particle tracking algorithm. *Bioinformatics (Oxford, England)*, 31(8):1279–1285, Apr. 2015.

[34] A. Mobberley, G. Papageorgiou, M. Butler, E. Kanoulas, J. Keanie, D. Good, K. Gallagher, A. McNeil, V. Sboros, and W. Lu. Particle tracking with neighbourhood similarities: A new method for super resolution ultrasound imaging. pages 29–40, 2023.

[35] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, Mar. 1957. Publisher: Society for Industrial and Applied Mathematics.

[36] S. Oh, S. Russell, and S. Sastry. Markov Chain Monte Carlo data association for general multiple-target tracking problems. In *2004 43rd IEEE Conference on Decision and Control (CDC)*, volume 1, pages 735–742 Vol.1, Dec. 2004. ISSN: 0191-2216.

[37] T. Opacic, S. Dencks, B. Theek, M. Piepenbrock, D. Ackermann, A. Rix, T. Lammers, E. Stickeler, S. Delorme, G. Schmitz, and F. Kiessling. Motion model ultrasound localization microscopy for preclinical and clinical multiparametric tumor characterization. *Nature Communications*, 9(1):1527, Apr. 2018. Publisher: Nature Publishing Group.

[38] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, Dec. 1979.

[39] I. Smal, K. Draegestein, N. Galjart, W. Niessen, and E. Meijering. Particle filtering for multiple object tracking in dynamic fluorescence microscopy images: application to microtubule growth analysis. *IEEE transactions on medical imaging*, 27(6):789–804, June 2008.

[40] O. Solomon, R. J. G. van Sloun, H. Wijkstra, M. Mischi, and Y. C. Eldar. Exploiting flow dynamics for superresolution in contrast-enhanced ultrasound. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 66(10):1573–1586, Oct. 2019.

[41] 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, and frequency control*, 65(2):149–167, Feb. 2018.

[42] T. Szabo. Diagnostic ultrasound imaging—inside out. Sept. 2004.

[43] I. Taghavi, S. B. Andersen, C. A. V. Hoyos, M. Schou, F. Gran, K. L. Hansen, M. B. Nielsen, C. M. Sørensen, M. B. Stuart, and J. A. Jensen. Ultrasound super-resolution imaging with a hierarchical Kalman tracker. *Ultrasonics*, 122:106695, May 2022.

[44] S. Tang, P. Song, J. D. Trzasko, M. Lowerison, C. Huang, P. Gong, U.-W. Lok, A. Manduca, and S. Chen. Kalman filter-based microbubble tracking for robust super-resolution ultrasound microvessel imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 67(9):1738–1751, Sept. 2020.

[45] R. J. G. van Sloun, O. Solomon, M. Bruce, Z. Z. Khaing, H. Wijkstra, Y. C. Eldar, and M. Mischi. Super-resolution ultrasound localization microscopy through deep learning. *IEEE transactions on medical imaging*, 40(3):829–839, Mar. 2021.

[46] K. Wei. Contrast echocardiography: Applications and limitations. *Cardiology in Review*, 20(1):025, Feb. 2012.

[47] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report, University of North Carolina at Chapel Hill, USA, Oct. 1995.

[48] J. Yan, T. Zhang, J. Broughton-Venner, P. Huang, and M.-X. Tang. Super-resolution ultrasound through sparsity-based deconvolution and multi-feature tracking. *IEEE transactions on medical imaging*, 41(8):1938–1947, Aug. 2022.

[49] J. Zhu, E. M. Rowland, S. Harput, K. Riemer, C. H. Leow, B. Clark, K. Cox, A. Lim, K. Christensen-Jeffries, G. Zhang, J. Brown, C. Dunsby, R. J. Eckersley, P. D. Weinberg, and M.-X. Tang. 3D super-resolution US imaging of rabbit lymph node vasculature in vivo by using microbubbles. *Radiology*, 291(3):642–650, June 2019. Publisher: Radiological Society of North America.

# Appendices

# A Localisation Pipeline

This appendix aims to provide greater detail to the localisation pipeline as was described in Section 4 and visualised in Figure 5. The localisation workflow follows the methods outlined by Kanoulas et al. [27]. The MATLAB implementation of the code was provided by Dr George Papageorgiou and team. The parameter tuning was performed by the author as part of the localisation pipeline optimisation during this dissertation, as discussed in Section 4.3.

The pre-processing steps are applied to each frame independently. For each frame we start with the raw image, a $512 \times 512$ sized grayscale image matrix of intensity values within the range $[0, 255]$ per pixel. The goal of the pre-processing steps is to isolate the MB signal from the background and noise. At a high level, the raw image is transformed through brightness adjustment, Haar-like feature filtering, uniform smoothing, after which candidate MB regions are mapped and refined using watershed segmentation. The final output is a map for each frame which are the same size of the original images and contain labelled regions of probable MB locations. The following is a simplified example of such a map:

$$\text{map} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 3 & 3 & 3 \\ 2 & 2 & 0 & 0 & 4 & 4 \\ 2 & 2 & 2 & 0 & 4 & 4 \\ 2 & 2 & 2 & 4 & 4 & 4 \end{bmatrix}.$$

In this example, there are four detected MBs labelled within a $6 \times 6$ sized image, with zeroes representing the regions the pre-processing has determined to be background or noise. The MB segments do not need to be regularly shaped.

Next is the localisation stage, where we apply a centroid-based method [8]. The map output of the pre-processing stage is applied as a mask to the original raw image to extract intensity values of each detected MB. The centre of mass (COM) position of each segment is computed using the intensity-weighted coordinates of the pixels:

$$x_{\text{COM}} = \frac{\sum_{i=1}^{N} I_i \cdot x_i}{\sum_{i=1}^{N} I_i}, \qquad z_{\text{COM}} = \frac{\sum_{i=1}^{N} I_i \cdot z_i}{\sum_{i=1}^{N} I_i},$$

where $N$ is the number of pixels in the segmented region, $I_i$ and $(x_i, z_i)$ represent the intensity and coordinates of the $i$-th pixel in the region, respectively. The resulting coordinates, $(x_{\text{COM}}, z_{\text{COM}})$, are used as the final localisation of the MB. This computation is repeated for each labelled region in the map for every frame. In addition to the position, the mean intensity and area (in pixels) of each segmented regions in every frame is recorded.

To summarise, we have for each frame a list of detected MBs, including their position, intensity (normalised from 0-1), and size (in terms of area). To finalise the localisations, we apply simple size and intensity thresholding criteria, with tuned parameters defined in Table 3. These parameters were tuned iteratively and their performance was evaluated using the localisation metrics (Section 4.2). Any detections outside the thresholds are discarded. The remaining detections form the final set of localisations across all frames.

# B Baseline Tracking Methods

This section describes the baseline tracking method implementations in greater detail, as mentioned in Section 5.2. These methods serve as the comparison to our proposed Kalman filter-based implementation.

For both algorithms, CostMatrix is a pairwise Euclidean distance matrix between localisations in frames $k-1$ and $k$ with size nTracks $\times$ nDetections, where nTracks is the number of active tracks in frame $k-1$ and nDetections is the number of detections in frame $k$. The outputs, $\mathbf{M}$, $u_R$, and $u_C$ follow the nomenclature presented in the documentation of the MATLAB function matchpairs.m. $\mathbf{M}$ is a matrix where the $i^{\text{th}}$ row contains a pair of indices representing an assignment from a track in frame $k-1$ to a detection in frame $k$. $u_R$ is a list of unassigned row indices, corresponding to tracks in frame $k-1$. $u_C$ is a list of unassigned column indices, corresponding to detections in frame $k$. These algorithms are applied across all frames and use the $\mathbf{M}$, $u_R$, and $u_C$ to either append links to the respective tracks, end tracks, or start new tracks. The velocity rejection (Section 5.6.1), short link rejection (Section 5.6.2), and short track track rejection (Section 5.8.3) are applied in both methods.

**Greedy Nearest Neighbour**

Algorithm 1 presents the greedy nearest neighbour assignment strategy. A simple distance threshold, as defined in Section 5.6.1, is first applied to invalidate implausible links. Each track is sequentially linked to the closest available detection in the next frame. A one-to-one constraint is enforced by setting the corresponding column to infinity in CostMatrix, preventing it from being considered for subsequent tracks. This greedy implementation is based on local proximity and not on global optimal assignment. This implementation is similar to that of [18, 33].

**Algorithm 1** Greedy Nearest Neighbour Assignment

**Input:** CostMatrix, $L_{\max}$
**Output:** $\mathbf{M}$, $u_R$, $u_C$
Set entries in CostMatrix greater than $L_{\max}$ to $\infty$
$(\text{nTracks}, \text{nDetections}) \leftarrow$ size of CostMatrix
$\mathbf{M} \leftarrow [\,]$ ▷ Initialise
assignedTracks $\leftarrow$ false(nTracks) ▷ Initialise
assignedDetections $\leftarrow$ false(nDetections) ▷ Initialise
**for** $i = 1$ to nTracks **do**
    $(\text{minDist},\ j) \leftarrow \min\big(\text{CostMatrix}(i, :)\big)$
    $\mathbf{M} \leftarrow [\mathbf{M}; (i,\ j)]$
    assignedTracks$(i) \leftarrow$ true
    assignedDetections$(j) \leftarrow$ true
    CostMatrix$(:,\ j) \leftarrow \infty$ ▷ One-to-one constraint
**end for**
$u_R \leftarrow [\, i \mid i \notin \text{assignedTracks}\,]$
$u_C \leftarrow [\, j \mid j \notin \text{assignedDetections}\,]$

**Hungarian/Munkres Algorithm**

Algorithm 2 presents the Hungarian/Munkres Algorithm assignment strategy. Again, the distance threshold as defined in Section 5.6.1 is applied to the cost matrix. The implementation uses Yi Cao's MATLAB function munkres.m [6], which applies the Munkres variation of the original Hungarian method [35], with the additional ability to allow for partial assignments. Cao's function, which is available on the MATLAB file exchange, was found to be more computationally efficient than MATLAB's built-in function assignmunkres.m. The output of munkres.m is a vector $a$ of length nTracks, where $a(i)$ is the index of the detection assigned to track $i$. If $a(i) = 0$, track $i$ is unassigned. From $a$, we construct the assignment matrix $\mathbf{M}$ and unassigned rows and column lists. This method guarantees global optimal partial assignment. This implementation is similar to the work of the Harvard Medical School Massachusetts General Hospital, whose work is described in the ULTRA-SR challenge paper [29] and also utilises a wrapper of Cao's function.

**Algorithm 2** Hungarian/Munkres Algorithm Assignment

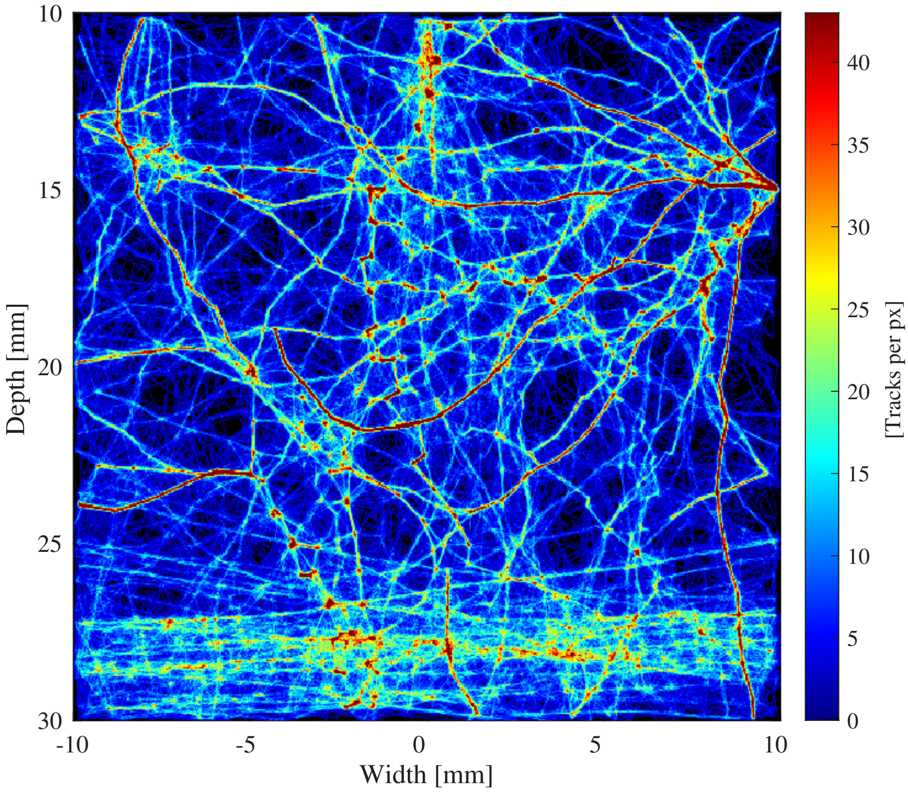
**Input:** CostMatrix, $L_{\max}$
**Output:** $\mathbf{M}$, $u_R$, $u_C$
Set entries in CostMatrix greater than $L_{\max}$ to $\infty$
a $\leftarrow$ munkres(CostMatrix) ▷ Apply Munkres assignment
assignedTracks $\leftarrow [\, i \mid a(i) > 0 \,]$
assignedDetections $\leftarrow a$(assignedTracks)
$\mathbf{M} \leftarrow [\,$assignedTracks(:), assignedDetections(:)$\,]$
$u_R \leftarrow [\, i \mid i \notin$ assignedTracks$\,]$
$u_C \leftarrow [\, j \mid j \notin$ assignedDetections$\,]$

**Baseline Method Tracking Performance on SL-95**

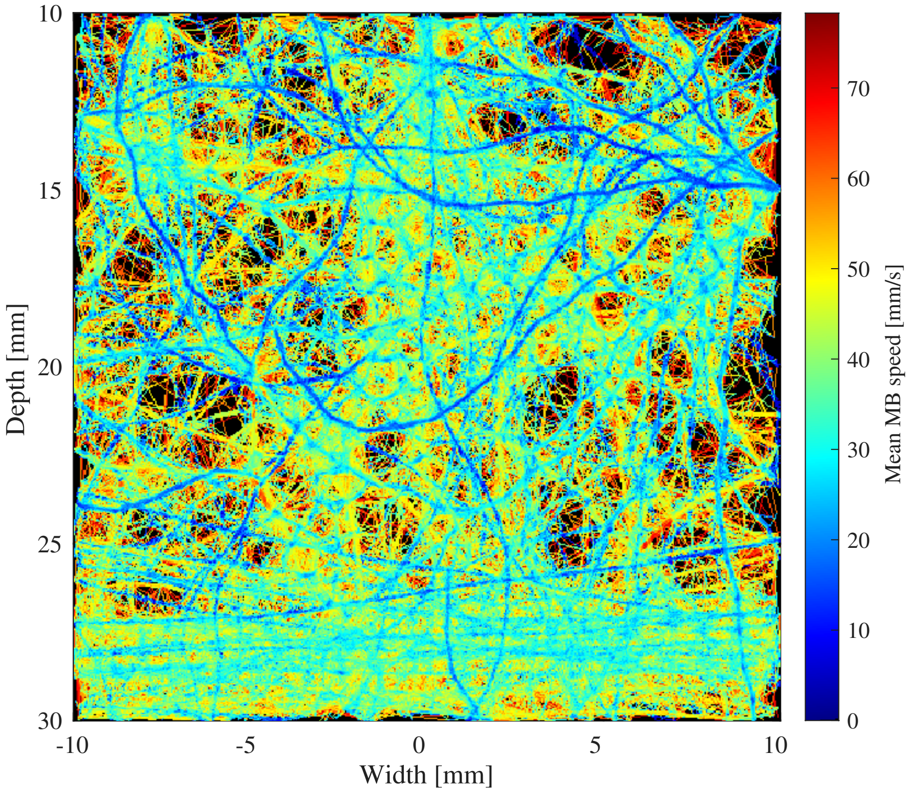
For comparison purposes, Table 16 summarises the performance of the final proposed Kalman filter-based method (Section 6.1.6) and the two baseline methods when applied on the SL-95 dataset. This is distinct from the comparison shown in Table 8, where the methods were applied on the ground truth localisations. The baseline methods do not perform strongly when noise is introduced in the localisations. Figures 17 and 18 display the maps generated from the tracking results of the baseline methods. They both show a higher track density than what is observed for the Kalman implementation (Figure 15), but with increased noise from erroneous links in the mean MB speed maps.

| | Tracking Metric | | | Effective Localisation Metric | |
|---|---|---|---|---|---|
| **Method** | **Precision (%)** | **Recall (%)** | $J_{\mathbf{map}}$ | **Precision (%)** | **Recall (%)** |
| Kalman Filter | 82.66 | 81.15 | 0.1765 | 45.82 | 38.76 |
| Nearest Neighbour | 54.96 | 56.51 | -0.5792 | 95.51 | 92.44 |
| Hungarian Algorithm | 46.46 | 47.64 | -0.6721 | 94.98 | 94.87 |

Table 16: Tracking performance of the final Kalman filter-based method compared with baseline methods (Nearest Neighbour and Hungarian algorithm) when applied to the SL-95 dataset.

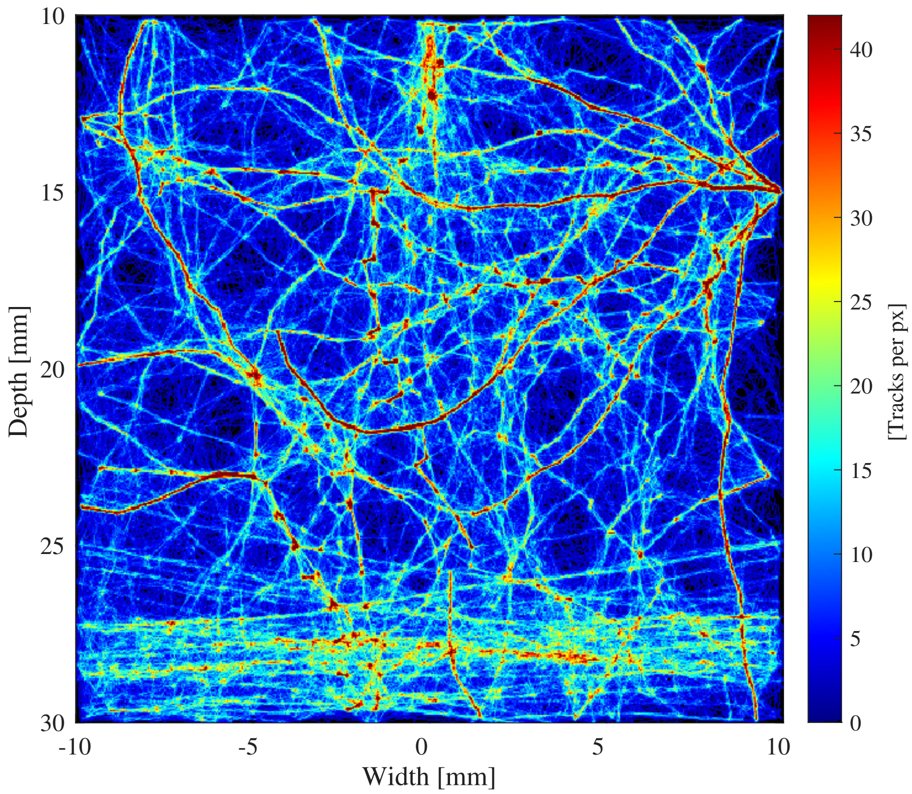



(a) Track density map.

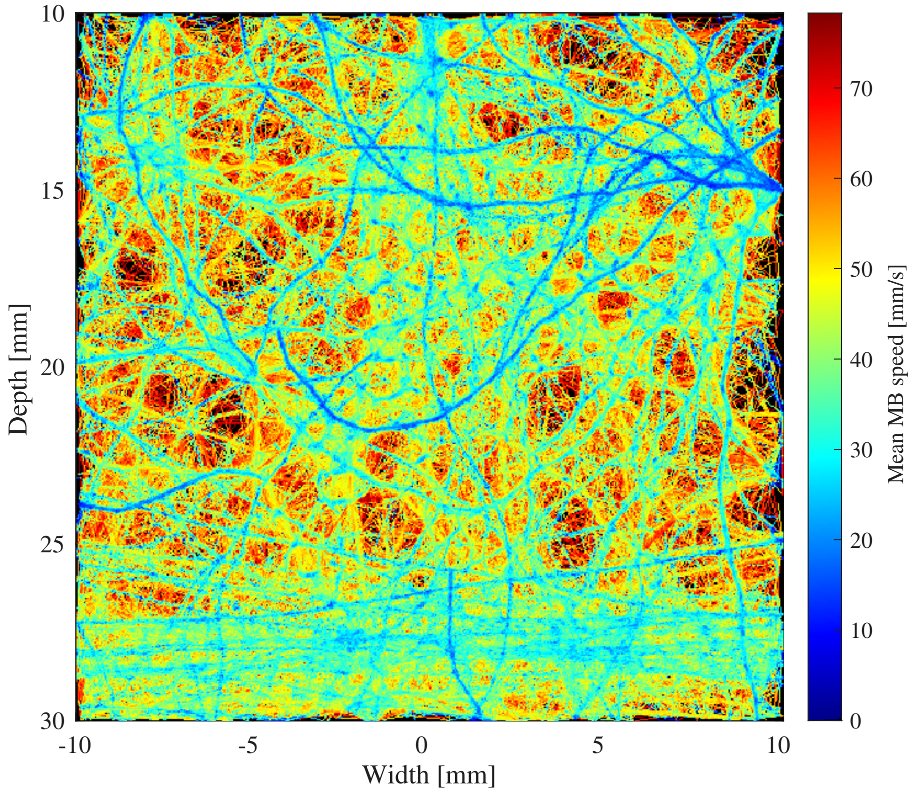



(b) Mean MB speed map.

Figure 17: Track density and velocity maps generated from tracking results of the Nearest Neighbour method on the SL-95 dataset. (a) Number of tracks per pixel. (b) Mean MB speed per pixel.




(a) Track density map.




(b) Mean MB speed map.

Figure 18: Track density and velocity maps generated from tracking results of the Hungarian algorithm method on the SL-95 dataset. (a) Number of tracks per pixel. (b) Mean MB speed per pixel.