

Visual Tracking of Magnetic Microrobots in Biomedical Research Applications

B. (Boudewijn) van den Berg

BSc Report
Biomedical Engineering

Committee:

Dr. S. Misra
Dr. L. A. Sanchez
Prof. Dr. ir. C. H. Slump
Dr. I. Vellekoop

August 2014

Report nr. 011RAM2014
Robotics and Mechatronics
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Bachelor's thesis

Title: Visual Tracking of Magnetic Microrobots in Biomedical Research Applications

Author: Boudewijn van den Berg

Student nr.: S 1219766

Date: 26-08-2014

Group: Robotics and Mechatronics

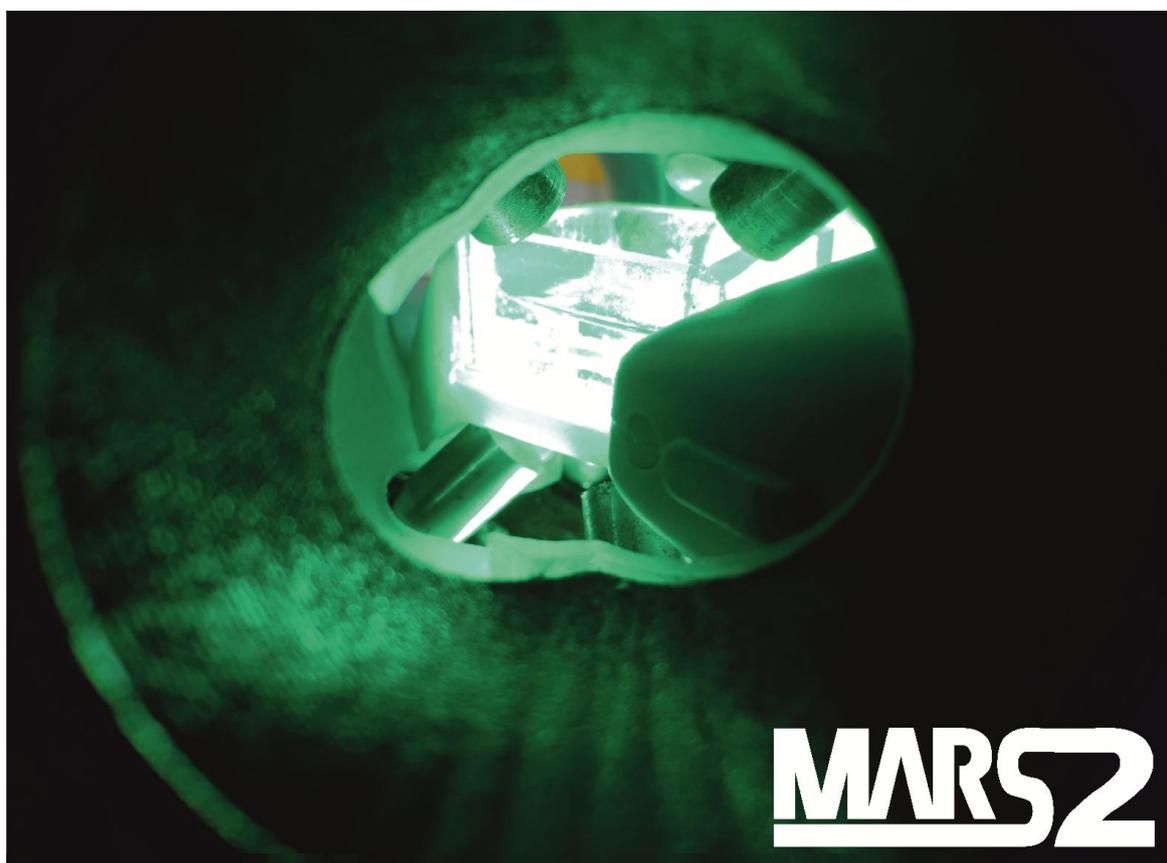
Comittee:

Dr. Sarthak Misra (chairman)

Dr. Alonso Sanchez (supervisor)

Prof. Dr. ir. Kees Slump

Dr. Ivo Vellekoop



**Faculteit Technische Natuurwetenschappen
Opleiding BSc Biomedische Technologie**

UNIVERSITEIT TWENTE.

Visual Tracking of Magnetic Microrobots in Biomedical Research Applications

Boudewijn van den Berg, Alonso Sanchez and Sarthak Misra

Abstract—Magnetic microrobots provide new opportunities in the fields of biological micromanipulation and minimally invasive surgery (e.g., micromanipulation of cells [2], ophthalmic microsurgery [3] and development of new drugs [4]). A major requirement for implementation of these microrobotic applications is precise control of the microrobots. To enable precise control, an accurate and robust position estimation is essential. Therefore, position estimation of microrobots using microscopy and ultrasound equipment is outlined in this work and implemented into a C++ object-oriented platform-independent library. Using the developed system, tracking of multiple types of microrobots using optical microscopy images is demonstrated, i.e., Janus particles are tracked with a precision of $\pm 3.1 \mu\text{m}$, microparticles with a precision of $\pm 14.6 \mu\text{m}$ and microjets with a precision of $\pm 90.4 \mu\text{m}$. Furthermore, microrobot tracking using B-mode ultrasound images is demonstrated, i.e., microparticles with a precision of $\pm 121.0 \mu\text{m}$ and microjets with a precision of $\pm 242.4 \mu\text{m}$. The algorithm is made to enable simultaneous tracking of multiple microrobots in a clutter environment (with other microrobots or debris). Finally, by combining the 2D tracking algorithm with an object matching algorithm 3D tracking of microparticles in a microscope is demonstrated and by combining the 2D tracking algorithm with a scanning algorithm 3D tracking of microparticles in ultrasound images is demonstrated (ultrasound tracking precision: $\pm 176.3 \mu\text{m}$, microscope tracking precision: $\pm 21.6 \mu\text{m}$). This new tracking system could be used for accurate position feedback in the control of magnetic microrobots in biomedical research applications (e.g. micromanipulation of cells [2]).

I. INTRODUCTION

A. State of the art

Recent developments in the design and control of microrobots have led to promising applications: Magnetic microjets have been used to assemble multiple micro-objects [1], transport cells [2], a five-degree-of-freedom magnetic microrobot was able to puncture vasculature in the rabbit's eye, and potential drug delivery applications are being studied [3]. In addition to these recent applications, possible uses of microrobotics in biomedicine include brachytherapy, biopsy, micromanipulation of cells, *in vitro* diagnostics and pharmacological development of new drugs [4]. However, challenges in control, localization, communication and design have to be tackled in order to make progression towards final deployment of this technology [5].

This work was supported by funds from the Netherlands Organization for Scientific Research (NWO) Innovative Medical Devices Initiative (IMDI) - Project: USE (Ultrasound Enhancement).

All authors are affiliated with MIRA-Institute for Biomedical Technology and Technical Medicine, University of Twente, 7500 AE Enschede, The Netherlands

A major requirement for controlling microrobots is visual servoing. Using visual feedback from a microscope, it is possible to servo individual magnetic microparticles in 2D with a positioning error of $27 \mu\text{m/s}$ up to an average velocity of $283 \mu\text{m/s}$ using PI control [9]. With magnetic

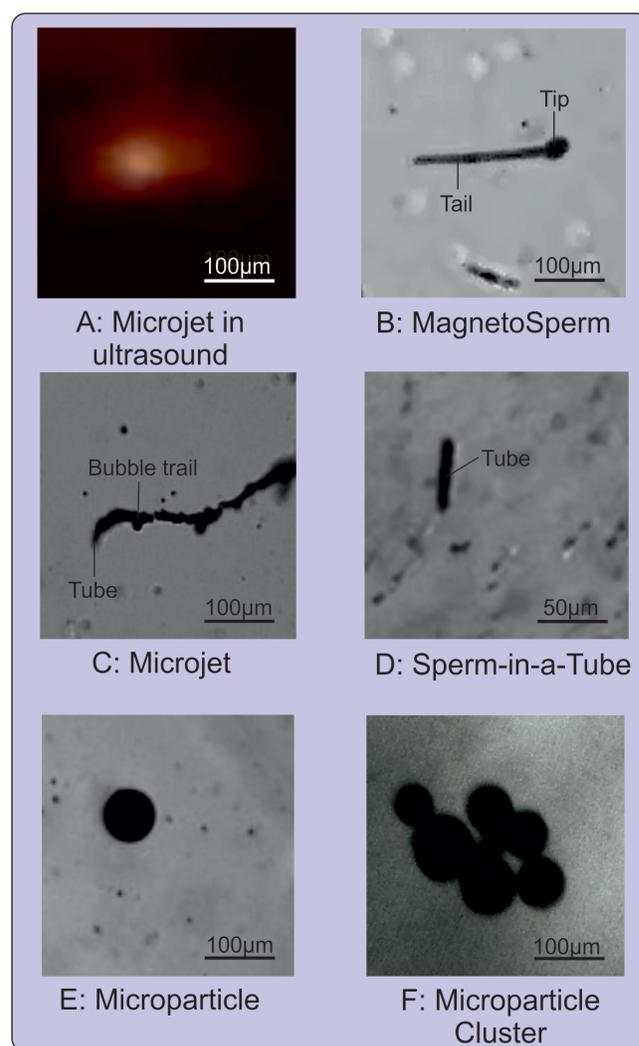


Fig. 1. Multiple types of magnetic microrobots. A: Microjet [6] in an ultrasound image (other microrobots that have been measured on ultrasound, microparticles and magnetosperm, have a similar appearance). B: MagnetoSperm [7] in a microscope. C: Microjet [6] in a microscope. The microjet itself is in the tip of the bubble trail. D: Sperm-in-a-Tube [8]. This microrobot consists of the same type of tube as is used for microjets. However, this tube is propelled by a spermcell which is trapped inside. E: Microparticle. F: Microparticle Cluster. These clusters are formed by microparticles under the influence of a magnetic field.

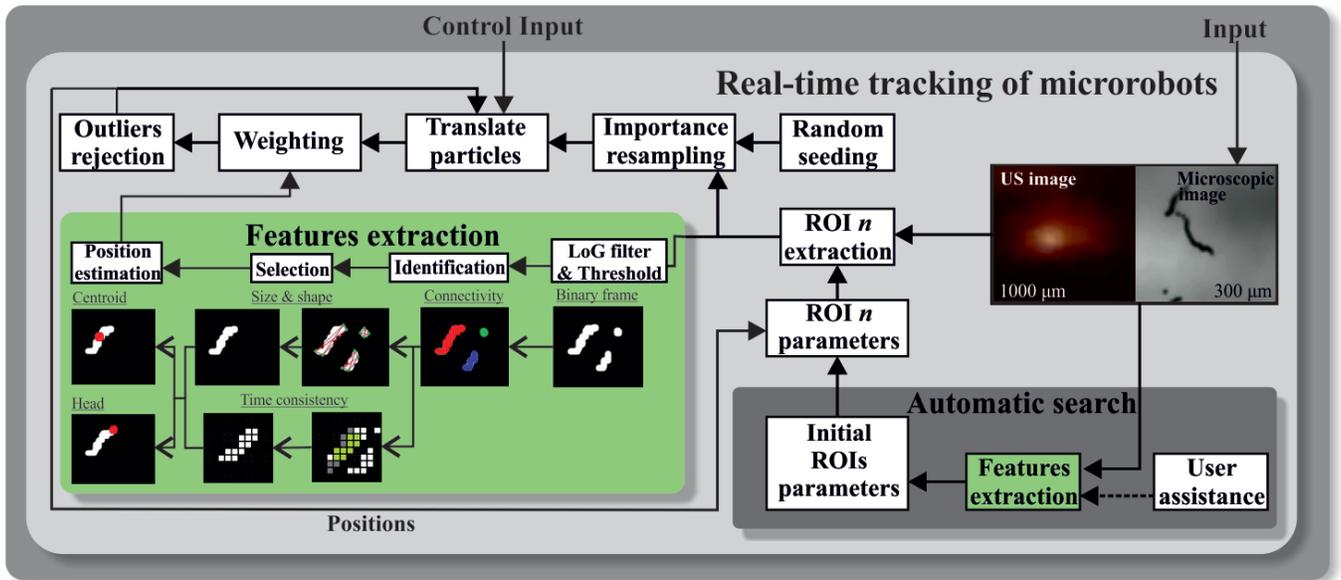


Fig. 2. Flow chart of the autosearch and tracking algorithm. All steps can be specifically tuned to optimize tracking of a specific type of microrobot. Microrobots can be identified by user input or by the autosearch. After the object identification tracking is initialized, after which object position is estimated for every frame using a particle filter. Both the autosearch as well as the tracker use features extraction to estimate object positions based on the measurement. In this features extraction the frame is filtered by a Laplacian of Gaussian (LoG) filter, thresholded and labelled. Subsequently, the automatic search selects objects based on size and shape and the tracker selects objects based on temporal consistency, after which the position of an object can be estimated. The tracker uses this position estimation and a position estimation based on optical flow to weight particles of the particle filter. After weighting the particles are used again for position estimation in the next frame. To do this the particles are resampled based on their weights and translated based on the measured object velocity after which new weights can be assigned.

microjets, a type of magnetic microrobots that requires very little magnetic field due to its self-propelling force (average velocity: $115 \mu\text{m/s}$), a controller positioning error of $356 \mu\text{m}$ is reached using sliding-mode control in 2D [10]. Also, using visual feedback, it is possible to control multiple microrobots with different geometries along individual paths [11]. However, in a lot of applications of microrobots (i.e. targeted drug delivery and minimally invasive surgical interventions), it is not feasible to obtain visual measurements using a microscope. Therefore, other visual feedback methods are used for the control of microrobots in these applications. Two major imaging modalities which offer the opportunity to localize individual microrobots inside opaque objects are ultrasound and magnetic resonance (MR) imaging. Using MR based measurements, it is possible to control magnetic microparticles inside capillary-sized flow channels [12]. However, the delay in MR based systems might lead to control instability and a limited bandwidth of real-time controller implementation. Also the implementation costs of an MR based localization of microrobots is a major disadvantage of this modality [13]. Therefore ultrasound (US) has recently been implemented to provide visual feedback for the control of microrobots. Using ultrasound position measurements, Khalil et al. succeeded in controlling magnetic microparticles along an S-curve with a positioning error of $22.0 \mu\text{m/s}$ [13]. Later on, Sanchez et al. used a similar control method to steer microjets in an eight shape with a positioning error of $183.2 \mu\text{m}$ with an average velocity of $207 \mu\text{m/s}$ [14].

The positioning error in previous control systems is mainly due to the inaccuracy of the visual feedback [14]. The visual feedback is provided by a tracking algorithm, and uncertainties in the position and speed estimations of this algorithm lead to this inaccuracy. For all types of visual feedback, the tracker might suffer from inaccurate position measurements due to a lack of robustness. In ultrasound-based visual feedback, the lack of resolution also is a major factor affecting the inaccuracy. This results in a less accurate measurement of position and it also induces a delay in the registration of changes in direction and speed. Furthermore imaging artifacts, other objects in the workspace and an image background with many features make it harder to recognize and estimate the real position of a microrobot while travelling through a sample. All these factors represent a real challenge for visual servoing inside human tissue.

The tracking methods used by Khalil et al. and Sanchez et al. both use low-level image processing algorithms, similar to an algorithm used in 2004 by Yesin et al. [15] to track intraocular microrobots. In these algorithms, the frames are first equalized and thresholded and then eroded and dilated, which leads to localization of the microjets in low-noise frames of a clean environment with microrobots [14,15]. However these algorithms are not robust in the presence of background intensity variations and imaging artefacts leading to estimation errors. Consequently, frequent human input during tracking is necessary to adjust the estimated position.

Other tracking algorithms have been developed for the tracking of other types of microrobots and to track the migration of cells. For tracking of a microrobot on MRI

Felfoul et al. [16] used correlation in the frequency domain with a mask representing the microrobot shape to estimate the position of the microrobot with a standard deviation (STD) of 542 μm . Buerkle et al. [17] used another correlation technique using pairwise geometric histograms (PGH) to track microrobots under a microscope. However, these techniques are restricted to tracking invariant shapes due to the use of correlation. To track migrating cells Jiang et al. [18] used scale invariant feature transforms (SIFT) to register the movement of features on the cells. This technique is less sensitive to changes in shape, however it required some individual features based on which objects could be tracked. Chatterjee et al. [19] used bipartite graphs to track cells. For this method the objects should also contain properties based on which they can be distinguished from other objects. Another approach which is used for tracking of cells is the use of sequential methods, in particular particle filters. Xiuzhuang et al. [20] used a particle filter based on the mean shift of the object in the image, which was determined by the shift of pixel intensities, to track sperm cells. Smal et al. [21] used a particle filter algorithm that integrated dynamical properties in its position estimation to robustly track large groups of fluorescent proteins on microtubules. This latter method can also be tuned to use other properties in its estimation of the trajectory of the object and integrate multi-sensor input, which makes this method really flexible in its application.

B. Contributions

To cope with the aforementioned tracking and estimation problems and reduce the controller positioning error of magnetic microrobotic control systems a general algorithm for tracking of microrobots is studied and developed in this paper. Because this algorithm should be able to track a variety of microrobots, it should robustly track inconsistent shapes (e.g. microrobots in ultrasound images, bubble tails of microjets or swimming sperm on a microscope). Furthermore, it should be robust to the proximity of other microrobots. Therefore a particle filter¹ is used as a main component in this algorithm to track the microrobots. To enable tracking of multiple microrobots neglecting other objects, an automatic microrobot search function is included in the system, which selects objects with a high likelihood to be microrobots by using a connectivity algorithm to identify objects and exclude objects from the algorithm based on size and shape. To increase robustness, the particle filter uses a more elaborate estimation of the object position compared to Smal et al. and Zhou et al. by combining information from the filtered measurement with optical flow feature tracking (fig. 7) and statistical analysis of the position and speed of the microrobot.

In the next section, the design of the tracking algorithm will be outlined. Subsequently, the tracking algorithm is evaluated based on its performance using real-time microscope

¹For a comprehensive explanation on particle filters, please read "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking" by Arulampalam et al. [22].

Algorithm 1: Connectivity Labelling

- The function $zeros()$ returns an array of zeros.
 - The function $Max()$ returns the maximum of the array.
 - $CSize$ is the size of the connectivity region, defined as the distance from the midpoint to the border.
 - $MaxCR$ is the maximum value within an extracted region of the size $CSize$.
 - $ObjectNumber$ is the number assigned to an identified object.
-

Initialize binary frame:

$BinaryFrame = zeros(framesize)$

Initialize maximum object number: $MaxObjects = B$

Initialize connectivity region size: $CSize = C$

```

for  $i = CSize$  to  $Height - Csize$  do
  for  $j = CSize$  to  $Width - Csize$  do
    if  $BinaryFrame(i, j) = 1$  then
       $MaxCR = Max(BinaryFrame($ 
         $i - Csize$  to  $i + Csize, j - Csize$  to  $j +$ 
         $Csize))$ 
      if  $MaxCR > 1$  then
         $ObjectNumber = MaxCR$ 
      end
      if  $MaxCR == 1$  then
         $ObjectNumber = MaxObjects - 1$ 
      end
      for  $k = i - CSize$  to  $i + CSize$  do
        for  $l = j - CSize$  to  $j + CSize$  do
          if  $BinaryFrame(k, l) = 1$  then
             $ObjectFrame(k, l) =$ 
               $ObjectNumber$ 
          end
        end
      end
    end
  end
end

```

and ultrasound input of samples containing microparticles, microjets and Janus particles.

II. TRACKING ALGORITHM

Although different types of microrobots each have individual constraints on a tracking algorithm, a general tracking strategy that can be implemented on all types of microrobots is defined in this work (fig. 2). Within this general tracking strategy, individual steps can be adapted to create a robust tracking algorithm for a specific type of microrobot.

In this work steps have been adapted for multiple types of microrobots, including:

- Microjets
- Microparticles
- Janus particles

For this selection, a single configuration is used for microparticles and Janus particles on ultrasound since no

TABLE I

THE CONFIGURABLE PARAMETERS OF THE ALGORITHM AND THE VALUES USED FOR TRACKING OF SPECIFIC MICROROBOTS.

| Parameters | Dependent On | Effect | Microjets | Microparticles | Janus Particles |
|---------------------------|---------------------------------------|--|---------------|----------------|-----------------|
| Autosearch: | | | | | |
| Kernel size | Robot size and kernel STD | Makes filter sensitive for objects of a certain size. | 15 pix. | 51 pix. | 51 pix. |
| Kernel STD | Robot size and kernel size | Makes filter sensitive for objects of a certain size. | 3.0 pix. | 3.0 pix. | 3.0 pix. |
| Threshold size | Robot size and contrast | Makes threshold sensitive for objects of a certain size. | 151 pix. | 151 pix. | 151 pix. |
| Threshold height | Robot brightness and contrast | Filters out objects that are not bright enough. | -60 | -100 | -100 |
| Min. and max. object size | Robot size | Filters out objects which are too big or too small. | 100, 400 pix. | 80, 150 pix. | 80, 150 pix. |
| Min. and max. shape ratio | Robot shape | Filters out objects based on shape. | 3, 100 | 0, 3 | 0, 3 |
| Tracker: | | | | | |
| ROI size factor | Robot shape | Determines the relative size of the ROI with respect to the object size. | 0.009 | 0.001 | 0.001 |
| Min. and max. ROI size | Variety in robot sizes | Restricts the ROI size. | 60, 70 pix. | 18, 30 pix. | 18, 30 pix. |
| Number of particles | Computation power | Influences accuracy and robustness of the tracker. | 200 | 200 | 200 |
| Averaging time | Consistency of the measurement | Provides consistency in the speed measurement. Introduces delay. | 40 frames | 3 frames | 3 frames |
| Optical flow weight | Reliability of optical flow estimate | Changes the relative weight of the optical flow probability density. | 1.0 | 1.0 | 0.5 |
| Head/centroid weight | Reliability of head/centroid estimate | Changes the relative weight of the head/centroid probability density. | 1.0 | 4.0 | 4.0 |

distinction can be made between the different types of microrobots on ultrasound because of their similar appearance. The algorithm can be adapted to other specific types of microrobots by changing the parameters in table 1. The tracking of microjets on ultrasound and on microscope includes a special head-tracking function to differentiate the robot from its bubble trail.

A. Automatic Search

Before tracking can be executed, potential microrobots must be identified. Even though user input identifying a single microrobot might be sufficient information to initialize individual tracking, a more general selection strategy needs to be defined to provide a reliable search algorithm for multi-object tracking scenarios, given that all microrobots in the frame need to be selected for tracking.

An important first step of both the automatic search and the tracking is the pre-processing of the image. The purpose of this pre-processing is to make the algorithm less sensitive to artifacts and noise, including shadows, reflections and features outside the focus of the microscope, and to make the algorithm more sensitive to the features of the microrobot itself. Therefore the frames should be filtered with a filter that is only sensitive to spatial intensity fluctuations of the size of the microrobotic features. To realize this spatial sensitivity the frames are filtered with a Laplacian of Gaussian (LoG) filter, in which the Laplacian aspect makes the filter sensitive to intensity variations and the Gaussian aspect determines the size of the intensity variations to which the filter is sensitive. A specific filter for every type of microrobot can be designed based the approximate size of the microrobot in the frame (in pixels), by choosing a corresponding STD of the LoG filter kernel. Next, the filter kernel can be generated using equation 1, in which $LoG(0, 0)$ is the midpoint of the kernel.

$$LoG(x, y) = \frac{1}{\pi\sigma^4} \left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

To enable labelling of the objects, the frame is converted to a binary frame. A consistent method to do this, which is insensitive to inter frame intensity fluctuations and sensitive

to the size of the objects, is converting the image using an adaptive threshold.

$$T(x, y) = Threshold(x, y) \quad (2)$$

$$BinaryFrame(x, y) = \begin{cases} 0 & \text{if } Frame(x, y) < T(x, y) \\ 1 & \text{if } Frame(x, y) > T(x, y) \end{cases} \quad (3)$$

This threshold is set for every pixel by calculating a Gaussian weighted average of the pixel values in an area around the pixel $Frame(x, y)$ of which the size is determined by a configurable parameter $ThresholdSize$ and adding or subtracting a constant value from this average. By adjusting the constant value, this method can be enhanced for individual types of microrobots. Hereafter, because objects can consist of multiple blobs, inter-blob connectivity within the frame is analyzed to identify and label the objects. This is done by algorithm 1, in which the connectivity region size is set to the maximum distance between two blobs of the same object.

When the objects are identified and labelled, a selection can be made of the objects which are most likely to be the

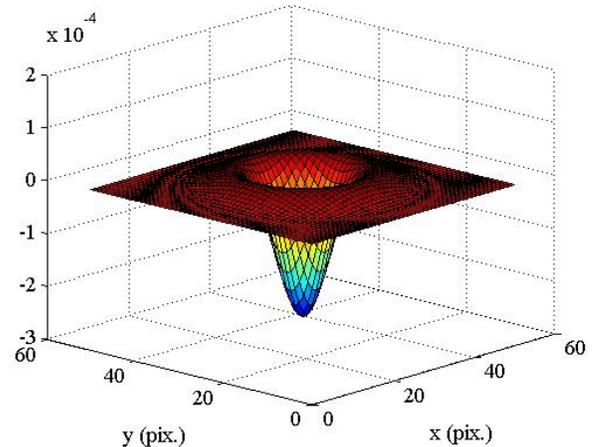


Fig. 3. A visualization of the LoG kernel. Due to its shape, it is sensitive to features of a specific size.

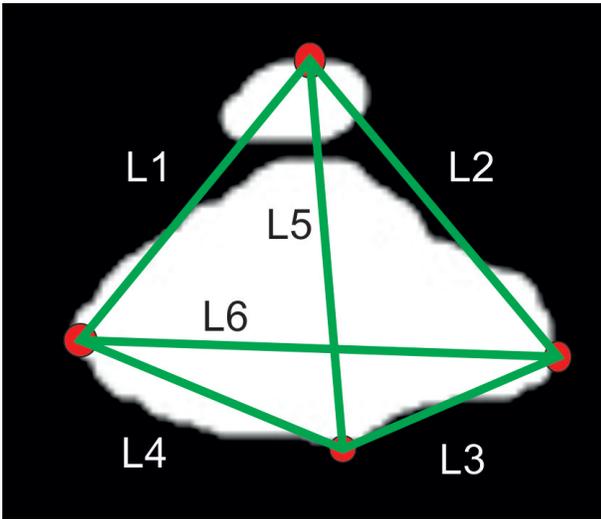


Fig. 4. A visualization of the shape analysis algorithm applied to an ultrasound blob of a microjet. The algorithm measures the distances between the points, determines the variance in those distances and divides this variance by the amount of pixels belonging to the object. Characteristic for moving microjets in ultrasound images is the side blob, which appears if the jet has a long trail of bubbles.

specified type of microrobots. Given that the microscopic and ultrasound appearance of most microrobots is respectively a silhouette and a reflection, this selection is based on their size and shape. The size is determined by counting the pixels of the object, and is only dependent on the real size of the objects and the scale of the frame. To select the right shapes, a simple but effective algorithm was designed for this purpose (algorithm 2). The algorithm determines the amount of white pixels and the coordinates of the highest, lowest and side points of every object. The variance in the length of the lines between these points, as drawn in figure 4, indicates a shape. Complementary to this, the pixel count indicates to which extend this shape is filled. Subsequently a measure for the shape of the object follows from the ratio between the variance and the pixel count (fig. 4 and fig. 5).

Since the objects have labelled coordinates in the frame, the blobs that do not match the given shape, or are bigger or smaller than respectively a given maximum or minimum size, can be removed from the binary image. The remaining objects can be relabeled and for tracking the coordinates of their centroids, determined by the average of the pixel coordinates of the object, can be stored in a vector which will be used to extract a region of interest (ROI) around every object being tracked.

B. Tracker

Now that the coordinates of the objects are known, the algorithm starts tracking these objects. To reduce computational complexity and, hence, achieve high tracking framerates, a ROI is chosen around each tracked object in which the analysis of the measurement can be implemented. The size of this ROI is determined by multiplying the size of the object with a factor dependent on the type of microrobot. The factor is chosen to make the ROI fully enclose the microrobot and

Algorithm 2: Size and Shape Selection

- The function *EuclideanDist()* returns the euclidean distance between two points.
 - *CSize* is the size of the connectivity region, defined as the distance from the midpoint to the border.
 - *Xmax*, *Xmin*, *Ymax* and *Ymin* are the positions of the minimum and maximum coordinates of the object.
 - *count* is the number of pixels of the object.
 - *Ratio* is a ratio indicating the shape of the object.
-

```

for k = 0 to MaxObjects do
  Initialize count = 0; Initialize Xmax.x = 0;
  Initialize Ymax.y = 0; Initialize
  Xmin.x = framewidth; Initialize
  Ymin.x = frameheight;
  for i = CSize to Height - CSize do
    for j = CSize to Width - CSize do
      if ObjectFrame(i, j) = k then
        count = count + 1;
        if j > Xmax.x then
          | Xmax.x = j; Xmax.y = i;
        end
        if j < Xmin.x then
          | Xmin.x = j; Xmin.y = i;
        end
        if i < Ymax.y then
          | Ymax.x = j; Ymax.y = i;
        end
        if i < Ymin.y then
          | Ymin.x = j; Ymin.y = i;
        end
      end
    end
  end
  L1 = EuclideanDist(Xmin to Ymin);
  L2 = EuclideanDist(Ymin to Ymax);
  L3 = EuclideanDist(Xmax to Ymax);
  L4 = EuclideanDist(Ymax to Xmin);
  L5 = EuclideanDist(Ymax to Ymin);
  L6 = EuclideanDist(Xmax to Xmin);
  LVar = Variance(L1, L2, L3, L4);
  Ratio = LVar / count;
  if count > MaxSize
  or count < MinSize
  or Ratio > MaxRatio
  or Ratio < MinRatio then
    for i = CSize to Height - CSize do
      for j = CSize to Width - CSize do
        if ObjectFrame(i, j) = k then
          | ObjectFrame(i, j) = 0;
        end
      end
    end
  end
end
end

```

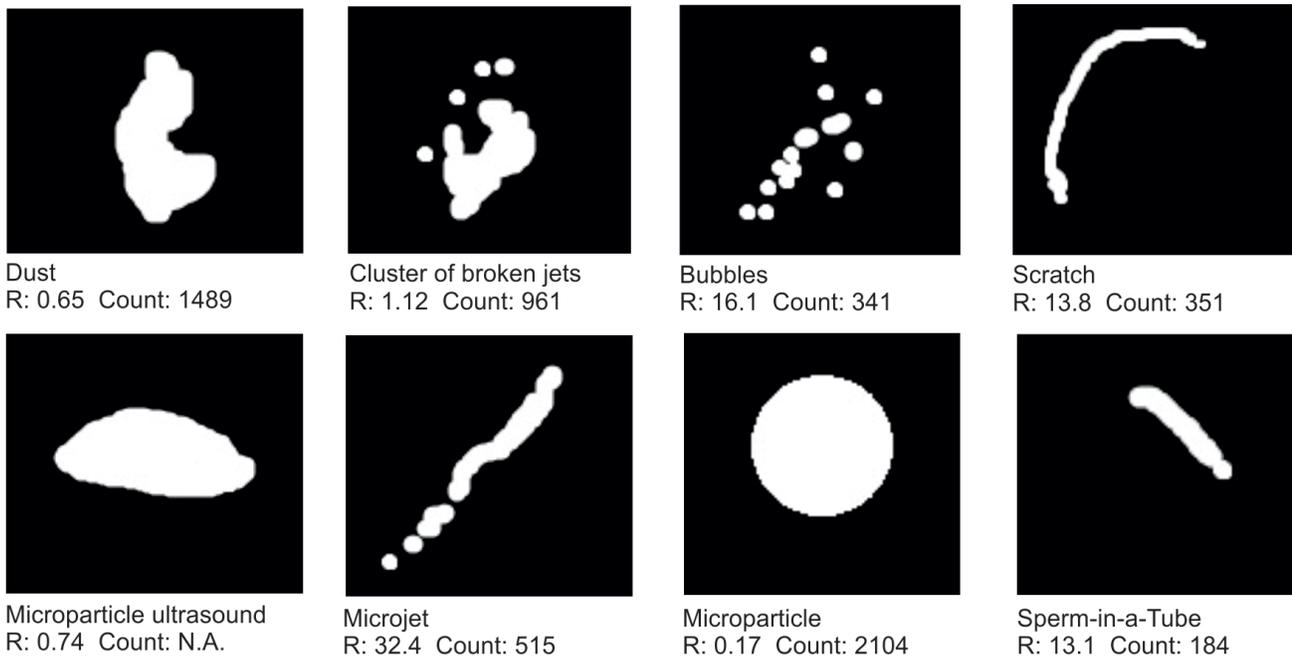


Fig. 5. The shape analysis algorithm applied to multiple types of blobs. The shape ratio is indicated by "R" and the number of pixels of the object by "count". The algorithm clearly distinguishes round shapes (microparticle) from irregular shapes (dust) and irregular shapes from long shapes (microjet, sperm-in-a-tube, scratch). Upper row: Binary blobs of objects that might interfere with the tracking. Bottom row: Binary blobs of multiple microrobots. Every blob is drawn on the same scale (one picture is approximately $200\mu\text{m}$ by $200\mu\text{m}$ except the picture of the microrobot in an ultrasound image).

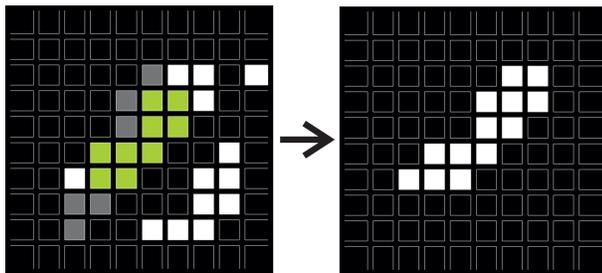


Fig. 6. Illustration of time consistency based object selection. In the first picture an overlay of the new binary ROI and the old binary ROI (in which other objects already have been filtered out by the selection) has been shown.

- White pixels from the old ROI are gray.
- White pixels from the new ROI are white.
- Overlapping white pixels are green.

As the middle object is the object with the most overlapping pixels, it is selected (second picture).

exclude as much of the rest of the frame as possible. In this way, the ROI includes all features necessary for the analysis and is as small as possible.

When the ROI has been selected the corresponding region is extracted from the filtered frame. The shape and size analysis steps from the autosearch are excluded to prevent unwanted loss of the object due to changes in shape. Instead, the binary frame is used to select the object that is temporally most consistent. This is done by segmenting the binary ROI using algorithm 2 and counting overlapping pixels between the filtered binary ROI from last iteration and binary ROI in the current iteration. This is illustrated in figure 6.

From the remaining object inside the ROI, the object

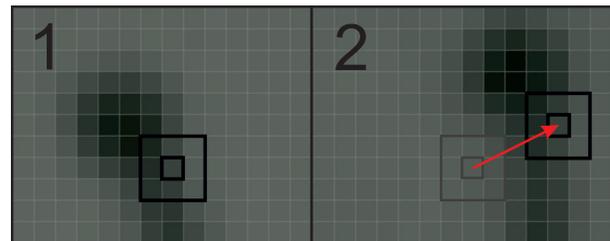


Fig. 7. Simplistic visualization of optical flow feature tracking. The most likely translation of a pixel is calculated by searching the least square fit of the color intensities of the 9 surrounding pixels in the area.

position is measured. For most microrobots, the measured position is most effectively defined as the centroid of the object, as this is the point which is dynamically most stable. However, for microjets an algorithm for headtracking is used, as these robots are at the tip of a trail of bubbles. This algorithm uses the measured speed of the jet and, if it is available, the control input to determine the first pixel of the object in the direction of movement, which provides an indication of the robot position.

Although this object analysis can already provide a good estimation of the position of some types of microrobots in case of high contrast frames (i.e. magnetic microparticles on a microscope, figure 1E), the tracker should also be robust in less reliable measurements. When a microrobot temporarily disappears from the frame, as is often the case with ultrasound measurements of microjets, the tracker should still make a reliable estimation of the position as this position is important for accurate control of the robot. Also, when

Algorithm 3: Head Tracking

- *ControlSetpoint* is the setpoint that is used for position control, if the is controlinput.
- *Searchpoint* is a point in front of the microrobot .
- *speed.x* and *speed.y* are the velocity of the object in respectively the X direction and Y direction.
- *EuclideanDist()* is a function returning the Euclidean distance between two points.
- *Head* is the position of the front pixel of the microrobot.

```
if ControlInput = true then
  | Searchpoint = ControlSetpoint
else
  | Searchpoint.x = C * speed.x
  | in which C * Xspeed >> speed.x
  | Searchpoint.y = C * speed.y
  | in which C * Yspeed >> speed.y
end
for i = 0 to ROIsize do
  for j = 0 to ROIsize do
    if BinaryROI(i, j) = 1 then
      | Dist =
      | EuclideanDist(Point(i, j) to SearchPoint)
      | if Dist < MinDist then
      | | MinDist = Dist
      | | Head.x = j
      | | Head.y = i
      | end
    end
  end
end
end
```

the measurement has a low resolution, the estimation of the position should still be as accurate as possible. Because the tracking algorithm is designed for usage in biomedical applications, the tracker should also be robust to features from surrounding tissue and other objects. To do this robust and accurate tracking a sequential importance sampling particle filter is implemented.

In the first cycle the particles of the particlefilter are seeded with a Gaussian distribution, based on the approximate speed of the type of microrobot, around the centroid. This is done with independent distributions in x - and y -direction, because the direction is not known yet. The distributions are generated by taking the Box-Muller transform of random uniformly distributed numbers U_1 and U_2 , to obtain the standard normal distribution, and adjusting the mean and STD:

$$\text{Particle set: } \mathbf{N} = \begin{pmatrix} \mathbf{N}_{\text{pos}} \\ \mathbf{N}_{\text{weights}} \end{pmatrix} \quad (4)$$

$$\mathbf{N}_{\text{pos}} = \begin{bmatrix} X_{\text{centroid}} + \sqrt{-2\ln(U_1)} \cos(2\pi U_2) \sigma_{\text{speed}} \\ Y_{\text{centroid}} + \sqrt{-2\ln(U_1)} \sin(2\pi U_2) \sigma_{\text{speed}} \end{bmatrix} \quad (5)$$

$$\mathbf{N}_{\text{weights}} = 1 \quad (6)$$

Algorithm 4: Importance Resampling

- $n_{\text{particles}}$ is the number of particles.
- n_{split} is the number of particles that are resampled from the selected particle.
- *Totalweight* is the sum of all particle weights.
- *MeanWeight* is the average of all particle weights.
- \mathbf{N}_{pos} is the vector with particle positions.
- $\mathbf{N}_{\text{weights}}$ is the vector with particle weights.

$$\text{Totalweight} = \sum \mathbf{N}_{\text{weights}}$$

$$\text{MeanWeight} = \frac{\text{Totalweight}}{n_{\text{particles}}}$$

Sort particle set from high to low values based on particle weights:

Sort($\mathbf{N}, \mathbf{N}_{\text{weights}}$)

Split particles into new particles and sample until $\text{length}(\mathbf{N}) = n_{\text{particles}}$:

```
for i = 0 to nparticles do
  | nsplit = Nweights(i)/MeanWeight
  | count = count + nsplit
  | if count <= nparticles then
  | | for j = count - nsplit to count do
  | | | Npos,new(j) = Npos(i)
  | | | Nweights,new(j) = 1
  | | end
  | | else if count - nsplit <= nparticles then
  | | | nsplit = nparticles - (count - nsplit)
  | | | for j = count - nsplit to nparticles do
  | | | | Npos,new(j) = Npos(i)
  | | | | Nweights,new(j) = 1
  | | | end
  | | end
end
end
```

The positions of these particles are given by sub-resolution coordinates, which enables the particle filter to make a sub-resolution estimation of the particles position later on. Hereafter, these particles are updated every frame based on the average centroid or head speed, optical flow, and the centroid or head in the current frame. This is done by implementing the following steps sequentially:

1) *Importance Resampling*: At the importance resampling step, the weighted particles are split into particles of equal weight. Because the number of particles is kept equal, particles with low weights are excluded and particles with high weights result in a stack of multiple particles. The resampling method is outlined in algorithm 5.

2) *Particle Translation*: The position of these particles is updated based on a probability distribution of the speed. Because the speed of most microrobots is highly influenced by both external (i.e. temperature, air flux) and internal (i.e. individual robot behavior, field fluctuations) factors a real-time measurement is used to obtain a more accurate

Algorithm 5: Particle Translation

- σ_{speed} is the measured STD of the speed.
 - U is a uniform distributed random number.
 - n is a normal distributed random number.
 - ρ is the measured correlation coefficient between the velocities in x and y direction.
-

```
for  $i = 0$  to  $n_{particles}$  do
   $U_1 = rand()$ 
   $U_2 = rand()$ 

  Box-Muller transform:
   $n_1 = \sqrt{-2 \ln(U_1)} \cos(2\pi U_2)$ 
   $n_{2uncorr.} = \sqrt{-2 \ln(U_1)} \sin(2\pi U_2)$ 
  Correlation:
   $n_2 = \rho n_1 + \sqrt{1 - \rho^2} n_{2uncorr.}$ 

  Determining contribution of controlinput:
  if  $controlinput == true$  then
     $Cspeed.x = \frac{(setpoint.x - Pos.x)C}{|Pos.x - setpoint.x| + |Pos.y - setpoint.y|}$ 
     $Cspeed.y = \frac{(setpoint.y - Pos.y)C}{|Pos.x - setpoint.x| + |Pos.y - setpoint.y|}$ 
  else
     $controlspeed.x = 0$  and  $controlspeed.y = 0$ 
  end

   $N_{pos} =$ 
   $\begin{bmatrix} N_{pos,new}.x + speed.x + Cspeed.x + n_1\sigma_{speed}.x \\ N_{pos,new}.y + speed.y + Cspeed.y + n_2\sigma_{speed}.y \end{bmatrix}$ 
end
```

distribution. At the initialization of the filter a probability density of the speed of the specific type of microrobot is used, with its maximum at the center because the direction is still unknown. When a control algorithm is active, the control input can be taken into account to reshape the distribution. The algorithm keeps track of the speed and enables more accurate tracking by assuming consistency in speed and translating the particles with a covariant distribution. The translation of particles is outlined in algorithm 6.

3) *Weighting*: To obtain weights for the particles, two methods are used to estimate the object position. This increases the accuracy and improves the robustness. If at one step one method fails to correctly estimate the particle position, the other method can make up for this. In the first iteration the weighting is solely based on the measured centroid of the object. In the other iterations the weighting is based on the optical flow and either the centroid or the head of the object.

The centroid is again estimated by calculating the average pixel coordinates of the binary object and the head is estimated by the head tracking algorithm and for both values a Gaussian probability density function, based on the mean and STD of the measured values, is generated for weighting of the particles. The optical flow estimation is done using the Lucas-Kanade method [23]. In this method, it is assumed that neighboring pixels have similar motion. Therefore the optical

Algorithm 6: Weighting

- P_{of} is the estimated position based on the optical flow.
 - P_c is the position of the centroid or the head.
 - σ_{of} is the STD of the optical flow position estimation.
 - σ_c is the STD of the object based position estimation (head or centroid).
 - $N_{weights}$ is the vector with particle weights.
 - N_{pos} is the vector with particle positions.
-

```
for  $i = 0$  to  $n_{particles}$  do
   $f_{of}(i) =$ 
   $\frac{1}{\sigma_{of}\sqrt{2\pi}} e^{-\frac{1}{2} \left( \left( \frac{N_{pos}.x(i) - P_{of}.x}{\sigma_{of}} \right)^2 + \left( \frac{N_{pos}.y(i) - P_{of}.y}{\sigma_{of}} \right)^2 \right)}$ 
   $f_c(i) =$ 
   $\frac{1}{\sigma_c\sqrt{2\pi}} e^{-\frac{1}{2} \left( \left( \frac{N_{pos}.x(i) - P_c.x}{\sigma_c} \right)^2 + \left( \frac{N_{pos}.y(i) - P_c.y}{\sigma_c} \right)^2 \right)}$ 
   $N_{weights}(i) = f_{of}(i)f_c(i)$ 
end
```

Algorithm 7: Speed Estimation

- *EuclideanDist()* is a function returning the euclidean distance between two points.
 - *MaxSpeed* is the maximum velocity that is set for the speed sampling consensus.
-

```
for every iteration do
  if  $EuclideanDist(Pos(i) \text{ to } Pos(i-1)) <$ 
   $MaxSpeed$  then
     $TempSpeed.x.push(Pos(i).x - Pos(i-1).x)$ 
     $TempSpeed.y.push(Pos(i).y - Pos(i-1).y)$ 
  end

  if  $length(TempSpeed) > SampleNumber$  then
     $TempSpeed.erase(begin)$ 
  end

   $speed.x = Mean(TempSpeed.x)$ 
   $speed.y = Mean(TempSpeed.y)$ 
end
```

flow can be determined by estimating the displacement of these pixels. This is done by applying a least square fit method on partial derivatives of the image intensity.

Based on the optical flow resulting from this equation, the position of the tracked object in the next frame is estimated, generating another Gaussian probability density function.

To update the estimated position the weighted average of the coordinates of the particles is computed. This estimated position is used in the next iteration to update the ROI, and can be used as input for a control algorithm. Every iteration the estimated speed is updated by taking the average over the most recent frames while rejecting outliers:

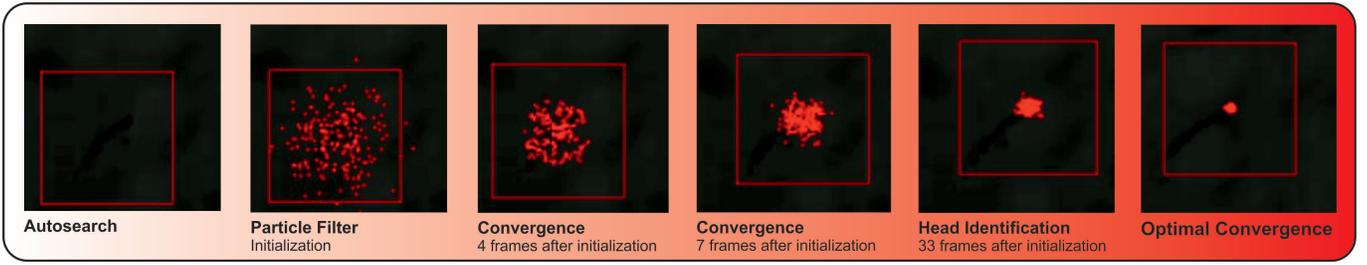


Fig. 8. Image sequence of showing the fast and accurate convergence of the particle filter, in this case applied to microjets. Converged particles provide a more accurate estimation of the position. The framerate is approximately 30 frames per second.

Algorithm 8: 3D Object Matching

- Objects are matched by the vector $BlobMatch$.
- Variables with subscript 1 are derived from input 1.
- Variables with subscript 2 are derived from input 2.

```

for  $i = 0$  to  $ObjectNumber1$  do
   $TempXdistance = 0$ 
  for  $j = 0$  to  $ObjectNumber2$  do
     $TempXdistance =$ 
       $abs(Pos1(i).x - Pos2(j).x)$ 
    if  $TempXdistance < Xdistance(i)$  then
       $Xdistance(i) = TempXdistance$ 
       $BlobMatch(i, 1) = i$ 
       $BlobMatch(i, 2) = j$ 
    end
  end
end
end
if  $ObjectNumber1 > 4$  and  $ObjectNumber2 > 4$  and
 $SingleObjectTracking = false$  then
   $ThresholdMax =$ 
     $Mean(Xdistance) + 2 * STD(Xdistance)$ 
   $ThresholdMin =$ 
     $Mean(Xdistance) - 2 * STD(Xdistance)$ 
else
   $ThresholdMax = MaxError$ 
   $ThresholdMin = 0$ 
end
for  $i = 0$  to  $ObjectNumber1$  do
  if  $Xdistance < ThresholdMin$  or
   $Xdistance(i) > ThresholdMax$  then
     $Xdistance(i) = -1$ 
     $BlobMatch(i, 1) = -1$ 
     $BlobMatch(i, 2) = -1$ 
  end
end
end

```

III. 3D TRACKING

A. Microscopic Tracking Using Orthogonal Viewpoints

For microscopic tracking of microrobots in 3D the 2D tracker is used on input from two microscopes. To achieve 3D tracking these microscopes are positioned perpendicular to each other (fig. 9). In this way two 2D trackers, when run synchronized, can track both the x and y and the x and z coordinates. The output of these two trackers is combined to

a 3D estimation of the position by assuming that the cameras are aligned in the x -direction. Based on this assumption, the tracker can recognize an object in two different frames by searching for objects in both frames with the smallest difference in x -coordinates.

To exclude objects that are not visible in both frames, and therefore cannot be tracked in 2D, matched objects that are not likely to be the same object are not matched and therefore excluded from 3D tracking. In multi-object tracking (when $ObjectNumber1 > 4$ and $ObjectNumber2 > 4$) this is done by comparing the difference in x -coordinates of an object to the mean difference in x -coordinates of all objects, as all objects should approximately have the same alignment error. In single object tracking, or tracking of less than 4 objects, the difference in x -coordinates is compared to a threshold, $MaxError$, which should be set based on the cameras and the set-up (more specific: spatial resolution, camera distance, camera alignment method). When there are multiple objects in one frame with an acceptable difference with one object in the other frame, the coordinates of that object are used to track both objects from the other frame, since the tracker does not have enough information to distinct the objects.

B. Ultrasound Plane Scanning

For ultrasound tracking in 3D a scanning algorithm is implemented to determine the height of the object and send frames of correct height to the 2D tracker. Due to limitations of the set-up to the maximum amplitude of the probe scanning, it is only possible to track one object. To initialize ultrasound tracking in 3D, the object should first be selected manually by scanning the probe over the sample and selecting an object in the 2D tracker. When the object is selected, the tracking and scanning algorithms take over.

The scanning algorithm tracks the height of the object based on the assumption that the object is at its brightest within the ROI when it is optimally positioned within the ultrasound plane. Therefore, when scanning the object with a small amplitude (e.g. in case of microparticles 3mm), the object height can be determined by registering the height of the ultrasound plane at the moment that the object has maximum intensity in the ultrasound frame. To increase the robustness of this algorithm, the ultrasound frames are, similar to the 2D tracking algorithm, filtered with a LoG kernel and thresholded with an adaptive threshold. Because

coordinates using (7), in which $X_{position}$ and $Y_{position}$ are the coordinates of the real position, X_{pixel} and Y_{pixel} are the coordinates of the position in pixels, FOV_x is the field of view in x direction, FOV_y is the field of view in y direction and $width_{frame}$ and $height_{frame}$ are the width and height of the frame in pixels.

$$x_{position} = x_{pixel} \frac{FOV_x}{width_{frame}} \quad (7)$$

$$y_{position} = y_{pixel} \frac{FOV_y}{height_{frame}} \quad (8)$$

Based on these positions the tracking STD, defined as the standard deviation of the set of errors of position estimation

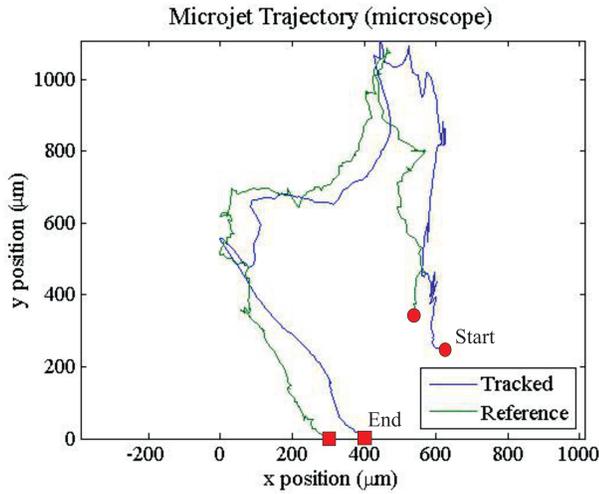


Fig. 11. Trajectory of the microjet in video 3, in which the jet was tracked with a tracking STD of $90.5 \mu m$. At some points in the trajectory there is a tracking offset. This happens because the tracker does not correctly track the tip of the microjets due to variations in speed. In the video it can be seen that there was flow leading to these variations.

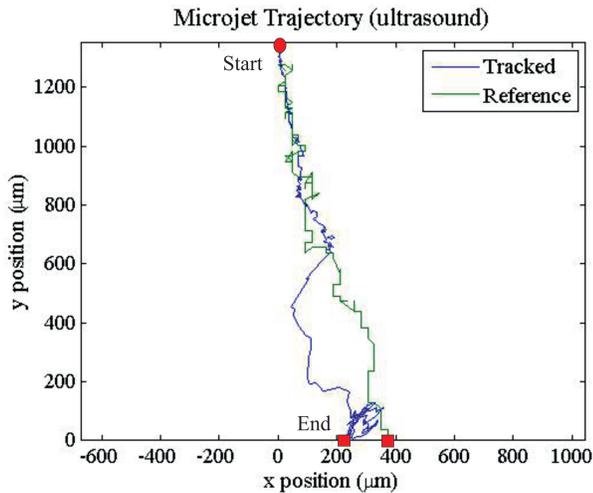
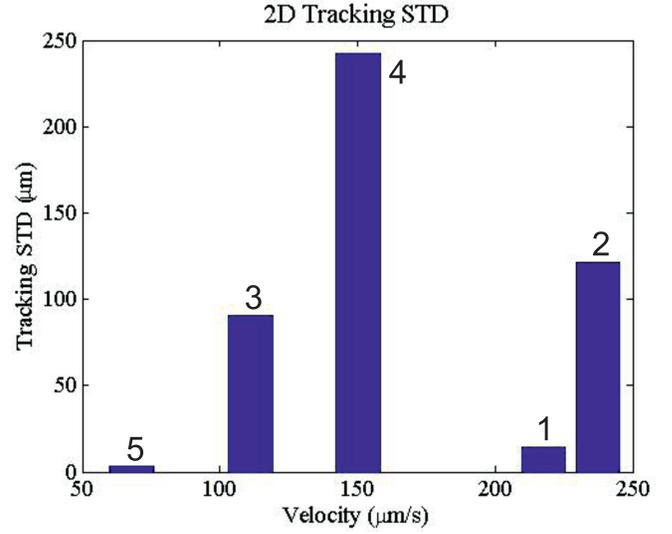


Fig. 12. (Trajectory of the microjet in video 4, which was measured with ultrasound. The jet was tracked with a tracking STD of $242.4 \mu m$. The tracking in this measurement is less accurate due to the low resolution of ultrasound. At the end of the video the headtracking algorithm catches the wrong blob, leading to a high position estimation error.

in each frame [24], and average velocity are determined which are summarized and illustrated in table 2. Since no manual position corrections during tracking were needed in these videos to maintain tracking, these are not shown in the results.

Figure 11 shows the trajectory of the microjet in video 3.



| Video | Name | Average velocity | Tracking STD |
|-------|----------------------------------|------------------|---------------|
| 1 | Microparticle Cluster Microscope | 217.6 $\mu m/s$ | 14.6 μm |
| 2 | Microparticle Cluster Ultrasound | 237.4 $\mu m/s$ | 121.0 μm |
| 3 | Microjet Microscope | 111.2 $\mu m/s$ | 90.4 μm |
| 4 | Microjet Ultrasound | 150.2 $\mu m/s$ | 242.4 μm |
| 5 | Janus Particle Microscope | 67.9 $\mu m/s$ | 3.1 μm |

TABLE II

TO DETERMINE THE 2D TRACKING ACCURACY BOTH THE REAL POSITION, WHICH IS SELECTED MANUALLY, AND THE TRACKER POSITION WERE REGISTERED DURING EACH FRAME THAT THE TRACKER WAS ACTIVE. BASED ON THESE POSITIONS THE TRACKING STD, DEFINED AS THE STANDARD DEVIATION OF THE SET OF ERRORS OF THE POSITION ESTIMATION IN EACH FRAME, AND AVERAGE VELOCITY WERE DETERMINED. THE TRACKING STD IN VIDEO 3 IS RELATIVELY HIGH DUE TO THE LARGE BUBBLE TRAIL. IF THE MICROJET HAS A SMALLER BUBBLE TRAIL THE TRACKING IS MORE ACCURATE. HOWEVER, IN THAT CASE THE MICROJET WILL ALSO BE SLOWER. IN VIDEO 5 THE TRACKING STD IS MAINLY DUE TO THE BUBBLES APPEARING FROM THE JANUS PARTICLE. HOWEVER, THE TRACKING STD IN THIS VIDEO IS MUCH SMALLER THAN THE TRACKING STD OF VIDEO 1 BECAUSE THE SCALE IS MUCH SMALLER. IN VIDEOS 2 AND 4 THE TRACKING STD IS HIGH COMPARED TO THE OTHER VIDEOS DUE TO THE LOW RESOLUTION OF THE ULTRASOUND OF APPROXIMATELY $50 \mu m$. MEASUREMENTS WITH THE MICROPARTICLES WERE MADE IN WATER AND MEASUREMENTS WITH THE MICROJETS AND JANUS PARTICLES WERE MADE IN HYDROGEN PEROXIDE (JANUS PARTICLES 30% AND MICROJETS 5%). ULTRASOUND MEASUREMENTS WERE MADE WITH THE SIEMENS ACUSON S2000 SYSTEM USING THE 14L6 PROBE AT 14 MHZ.

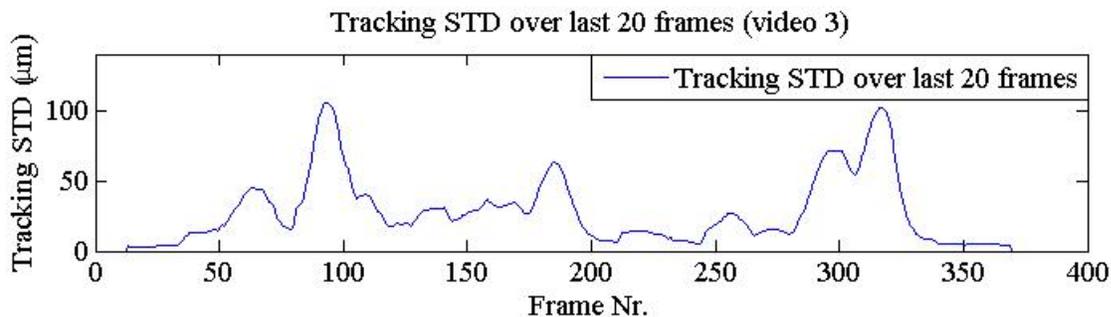


Fig. 13. The standard deviation of the position estimation over a period of 20 frames during video 3. Initially, the tracker does not localize the head of the microjet yet because it first has to determine the movement direction of the jet. Therefore, the tracker deviation initially increases as the tracker first converges to the centroid. A decrease of the STD is visible when the tracker localizes the head and converges to its position. However, the tracking STD shows another peak when the jet has made a turn, causing the tracker to track the wrong side of the jet with its trail until the correct speed of the jet is determined again. Later on, another peak is visible when another object comes close to the jet. The reason for this tracking deviation is further explained in "Tracker Robustness".

TABLE III

TRACKING STD AND AVERAGE VELOCITY DURING 3D TRACKING IN VIDEOS 7 AND 8. BECAUSE BOTH MEASUREMENTS WERE MADE AT THE SAME TIME, ULTRASOUND TRACKING IN 3D CAN BE COMPARED DIRECTLY TO MICROSCOPIC TRACKING IN 3D. BECAUSE IN ULTRASOUND TRACKING THE STD IN SCANNING DIRECTION (z) IS SIGNIFICANTLY DEVIATING FROM THE STD IN THE OTHER DIRECTIONS (x AND y) IT IS LISTED SEPERATELY. THIS IS DUE TO THE USAGE OF A DIFFERENT TRACKING METHOD AND ALGORITHM IN THAT DIRECTION.

| Video | Name | Average velocity | Tracking STD (x,y) | Tracking STD (z) | Tracking STD (x,y,z) |
|-------|-----------------------------|-----------------------|------------------------|----------------------|--------------------------|
| 7 | Microparticle Microscope 3D | 173.3 $\mu\text{m/s}$ | 19.7 μm | 9.0 μm | 21.6 μm |
| 8 | Microparticle Ultrasound 3D | 173.3 $\mu\text{m/s}$ | 20.8 μm | 178.3 μm | 176.3 μm |

At some points in the trajectory there is a tracking offset. This happens because the tracker does not correctly track the tip of the microjets due to variations in speed. In the video it can be seen that there was flow leading to these variations. Because the real position of the microrobots is determined manually offline by clicking on the position, also the determination of this position might have inaccuracies.

However, no other objective realistic scenario and more accurate method of position estimation was available on this setup. In figure 12 the trajectory of the microjet in video 4, which is measured with ultrasound, is shown. In this measurement, the manually estimated position is clearly more accurate than the tracked position as could be expected based on the inaccuracy of ultrasound.

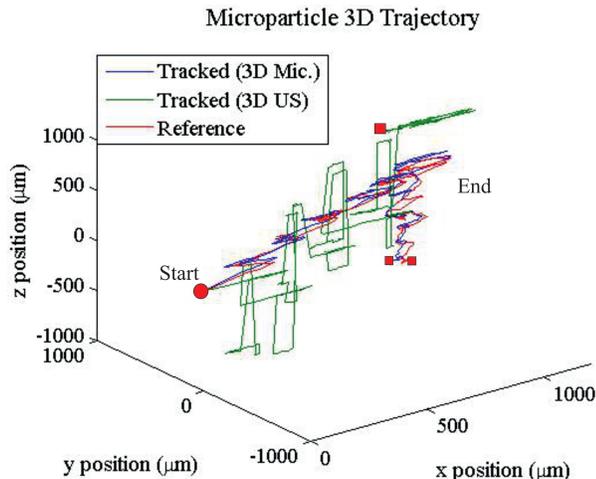


Fig. 14. The manually measured trajectory of the particle is compared to the trajectory tracked by 3D tracking with microscope and the trajectory tracked by 3D tracking with ultrasound. It can be observed that the ultrasound plane scanning algorithm loses track when the particle makes a quick movement downwards.

Figure 13 shows the STD of the tracking measured over a period of 20 frames in video 3. Initially, the tracker does not localize the head of the microjet yet because it first has to determine the movement direction of the jet. Therefore, the tracker initially has a high STD. A steep slope is visible when the tracker localizes the head and converges to its position. After this convergence the tracking STD has become much smaller. However, the STD shows another peak when the jet has made a turn, causing the tracker to track the wrong side of the jet with its trail until the correct speed of the jet is determined again. Later on, another peak is visible when another object comes close to the jet. The reason for this tracking deviation is further explained in the next section.

B. Tracking Robustness

To test the tracker's robustness to encounters with other objects or background, the tracker is implemented to track objects in proximity of other objects. Noticeable encounters in the videos of these measurements have been studied to identify the tracker's weak spots and are shown in video 6. Inside these videos the tracker proves robust to encounters with other objects at a distance further than the size of the

connectivity region. Tracking microjets it can be observed that the tracker loses track of the correct jet in case of an occlusion with another jet or an encounter closer than 5 pixels (fig. 15D). This is caused by the algorithm failing to distinguish the jets as separate objects. Despite this drawback, the connectivity region of 5 pixels still holds as the optimal connectivity region size for microjets, as a smaller connectivity region results in segmentation within jets, causing inaccurate tracking or loss of the microjet itself. As the connectivity region for microparticles is smaller, because most microparticles only consist of a single blob on the binary image, tracking of microparticles is more robust. However, in situations where particle blobs might be connected to other blobs in the thresholding step due to proximity the tracker still loses track in the current implementation of the algorithm, as is the case in figure 15D.

C. 3D Tracking

To evaluate performance of 3D tracking with microscopes, the algorithm is implemented to track individual microparticles on microscopes in 3D. To assess tracker performance in this mode the position estimation of the tracker is compared with the manually measured position on the camera frames to analyze tracker accuracy. To evaluate performance of 3D tracking with ultrasound, the ultrasound scanning and 2D tracking algorithms are implemented to track the same microparticle at the same time as the perpendicular camera tracking. To assess tracker performance in this mode 2D tracking STD and scanning error were measured separately by comparison of the ultrasound-based tracker position estimation with the position manually measured on the camera frames. In this way a comparison between 3D tracking with ultrasound and 3D tracking with microscopes can be made. Results of these measurements are displayed in table 3.

In figure 14 the manually measured trajectory of the particle is compared to the trajectory tracked by 3D tracking with microscope and the trajectory tracked by 3D tracking with ultrasound.

The 3D tracking on ultrasound clearly has a high tracking STD in the z-direction. Also, making this measurement, it was hard to let the 3D ultrasound scanning algorithm catch on to the microparticle. In figure 14 and in video 7 and 8 it also is visible that the scanning loses track of the correct height of the object as soon as it makes a quick movement downwards. The problems of the scanning algorithm to catch on to an object and to robustly track this object are due to both the scanning velocity as well as the way in which the object intensity is determined. Because of the low scanning velocity, which is necessary to prevent unwanted movement of the basin and the particle, the tracker only determines the position approximately every second. Due to this delay, the tracker loses track of objects translating more than the scanning amplitude (which was 0.7 mm at the measurement) within one second. Because the object intensity is determined based on the amount of pixels in the binary image, and this image is sensitive to fluctuations in background intensity and

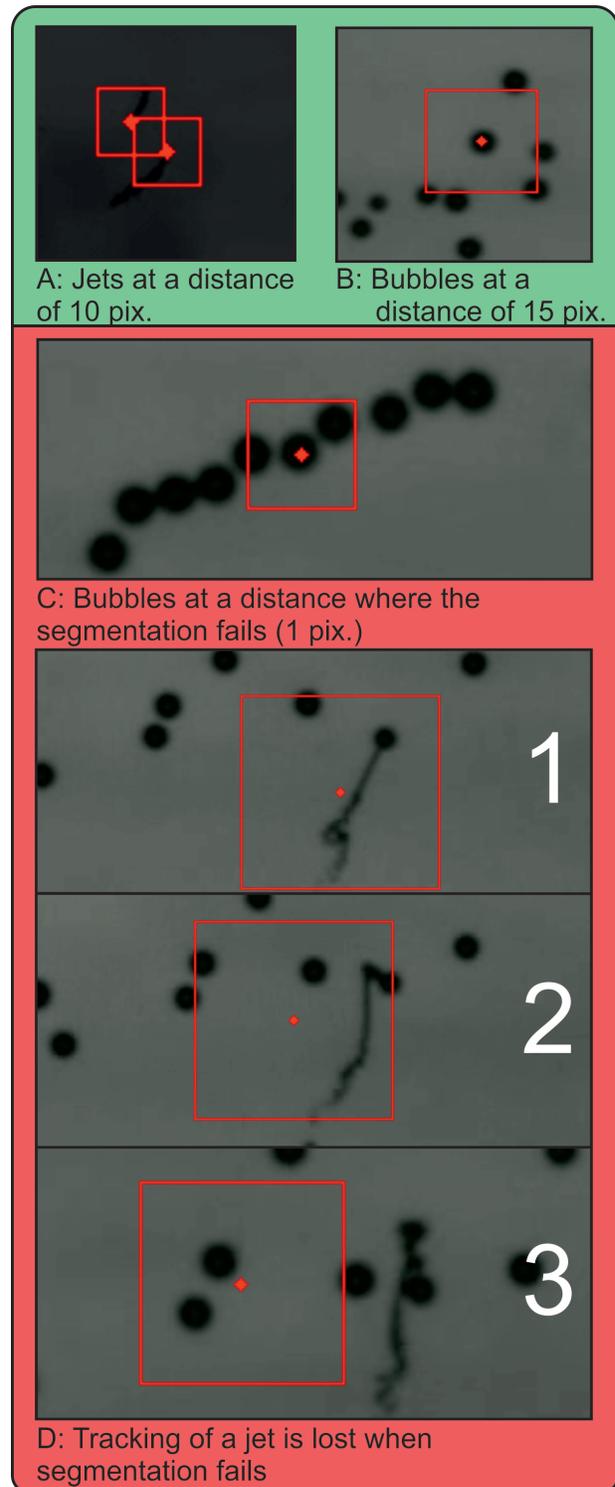


Fig. 15. Examples of encounters with other objects in which the tracker does not lose the tracked object (A and B) and events in the videos that cause the tracker to lose track (C and D). A: Two jets heading in opposite direction and passing close to each other. B: A group of bubbles in which the tracker is able to keep track of a single bubble. C: A group of bubbles where the tracker loses track of the correct bubble and moves to other bubbles. This happens because the bubbles are connected by the adaptive thresholding. D: A sequence of frames showing how the tracker loses track of a microjet passing too close to bubbles.

contrast, this leads to frequent errors in the determination of the height of the object.

V. DISCUSSION

A. Conclusion

In this work robust microscopic tracking of multiple types of microrobots in 2D is demonstrated. The tracker was able to accurately track microrobots with inconsistent shapes (e.g. microjets). Tracking showed robustness during close encounters with other objects. Although the tracker proves robust in these situations, it is observed that the tracker loses the object in situations of occlusion or encounters within the distance of the connectivity region. Based on these observations the 2D tracker is considered robust and accurate for microrobot tracking inside multi-object frames as long as no occlusions occur. Therefore the tracker is suitable for several biomedical research applications (e.g. micromanipulation of cells, IVF).

In 3D, the tracker was able to accurately track a microparticle in a microscope and therefore the tracker can be used for tracking of microparticles in biomedical research applications in 3D. Additionally, 3D tracking of the microparticle in ultrasound images was demonstrated. Although it proved feasible to track a microparticle in ultrasound images in 3D, the scanning algorithm necessary to do this 3D tracking was not robust and was only suitable for tracking of the particle along short distances. Therefore, future research is needed to enable biomedical application of ultrasound tracking in 3D.

B. Future Research

Since the tracker loses track when the object approaches another object at a distance closer than the connectivity region, future implementation of the tracking algorithm could have an enhanced performance by providing a more size and shape consistency based position estimation of the particle filter (e.g. by using the Hough transform [25]). Another issue to be solved with these occlusions is assignment of identities to the objects without errors due to the occlusion. Development of a decision framework for these events might prevent this type of mix-ups. However, a major challenge in including these measures is to leave insensitivity to size and shape variations of the microrobot itself.

Additionally, it was observed that the tracker is not robust for tracking objects in ultrasound images in 3D, due to inaccuracy in the determination of object intensity in the ultrasound frame and slowness of the scanning. Therefore the performance of 3D tracking in ultrasound images could be enhanced by development of an accurate method to assess the relative object intensity in an ultrasound image that effectively neglects artefacts and other objects in its estimation. To decrease the tracking delay that is caused by the scanning algorithm, the duration of one scanning cycle should be decreased. This can be done by either providing a better fixation of the basin inside the setup and increasing the scanning velocity, or effectively restricting the scanning amplitude based on statistical analysis of the object speed (e.g. with a particle filter).

Despite these tracking errors, the tracker provides robust position estimation of one or multiple objects of multiple types of magnetic agents in other experimental situations. Therefore future applications of the tracker include multi-object and multi-agent control. This might enable microrobot collaboration leading to execution of complex tasks by groups of microrobots, on which innovative work already has been performed by Pawashe et al. [26].

In this work, sub-resolution position estimation and tracking of microjets, microparticles and Janus particles on ultrasound is demonstrated. Although the microparticles are visible on the ultrasound measurements, it is observed that microjets and janus particles are not. For now, tracking of microjets and janus particles has solely been done based on the ultrasound signal that was reflected by the bubbles emitted from jets and janus particles. As the radius of the bubbles of jets can reach up to half of the length of the jets (approximately 20 microns) inside the basin in which the measurements are made and the water-air boundary has a high reflection coefficient (99%), the bubbles of the jets are clearly visible on ultrasound measurements.

Multiple reasons can be hypothesized why the jets and the Janus particles themselves are not visible on ultrasound. One reason for the invisibility of small microrobots on ultrasound could be their size. As the radius of janus particles and the width of microjets are far below the wavelength on the maximum frequency of the used imaging system (Siemens Acuson S2000, with a maximum frequency of 18MHz with the 18L6 probe), most of the reflection might scatter, causing no measurable reflection. Another reason of the invisibility of the jets and janus particles might be their materials. As the outer material of jets is titanium and the reflection coefficient of the water-titanium interface is only 8%, their reflection intensity will be less than one tenth of the intensity of a bubble with the same shape and size. The outer material of janus particles, platinum, results in a better reflection coefficient of the interface of approximately 50%. However, the janus particles are even smaller than microjets, which also contributes to the loss of signal.

Therefore, to enhance the future prospects of usage of ultrasound-controlled microrobots (i.e. in medical treatments), a major challenge for future research is to provide means for imaging of these microrobots, as this might enable application of these robots *in vivo*. High-frequency ultrasound might provide means to do this because of the short wavelength. However, attenuation of the ultrasound is severely increased because of the increase in frequency, which might prevent this technique from being used in a medical environment. Magneto-motive ultrasound [27] might provide another method to enhance visibility of microrobots on ultrasound, by enhancing the ultrasound signal with oscillations of the microrobots.

For now, the algorithms and methods presented in this work enable effective position measurement for real-time control of most microrobots. This accurate position estimation is suitable for use in 2D as well as 3D microrobot applications such as manipulation of cells and local drug delivery. In

line with this development, these applications can be further examined.

Besides development of the tracking algorithm, other efforts have been made in this work to enable both microscopic and ultrasound tracking of microrobots. To provide for optical and ultrasound measurements of the microrobots the MARS 2 setup has been used, which is the new version of the MARS 1 setup and specifically adapted for 3D tracking of microrobots on ultrasound. Additionally the tracking algorithm has been implemented in an object-oriented library and a user-friendly interface has been designed for the user to interact with the program and implement the tracker. All of this will be outlined in the next section.

VI. APPENDIX

A. MARS 2 Setup

The original setup, which was named MARS 1, had to be adapted to enable ultrasound tracking of microrobots in 3D and to improve the setup. These improvements are highlighted in this appendix. For the completeness, also the most important features of the original setup are explained.

1) *Probe:* The setup is configured for usage of the Siemens 14L5 Transducer. This probe is chosen for its compact design and high frequency range (5-14 Hz). The probe is moved over the sample by the probe holder system based on the plane scanning algorithm. For more information on the probe holder system, please refer to the section "Probe Holder System" in this appendix. For more information on the ultrasound plane scanning algorithm, please refer to the section "Ultrasound Plane Scanning" in the report.

2) *Basin:* The basin in this design is designed to facilitate ultrasound imaging of the microrobots. This is achieved by providing more movement space for the microrobots to enable clearly visible microrobot movement within the basin. To enable accurate microscopic measurement as well as good transmission of ultrasound into the sample the basin is made of multiple materials. The sides and bottom of the basin are made of acrylic for a clear microscopic view inside the basin. The front is made of PVA that, due to its acoustic impedance which is similar to that of water, enables efficient transmission of the ultrasound waves.

3) *Microscope:* An optical system with an adjustable zoom with a minimum of 2.4X and a maximum of 24.0X is mounted on a linear stage to enable precise focusing of the system. For recording a CCD sensor is used with a pixel width and pixel height of $5.50 \mu m$, providing a theoretical maximum resolution of approximately $0.50 \mu m$.

4) *Coils:* The coil configuration of the MARS 1 setup is maintained in this design to ensure consistency in field homogeneity and field strength. The coils each have 1680 windings of copper wire up to a coil diameter of 40 mm with a core diameter of 10 mm. The coil currents are hardware-limited to a maximum current of 2 Ampere. Together, the

coils can generate a homogeneous magnetic field of up to 65 mT at the center of the system where the basin is located.

5) *Sample Insertion System:* The sample insertion system serves to fix the sample, a basin filled with microrobots in a liquid, in the correct position. The sample insertion system is adapted to fit the new shape and new requirements of the design. The tubes of the system exactly fit the helmets and the coils, for a robust attachment to the system. Compared to the old setup the inside size of the tubes is increased to provide space for a larger basin. For insertion and removal of the basin, a removable basin holder is designed which can be clamped inside the insertion tube of the system. The basin is clamped into this holder, which provides robustness of the basin to the movements of the ultrasound probe.

6) *Helmets:* The helmets serve to fix all the components of the system at the right distances from each other. Compared to the old setup, the helmets were adapted to provide entrance of the probe to the basin. Because it is most convenient for the user to attach the probe at the front side, this required translocation of the entrance for the basinholder

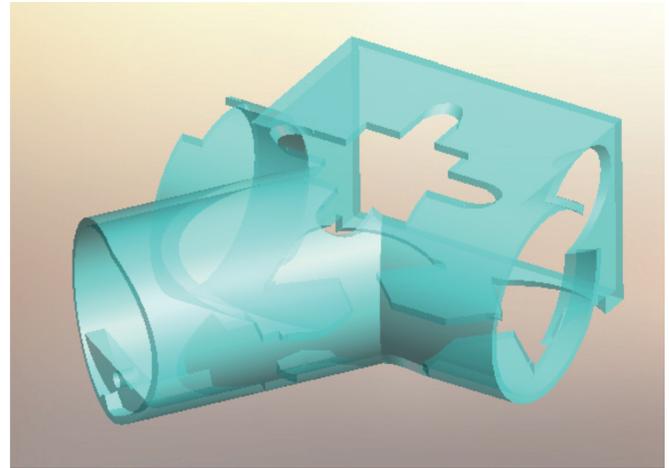


Fig. 18. The tubes of the sample insertion system. Surfaces of the coils and cores have been cut out to enable an optimal coil configuration. The basin holder exactly fits inside this system.

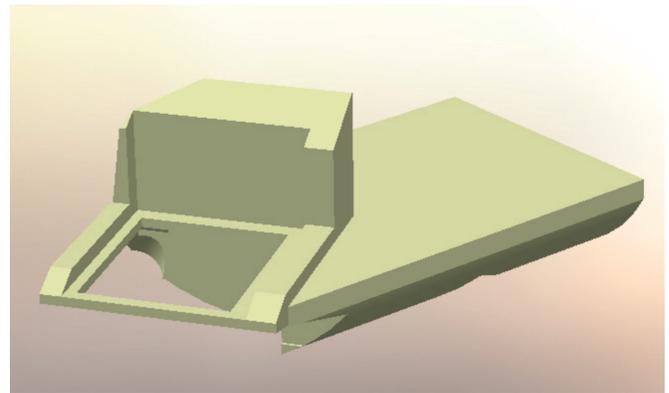


Fig. 19. Removable Basin Holder. The basin is fixed inside the holder by clamping edges.

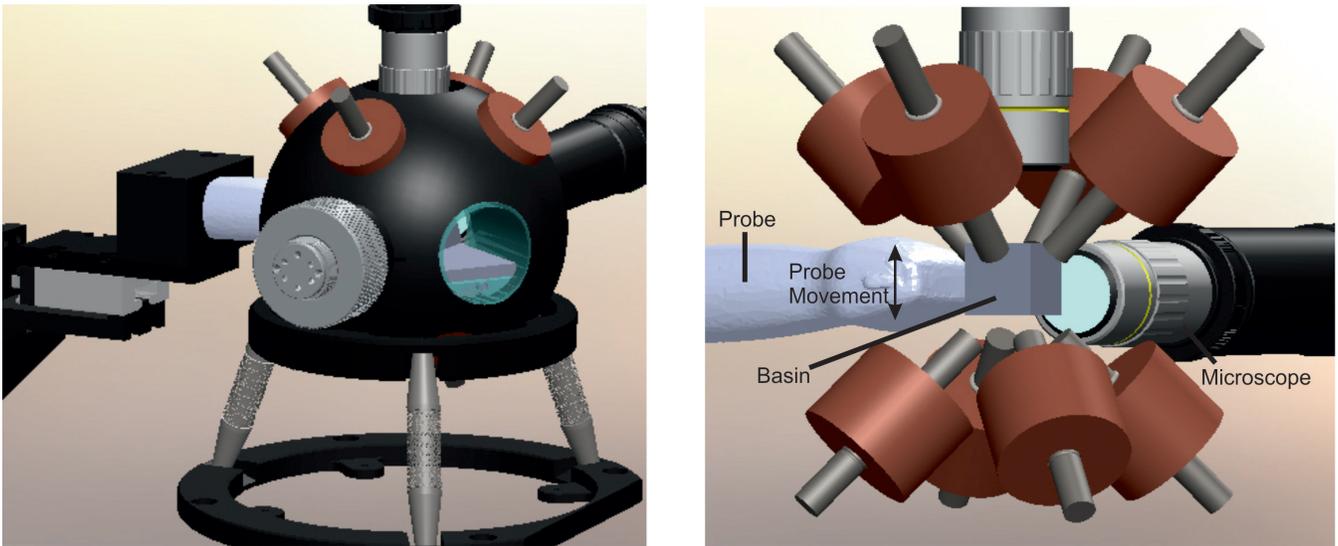


Fig. 16. Left: General overview of the MARS 2 setup. Right: Configuration of the coils, the microscopes and the ultrasound probe around the microrobot basin.

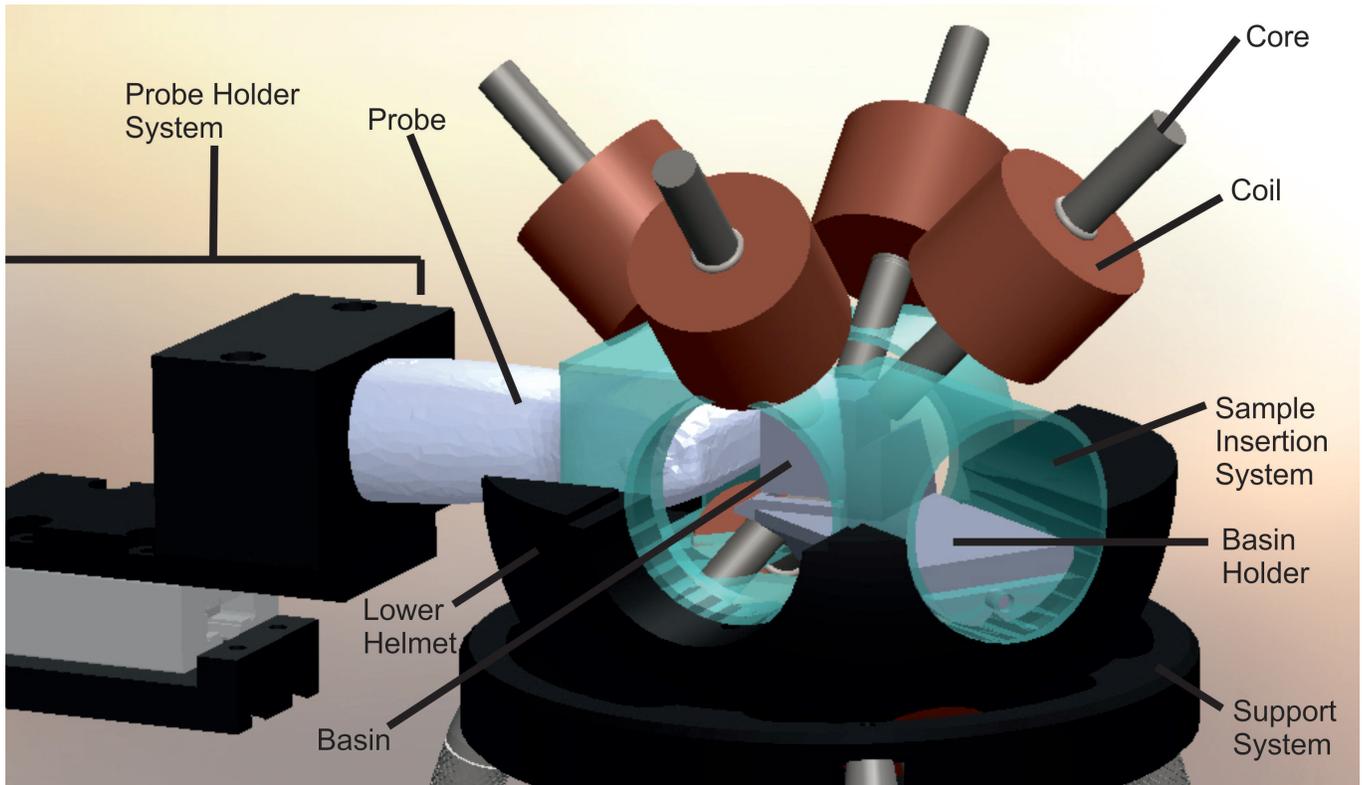


Fig. 17. Inside view of the MARS 2 setup.

to the other side. However, it has been placed slightly to the right to make it more accessible. To ensure that the basin and basinholder are properly placed into the setup and to check the contact of the probe with the basin, a looking hole has been made to the side of the probe entrance.

7) *Support System*: The support system is designed to carry the rest of the system and keep it perfectly aligned to

the microscopes and the ultrasound probe. In the old setup the upper ring was deformed causing the system to misalign, resulting in failure of the control of microrobots. To prevent this in the new setup, the new version of the upper ring is twice as thick as its old version.

8) *Probe Holder System*: The probe is scanned over the sample by a linear stage. The probe holder system, which

is attached to this linear stage, serves to exactly position the probe at the right initial height and to put the probe with a pressure of approximately 2N against the sample to obtain consistent ultrasound measurements. To enable this exact positioning, the probe holder system is adjustable in three degrees of freedom. The height can be adjusted by adjusting the height of the linear stage, which is mounted on a vertical optical rail. Also the height can be adjusted via the linear stage, although there should remain enough movement space within the range of the linear stage for scanning. To move the probe closer or further away from the linear stage rail 1 can be used and to move the probe towards the sample rail 2 can be used. Both of these rails are made to clamp and can be fixed at one position by bolts to prevent unwanted movement of the probe.

The probe itself is clamped by the probe holder which is 3D printed to exactly fit the surface of the probe and a rubber layer is added on the inside of the probe holder for robust attachment of the probe. To put a constant pressure on the basin the probe and probeholder are mounted onto a low friction carrier and rail via the carrier plate. This carrier only has a few millimeters movement space and is constantly pressed by a spring, which maintains the force of approximately 2N. When the probe is not in position at the sample, this carrier is stopped at the end of the rail by a beam in front. To make consistent ultrasound measurements, the probe should be exactly positioned at the basin to press the probe backward against the spring which will ensure a constant pressure on the probe during scanning. This exact positioning is done by precise attachment of rail 2.

B. Setup Recommendations

As is described in last section, many improvements have been made to the setup in this work. However, new possibilities for improvements came up after first implementation of the new setup.

One important drawback of the setup is the magnetic field strength. During experiments with microparticles in 3D it was not possible to lift big clusters of particles (ζ 10 particles). Additionally, it was not possible to control Janus



Fig. 20. Photograph of the new setup, which was realized by 3D printing parts from CAD drawings, adding a layer of paint and attaching parts together with bolts.

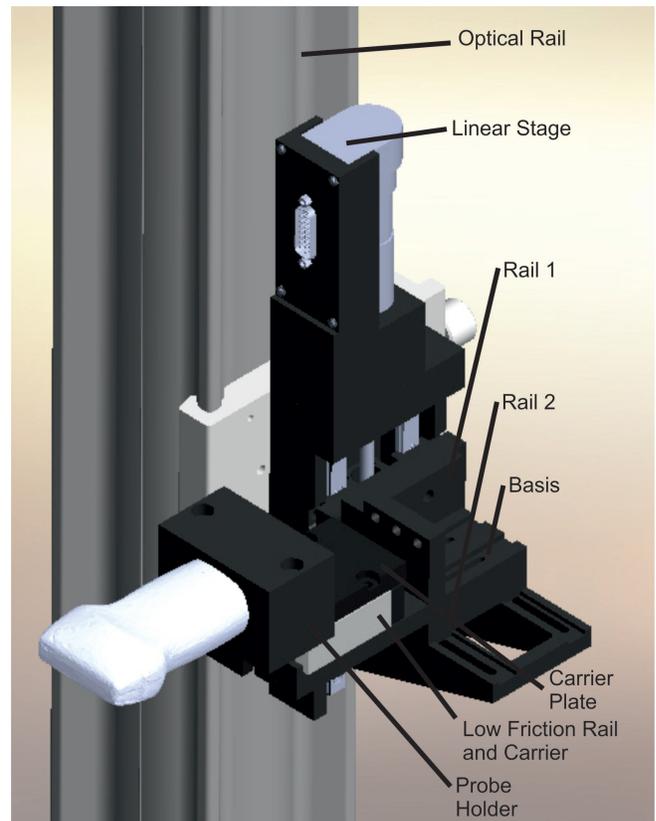


Fig. 21. The probe holder system. The optical rail enables adjustment of the height. Rail 1 and rail 2 enable adjustment in two other degrees of freedom for precise positioning of the initial position of the probe. The three holes on the basis serve to fix springs that can be inserted between the basis and the carrier plate.

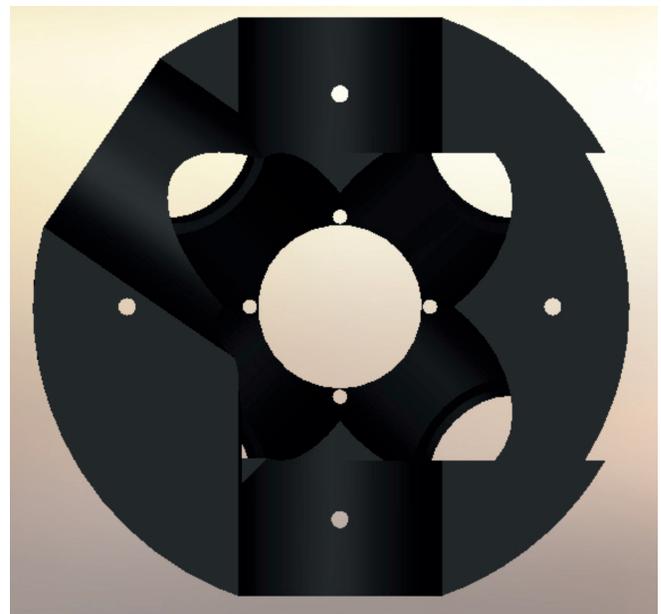


Fig. 22. Top view of the lower helmet. The hole in the bottom and the hole on top are for the illumination modules, the square hole in the front is for the probe, the hole to the left is for insertion of the sample and the remaining hole is for the horizontal microscope.

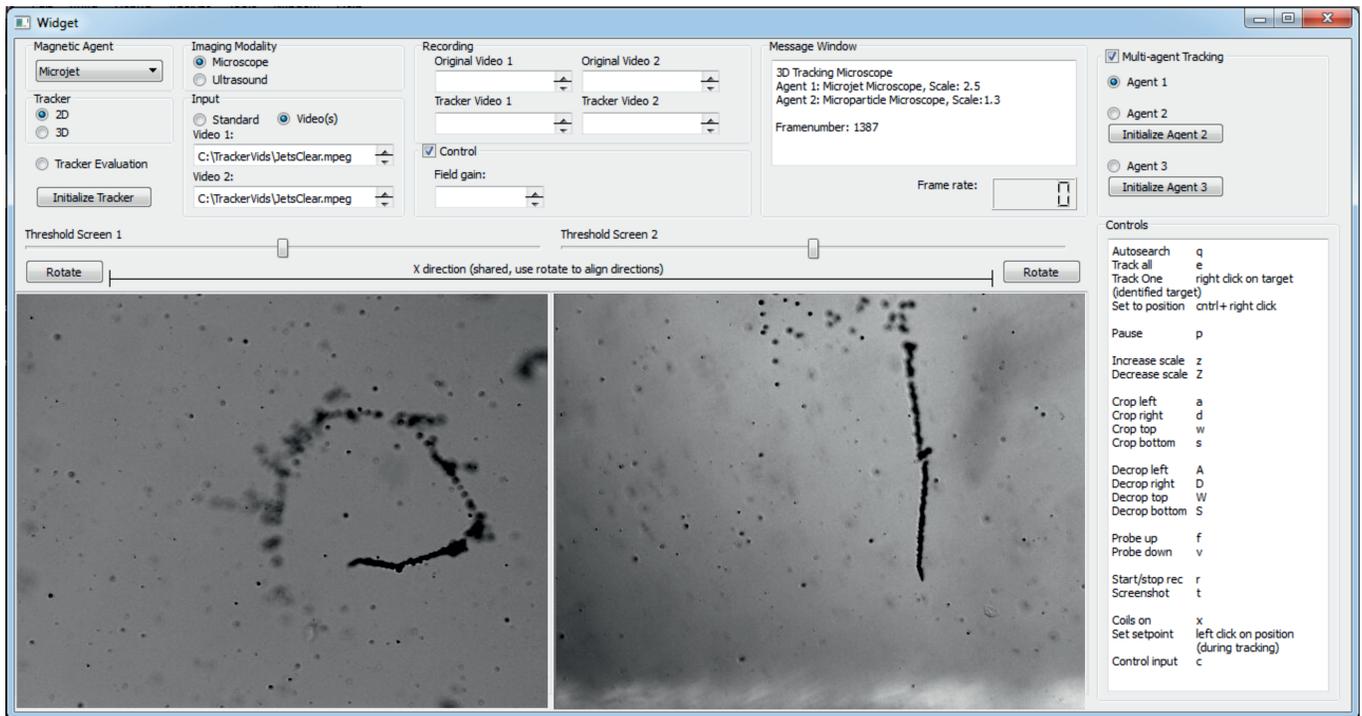


Fig. 23. The graphical user interface. The left screen shows input from either the ultrasound or the first camera and the right screen shows input from the second camera if microscopic 3D tracking is implemented. In the lower right corner the controls are displayed. Video can be recorded or read-out on paths specified by the user.

particle in an environment with flow. Although these particles aligned to the magnetic field, they did not translate in setpoint direction at maximum magnetic field strength. Therefore, the setup could be enhanced by using larger coils. However, larger coils also require redesign of the rest of the setup, including an increased radius of the helmets, an increased radius of the support system and adaptation of the sample insertion system.

Besides the coil size, it is also recommended to adapt other coil parameters in a new design. It was observed that after using the coils for control the cores remain magnetized. Some of this magnetization seems due to self-induction of the coils. This problem could be solved by increasing the diameter of the copper wire. There is also some permanent magnetization remaining, which is due to magnetization of the core material. Therefore it is recommended to adapt the coil material to prevent this magnetization.

Making the measurements, a major problem was getting microrobots inside the field of view of the cameras. This was due to the small field of view of the cameras as well as the impossibility to translate the basin or the cameras. This severely increased the effort and time necessary for making measurements of microrobots and therefore translation of the microscopes or the basin is considered as an essential improvement for next version of the setup. Additionally, it will be easier to localize a microrobot when the basin is smaller. Therefore, it is recommended that the basin size is adapted to approximately the size of the field of view of the cameras. In that case, a microrobot will always be inside the field of view of the camera. However, it should be taken

into account that the field of view of the cameras in this case should be big enough to include multiple microrobots with enough movement space to enable movement above the resolution of the ultrasound.

C. Tracker Interface

The communication between the tracker program and the user is provided by its graphical user interface. This user interface has been designed for efficient and intuitive usage, to enhance the usability of the tracker for inexperienced users. To enable this efficient and intuitive usage, an important goal in the design was to keep the basic usage of the tracker program as simple as possible.

To keep it simple for the user to start using the tracker, parameters to track the microrobots are pre-set for multiple types of microrobots, including:

- Microjets
- Microparticles
- Janus particles

For these microrobots, separate parameters have been set for microscopic tracking and for tracking on ultrasound, which are automatically selected by selecting the imaging modality. For the microjets, these ultrasound settings are specific due to their bubble trails. For the other types of microrobots this is not the case.

To give the user the opportunity of adjusting the sensitivity of the tracker, the threshold, which is of main influence to the sensitivity, is made adjustable in the interface. Therefore only an initial threshold is fixed in the settings, which in general

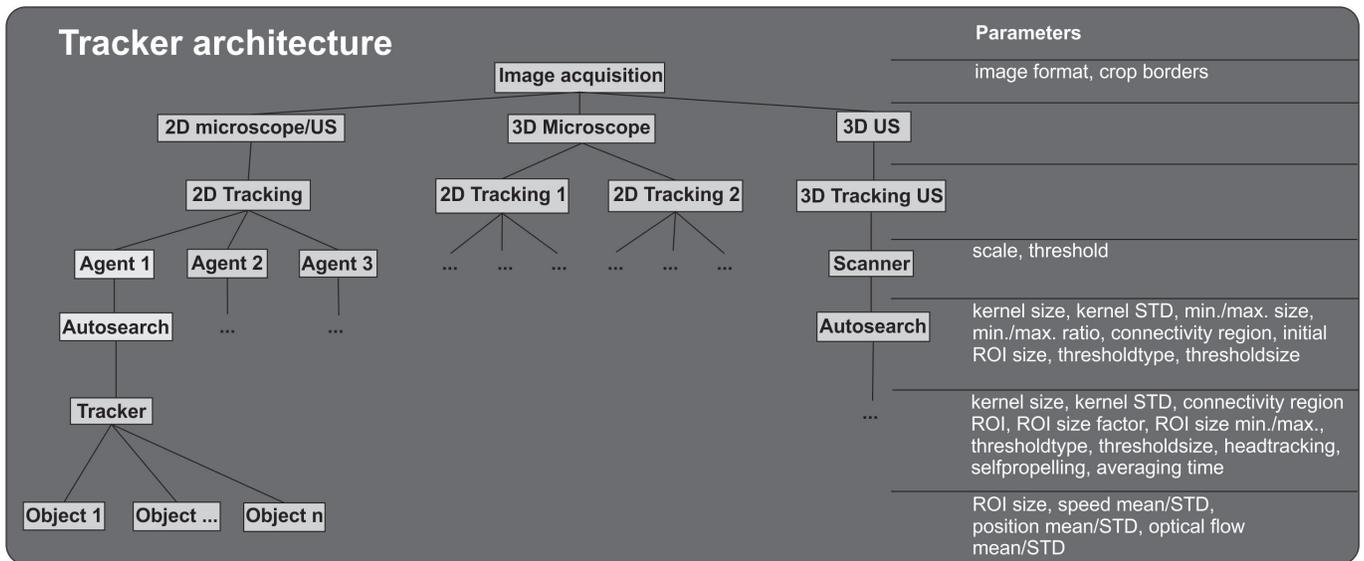


Fig. 24. Architecture of the tracker program, showing the order in which parameters are set and calculated. The autosearch and tracker are implemented object-oriented to enable transition between multiple types of agents during tracking.

is close to the optimal threshold for tracking. Adjustment of the threshold is interfaced by a slider bar, which enables efficient adjustment of the threshold during tracking.

Because the kernel as well as the adaptive threshold are sensitive to object size, either the kernel and threshold size or the scale of the image should be adjustable. Because the kernel has a limit on its STD compared to its size due to loss of function energy, which leads to black-out of the image, the image scale is used in the interface to adjust the relative size of objects. This is done by either increasing or decreasing the amount of pixels in the image by interpolation.

Besides the scale, also the region of the image can be adjusted. The top, bottom, left and right side can be cropped individually to specifically select a region in the image in which the objects of interest appear. Because the size of the image cannot be varied by an unrestricted amount of pixels per frame during tracking, the both the cropping and scale

adjustment are realized by pressing keys, which will change the image change size by a fixed amount of pixels for each iteration.

Both the adjustable and the pre-set parameters are set for one instance of the tracker. In the tracker program, it is possible to open multiple instances of the tracker using specific tracker settings. In this way, multiple types of microrobots can be searched and tracked with specific tracker settings for each type at the same time, while the algorithm distinguishes between the types of microrobots and tracks the position of each individual microrobot of each selected type.

The tracker program can be initialized in three different modes of tracking by choosing the corresponding combination of buttons:

- Microscopic or ultrasound tracking in 2D
- Microscopic tracking in 3D
- Ultrasound tracking in 3D

The initial setting of the tracker program is the 2D tracking mode. This mode enables robust tracking of magnetic agents in either microscope or ultrasound input in 2D and uses only one screen of the interface since there is only one input.

For microscopic tracking in 3D both screens are used: one for each microscope. To select an object for tracking only on one of the screens, no matter which one, an object has to be selected. The algorithm automatically searches for the corresponding object on the other screen. Both this mode as well as the 2D tracking mode are implementable for multiple agents.

The mode for ultrasound tracking in 3D is more limited than the other two. Due to the scanning limitations of the setup only one object can be tracked. Therefore, in this mode, both multi-object and multi-agent tracking are not possible.



Fig. 25. Multi-agent tracking applied to bubbles using the microparticle tracker settings and a microjet using the microjet tracker settings.

Tracker Initialization

1. Select magnetic agent from the dropdown menu.
2. Select the type of tracker ("2D" or "3D") and the imaging modality ("Microscope" or "Ultrasound").
3. In case of video input, select the "Video(s)" button in the input menu and enter the paths to the video file/files in the "Video" text box/boxes.
4. To record a video under a specific filename when pressing 'r', enter the video file names in the text boxes in the recording menu.
5. When ready, press the "Initialize Tracker" button.

Tracker Evaluation

1. Do steps 1 to 3 of the tracker initialization. The evaluation only works for videos, due to the need of user input. Videos of 3D microscopic tracking should be analyzed separately in 2D mode.
2. Select the "Tracker Evaluation" button.
3. Initialize the tracker program by clicking the "Initialize Tracker" button.
4. Scale and adapt the threshold if necessary, as is described in steps 2 and 3 of "Search, Selection and Tracking".
5. Select microrobots to track, as described in step 5 of "Search, Selection and Tracking". For the evaluation it is only possible to track a single microrobot of a single agent.
6. For every frame of the evaluation video: - Press 'p' if the tracker has correctly tracked the point the right point on the object (head or centroid).
- If the tracker did not find the point correctly, click with the right mouse button on the correct position.
7. Exit the program. The data folder will contain a list with tracker positions and correct positions.

Like the 2D tracking mode, only one input is available and displayed. When implementing this mode, the user can first choose the initial ultrasound plane by scanning over the sample. After activation the scanning algorithm takes over and automatically updates the frames of the 2D tracker. In this mode the framerate is much lower than in the other modes due to the implementation of the scanning algorithm, which first has to search for the correct frame before 2D analysis.

For the control of magnetic agents, a position-based PID controller has been integrated inside the tracker to control a tracked microrobot. To use this function of the program, the electromagnetic system should be connected. To turn the system on or off the 'x' key can be pressed. After turning on the system, the magnetic agent is automatically servoed to a position that can be indicated by left-clicking on the tracker output screen. The system starts controlling the agent as soon as this position is indicated. The user can choose if the controlinput is used for tracking or not by using the 'c' key to turn this type of tracking on or off.

Search, Selection and Tracking

1. If there are unnecessary regions on the borders of the image (i.e. edges of a basin) crop the image borders using 'w','a','d' and 's'.
2. Set the right scale for the image. This can be done by looking at the shape drawn in the left corner of the tracker output. The magnetic agents should have approximately the same size and shape of that shape.
3. If the autosearch does not seem to find the magnetic agents correctly, adjust the threshold until it does.
4. For ultrasound tracking in 3D, scan the probe over the sample, using 'f' to go up and 'v' to go down, to search for objects.
5. The tracker program automatically starts searching for the selected type of magnetic agent. The following can be done to select objects to track:
 - Press "e" to track all objects found by the autosearch.
 - Click with the right mousebutton on a found object to track the object.
 - Click with the left mousebutton on a position in the image to start tracking at that position.
6. The program will now start tracking the selected agents.

Multi-agent Tracking

After doing the steps in "Search, Selection and Tracking":

1. Select the "Multi-agent Tracking" menu.
2. To select a second agent, select "Agent 2" and the type of magnetic agent to be tracked.
3. To initialize the second agent, click on the "Initialize Agent 2" button.
4. Scale and adapt the threshold if necessary, as is described in steps 2 and 3 of "Search, Selection and Tracking".
5. Select microrobots to track, as described in step 5 of "Search, Selection and Tracking".
6. The program will now start tracking a second agent.
7. To track a third agent, do steps 1 to 6 again for this agent.

**Settings for each agent can be adapted by changing the setting while the corresponding agent is selected.*

Also, the program has been designed to be easily extendable by other control algorithms. In the program structure, information of individual microrobots is registered within the *Microrobot* class. This enables development of more elaborate control algorithms within the program structure that can take into account the registered properties of the microrobot: size, shape, position and speed.

Manuals for several features of the program are shown in the text boxes.

REFERENCES

- [1] Solovev, A.A., Sanchez, S., Pumera, M., Mei, Y.F., Schmidt, O.G., Magnetic control of tubular catalytic microbots for the transport, assembly, and delivery of micro-objects. *Advanced Functional Materials*, 2010. 20(15): p. 2430-2435.
- [2] Sanchez, S., Solovev, A.A., Schulze, S., Schmidt, O.G., Controlled manipulation of multiple cells using catalytic microbots. *Chemical Communications*, 2011. 47(2): p. 698-700.
- [3] Kummer, M.P., Abbot, J.J., Kratochvil, B.E., Borer, R., Sengul, A., Nelson, B.J., Octomag: An electromagnetic system for 5-DOF wireless micromanipulation. *IEEE Transactions on Robotics*, 2010. 26(6): p. 1006-1017.
- [4] Nelson, B.J., I.K. Kaliakatsos, and J.J. Abbott, Microrobots for minimally invasive medicine. 2010. p. 55-85.
- [5] Sitti, M., Miniature devices: Voyage of the microrobots. *Nature*, 2009. 458(7242): p. 1121-1122.
- [6] Sanchez, S., Ananth, A.N., Fomin, V.M., Viehrig, M., Schmidt, O.G., Superfast motion of catalytic microjet engines at physiological temperature. *Journal of the American Chemical Society*, 2011. 133(38): p. 14860-14863.
- [7] Khalil, I.S.M., Dijkslag, H.C., Abelman, L., Misra, S., "MagnetoSperm: A microrobot that navigates using weak magnetic fields", *Applied Physics Letters*, 2014. In Press.
- [8] Khalil, I.S.M., Magdanz, V., Sanchez, S., Schmidt, O.G., Misra, S., "Biocompatible, accurate, and fully autonomous: A sperm-driven micro-bio-robot", *Journal of Micro-Bio Robotics*, 9(3-4): 79-86, August 2014.
- [9] Khalil, I.S.M., Metz, R.M.P., Reefman, B.A., Misra, S., Magnetic-based minimum input motion control of paramagnetic microparticles in three-dimensional space. in *Intelligent Robots and Systems (IROS)*, 2013 IEEE/RSJ International Conference on. 2013.
- [10] Khalil, I.S.M., Keuning, J.D., Abelman, L., Misra, S., Wireless magnetic-based control of paramagnetic microparticles. *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. 2012.
- [11] Diller, E., et al. Three dimensional independent control of multiple magnetic microrobots. *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [12] Martel, S., Felfoul, O., Mathieu, J., Chanu, A., Tamaz, M., Mohammadi, M., Mankiewicz, M., Tabatabaei, M., MRI-based medical nanorobotic platform for the control of magnetic nanoparticles and flagellated bacteria for target interventions in human capillaries. *International Journal of Robotics Research*, 2009. 28(9): p. 1169-1182.
- [13] Khalil, I.S.M., Ferreira, P., Eleuterio, R., Korte, C. L., Misra, S., Magnetic-Based Closed-Loop Control of Paramagnetic Microparticles using Ultrasound Feedback. *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. 2013.
- [14] Sanchez Secades, L.A., Magdanz, V., Schmidt, O. G., Misra, S., Magnetic Control of Self-Propelled Microjets under Ultrasound Image Guidance, in *BioRob 2014*.
- [15] Yesin, K.B., K. Vollmers, and B.J. Nelson. Guidance of magnetic intraocular microrobots by active defocused tracking. in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2004.
- [16] Felfoul, O., Aboussouan, E., Chanu, A., Martel, S., Real-time positioning and tracking technique for endovascular untethered microrobots propelled by MRI gradients. in *Proceedings - IEEE International Conference on Robotics and Automation*. 2009.
- [17] Buerkle, A., Schmoedel, M., Kiefer, M., Amavasai, B.P., Caparelli, F., Selvan, A., Travis, J.R., Vision-based closed-loop control of mobile microrobots for micro handling tasks. in *Proceedings of SPIE - The International Society for Optical Engineering*. 2001.
- [18] Jiang, R.M., Crookes, D., Luo, N., Davidson, M.W., Live-cell tracking using SIFT features in DIC microscopic videos. *IEEE Transactions on Biomedical Engineering*, 2010. 57(9): p. 2219-2228.
- [19] Chatterjee, R., Ghosha, M., Chowdhury, A.S., Rayb, N., Cell tracking in microscopic video using matching and linking of bipartite graphs. *Computer Methods and Programs in Biomedicine*, 2013. 112(3): p. 422-431.
- [20] Xiuzhuang, Z. and L. Yao. Efficient Mean Shift Particle Filter for Sperm Cells Tracking. in *Computational Intelligence and Security*, 2009. CIS '09. International Conference on. 2009.
- [21] Smal, I., Draegestein, K., Galjart, N., Niessen, W., Meijering, E., Particle Filtering for Multiple Object Tracking in Dynamic Fluorescence Microscopy Images: Application to Microtubule Growth Analysis. *Medical Imaging*, *IEEE Transactions on*, 2008. 27(6): p. 789-804.
- [22] Arulampalam, M.S., Maskell, S., Gordon, M., Clapp, T., A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 2002. 50(2): p. 174-188.
- [23] Baker, S. and I. Matthews, Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 2004. 56(3): p. 221-255.
- [24] Fornasini, P. (2008). *The Uncertainty in Physical Measurements: An Introduction to Data Analysis in the Physics Laboratory*, 1st ed., Springer, ISBN 978-0-387-78650-6.
- [25] Yip, R.K.K., P.K.S. Tam, and D.N.K. Leung, Modification of hough transform for object recognition using a 2-dimensional array. *Pattern Recognition*, 1995. 28(11): p. 1733-1744.
- [26] Pawashe, C., S. Floyd, and M. Sitti, Multiple magnetic microrobot control using electrostatic anchoring. *Applied Physics Letters*, 2009. 94(16).
- [27] Mehrmohammadi, M., Shin, T.H., Qu, M., Kruizinga, P., Truby, R.L., Lee, J.H., Cheon, J., Emelianov, S.Y., In vivo pulsed magneto-motive ultrasound imaging using high-performance magnetoactive contrast nanoagents. *Nanoscale*, 2013. 5(22): p. 11179-11186.