Martin Roznovjak

Feature Reprojection as Image Alignment Score in GNSS-Free UAV Localization

Metropolia University of Applied Sciences

Bachelor of Engineering

IoT and Cloud Computing

Bachelor's Thesis

22 November 2022

Abstract

Author:	Martin Roznovjak
Title:	Feature reprojection as image alignment score in GNSS-free UAV localization
Number of Pages:	50 pages + 80 appendices
Date:	22 November 2022
Degree:	Bachelor of Engineering
Degree Programme:	Information Technology
Professional Major:	IoT and Cloud Computing
Supervisors:	Jouko Kalmari, Software Designer Vesa Ollikainen, Senior Lecturer

---

Accurate absolute localization, commonly achieved with the help of Global Navigation Satellite Systems (GNSS), is critically important for unmanned aerial vehicle (UAV) navigation. However, GNSS can become unavailable or disturbed in certain environments. Therefore, different alternative solutions have been explored in the literature. A class of alternative approaches to absolute localization takes advantage of high-resolution satellite or aerial imagery and onboard video equipment.

The primary goal of this study was to assess the feasibility and characteristics of a novel solution to the visual self-localization of aerial video footage without the use of GNSS. The goal was achieved by developing, implementing, and evaluating a proof-of-concept solution and comparing it with the performance of a benchmark solution. The study was carried out for Huld Ltd. which has been developing a related product.

Both, the proposed and the benchmark solutions are Monte Carlo localization methods and use no other sensory information than video input and orthographic reference imagery (map). For absolute pose estimation, the proposed localization solution reprojects features from reference imagery to a query image. For comparison, the benchmark localization solution compares the query image to a corresponding part of the reference imagery using zero-mean normalized cross-correlation. The performance of the solutions was evaluated and compared on a diverse dataset of real aerial video footage.

The proposed localization solution achieved robust and consistent performance across the dataset and reached significantly lower localization error compared to the benchmark solution. The proposed method was found to be easily extendible and required no domain-specific engineering or tuning.

Keywords: feature reprojection, image alignment, season-invariant, absolute visual localization, aerial imagery, UAV, particle filter, computer vision

Contents

Appendices

Appendix 1: Grouped results

Appendix 2: Individual results

List of Abbreviations

BLS: Benchmark localization solution. An absolute visual localization method similar to the PLS used for comparison.

CRS: Coordinate Reference System. Framework for defining how coordinates map to physical locations on Earth.

FOV: Field of view. Visible area by a camera at a given moment.

GNSS: Global Navigation Satellite System. Global constellations of satellites that enable accurate localization on Earth, for example, Navstar (GPS), Galileo, BeiDou.

GT: Ground truth. The GNSS positioning data recorded simultaneously with the evaluation dataset footage.

IMU: Inertial measurement unit. A sensor that measures its acceleration, angular rate, and frequently also its orientation in a magnetic field.

LAF: Local Affine Frame. A parallelogram region of an image, typically used for describing the location, orientation, and scale of a feature.

NLS: National Land Survey of Finland. The official body responsible for cartographic and cadastral matters in Finland.

PF: Particle filter. A Monte Carlo method for stochastic state estimation.

PLS: Proposed localization solution. An absolute visual localization method proposed and evaluated in this work.

UAV: Unmanned aerial vehicle. An autonomous or remotely operated aircraft without any human onboard. Commonly referred to as a drone.

VO: Visual odometry. Egomotion estimation from a sequence of images.

ZNCC: Zero-mean normalized cross-correlation. Pearson correlation coefficient adapted for images.

1 Introduction

Unmanned Aerial Vehicles (UAVs) are now commonplace and rapidly expanding with a forecasted market size to reach over 38 billion USD by 2027 (MarketsandMarkets, 2022). They are or are predicted to be widely adopted across many civilian and military domains such as search and rescue, disaster relief, last-mile delivery, remote monitoring, surveying, and sensing, etc. (Belmonte, Morales and Fernández-Caballero, 2019)

Such a large market creates various submarkets and application-specific niches. A common requirement across the majority of the applications is to have an accurate localization of the vehicle. In outdoor, open environments this problem is usually addressed by an onboard Global Navigation Satellite System (GNSS) receiver. However, such a solution can be unavailable, infeasible, or inaccurate in, for example, GNSS-denied or low-gain environments (indoors), urban areas or offline contexts (recorded video footage). One of the potential solutions to this problem and an emerging field is vision-based localization, particularly, using an onboard ground-facing camera in conjunction with readily available orthographic imagery (satellite or aerial) of the immediate area.

Vision-based approaches may be used both in absolute and relative pose estimation and pose refinement (pose refers to the combination of position and orientation). Furthermore, visual localization often enables determining the physical locations of ground objects in the camera's field of view. These qualities make vision-based aerial localization systems strategically, academically, and financially interesting topic. (Lu *et al.*, 2018; Belmonte, Morales and Fernández-Caballero, 2019; Couturier and Akhloufi, 2021)

The case company, Huld Ltd., has been developing a GNSS-free visual localization solution for off-the-shelf UAVs. Work on the solution has motivated internal research on state-of-the-art and alternative methods for visual localization, including this thesis. However, the contents presented in this thesis are not directly related to Huld's actual solution and do not describe its working

principle nor its performance characteristics. Instead, this thesis proposes, explores, and evaluates a possibly novel method for estimating similarity or alignment between images in the context of visual localization. It presents a robust image matching score along with a way of avoiding two common sources of error in feature-based image registration pipelines, the feature matching and outlier filtering steps.

This work tries to solve the four-dimensional (4D) problem of finding the position and heading of a moving UAV (its camera), using only a video feed from the UAV, aerial orthophoto, and a rough initial estimate of its position. The presented solution is derived from existing works based on Monte Carlo localization (Elfring, Torta and van de Molengraft, 2021). The solution employs the proposed image matching score as the fitness function for hypotheses weighting in the Monte Carlo localization. The proposed solution's performance is compared against a benchmark solution which employs a frequently used fitness function in this context, the zero-mean normalized cross-correlation (ZNCC) (Roma, Santos-Victor and Tomé, 2002; Jurevičius, Marcinkevičius and Šeibokas, 2019). The comparison is performed on real-world footage that follows the same trajectory during different times of day and seasons.

Image registration, or alignment, is an intrinsic part of most published absolute visual localization methods for UAVs. In practical applications, finding a good image alignment is a challenging task if the task is not restricted to a very specific set of conditions. Achieving good image matches is complicated especially by the diverse ways image appearance can change, these can be for example changes due to temporal effects, viewpoint changes, different data collection and representation, etc. The method proposed in this work seeks to provide the foundation for a simple yet robust and flexible approach for addressing the viewpoint and appearance changes encountered in visual localization problems. (Couturier and Akhloufi, 2021)

The thesis is structured into 6 main chapters, including this chapter. Chapter 2 presents more information on the study, how it was performed and what data

was used. Chapter 3 explains some important theoretical aspects of the work and gives an overview of the most related or relevant works. Chapter 4 delves deeper into the details of the study and its implementation. Chapter 5 explores achieved results and describes how they were achieved. Finally, chapter 6 summarizes the work and discusses its findings and potential further work. The work also contains two appendices offering detailed results.

2 Materials and methods

The primary goal of the study is to assess the feasibility of the proposed approach and its potential benefits and drawbacks. The goal is achieved by developing, implementing, and evaluating a proof-of-concept solution together with finding, implementing, evaluating, and comparing it with a suitable benchmark solution.

The study consisted of the following main parts: literature review, design and planning of methods for the study, dataset acquisition, iterative implementation, and evaluation. This chapter describes the selected approach, the design of the study, and the evaluation dataset. The literature review is covered in section 3.1 and the implementation of the study, evaluation, and findings are described in chapters 4 and 5.

2.1 Overview of studied solutions

This section outlines the concept of the studied method, how it is used in the proposed localization solution (PLS), and how it relates to the benchmark localization solution (BLS) while highlighting the key differences between the proposed and benchmark solutions. A more detailed description and implementation details are described in chapter 4.

The solutions consist of one principal component, the state estimation, and its several subcomponents: state fitness scoring, motion estimation, a reference

database; and trivial helper components such as input and output handling. An overview diagram of the components is shown in Figure 1.

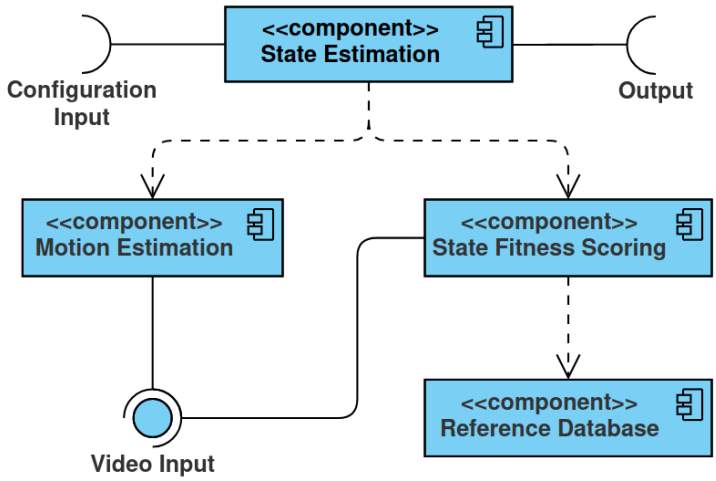


Figure 1. Overview diagram of the proposed visual localization solution.

As the introduction reveals, the visual localization solution is a Monte Carlo localization method, also known as a particle filter (PF). The PF represents the state estimation component, as a Monte Carlo method, the estimation follows a stochastic process. The working principle of a PF is to maintain a set of hypotheses for the true state and their likelihoods based on sensory evidence. The PF algorithm is explained in more detail in section 3.2. With the intention to highlight the properties of the central idea in this thesis, the image alignment score, a basic form of the PF algorithm is used. Since the core focus of this work is on the image alignment measurement, it is the main and only major difference between the proposed and benchmark solutions. The benchmark solution employs ZNCC, its definition can be found in the work of Jurevičius et al. (2019) and in Roma et al. (2002) with a comparison of additional cross-correlation methods.

The essential idea of the novel image alignment score is composed of 4 main steps. The steps are firstly described in their generic and abstract form, but their specific instantiation is explained in the implementation details, in section 4.5. Conceptually, the essential steps of the proposed image alignment score are:

1. Choose a subset of predicted or assumed correspondences between a query image and a reference image. An example of such correspondence is shown in Figure 2.

2. Extract local image patches of corresponding areas from step 1 in both images. Again, an example of extracted patches is shown in Figure 2.

3. Measure image similarity between the individual corresponding pairs of image patches using conventional methods. For example, using template matching or the distance between patch descriptors.

4. Calculate a single, robust representation of the patch-wise similarities from step 3, such as the median.

A patch correspondence between two images is illustrated in Figure 2. The images in the pair are often referred to as a reference image (Image A) and a query image (Image B). The correspondences between the images may be derived, for example, from the relative camera pose or using feature matching methods.

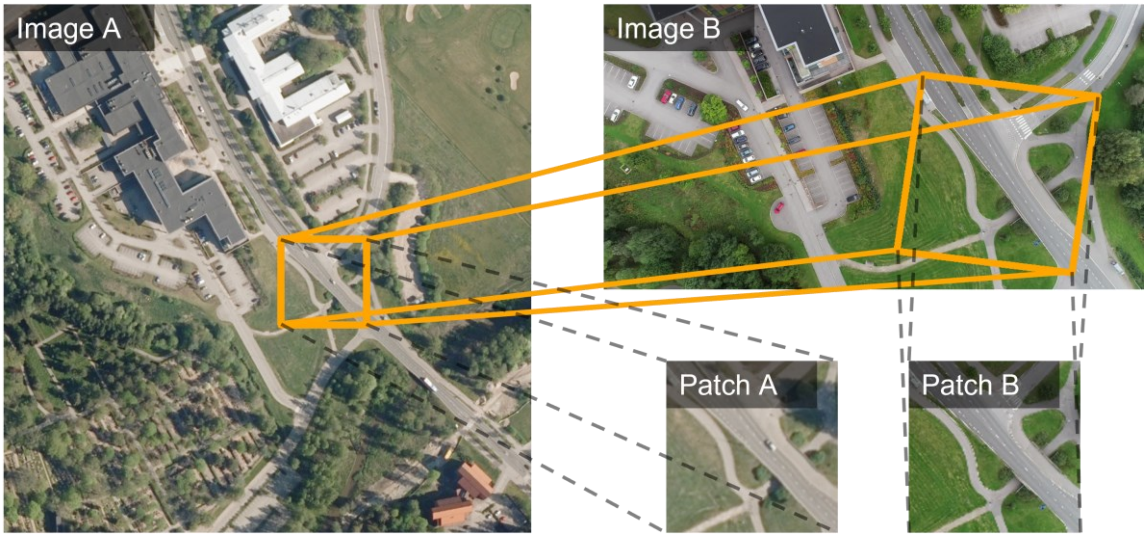


Figure 2. Reference image patch to query image patch correspondence. Image A is taken from the (NLS, 2020) dataset.

A true patch correspondence is shown in Figure 2. However, in real-world localization applications, such correspondences are unknown and need to be found. In the implemented PLS, the query-to-reference correspondences are derived from each state hypothesis in the particle filter, i.e., from the candidate poses. Also consider, that if the topology in the reference image is known, the local patches of the reference image can be defined in advance and their coordinates can be then reprojected into the query image space using the (predicted) relative pose. This means that precomputing and caching part of the computation is possible.

In localization problems, various sensors and possibly control inputs are used to estimate and predict the subject's motion. In GNSS-free UAV localization, at least an inertial measurement unit (IMU) is typically utilized. However, the used dataset does not contain such information, and therefore motion estimation needs to be based purely on visual information in this work. The selected motion estimation method is a basic, feature-based visual odometry with the assumption of planar topology, further explained in section 4.2. The planar assumption does not hold but is adequate for the dataset. Moreover, the PF should be robust to the resulting inaccuracies. Single-camera visual odometry cannot estimate the scale of the motion alone, instead, the scale of the motion can be implied individually by the estimated altitude of each state hypothesis in the PF.

The purpose of the reference database is to provide information against which the PF's state proposals can be gauged. In both solutions, the reference database ultimately interfaces georeferenced orthographic imagery. In the case of the benchmark solution, the database is queried individually for each state hypothesis, extracting, and warping a relevant portion of the underlying orthographic imagery. On the other hand, the proposed solution takes advantage of the possibility of caching computation and introduces another cached spatial database layer.

2.2 Study design

The case company has an existing body of work for comparing different approaches in visual localization across different metrics. However, due to intellectual property concerns, the work and its results cannot be made public. This reason further makes the case for implementing a method for direct comparison to increase the objectivity of evaluation and results. This section gives an overview of how the study was planned and performed considering the objective of assessing the feasibility of the proposed approach, its characteristics, benefits, and drawbacks.

Upon conceptualization of the PLS and the image matching score, a preliminary study was performed to review related works. After a brief review of the works, an initial plan for the study was devised. The initial plan consisted of a thorough review of related works, finding, and selecting appropriate works and datasets for evaluation, then the implementation, evaluation, and analysis of the solutions and finally the presentation of the achieved results.

Following the review, the breadth of the evaluation was limited to one work only for the benchmarking comparison. This decision was taken to limit the scope and meet resourcing constraints. The work of Jurevičius et al. (2019) was chosen as a feasible basis for the BLS. It was decided to omit the adaptive particle resampling from the original solution in this work's implementation of the BLS. The final evaluation dataset was selected (described in the next section), and a synthetic dataset for parameter tuning was created.

The study design was to implement the proposed and benchmark solutions and tune their parameters on a dataset similar but unrelated to the evaluation dataset. The intention of this approach was to avoid biasing the parameters towards the evaluation dataset. However, upon initial evaluation of the BLS, it was discovered that despite reaching similar localization performance on the synthetic dataset like Jurevičius et al. (2019), the performance does not

translate to the evaluation dataset. Along with additional difficulties faced during the implementation, the study design had to be modified.

The final adaptation of the study design abandoned the synthetic dataset and instead designated one of the videos from the evaluation dataset to be used for tuning. The flaws of deviating further from the original plan are to some extent less objective and directly comparable results with other works. However, the process emphasizes the practical challenges of the solutions.

2.3 Evaluation datasets

One of the datasets the case company obtained for validation of its solution has been made available for this thesis. The dataset consists of 28 aerial video recordings, each covering the same trajectory but in changing conditions. The recordings were taken between September 2020 and November 2020 in Espoo, Finland, using the commercial off-the-shelf DJI Phantom 4 Pro quadcopter without modifications. The flight track is shown in Figure 3, the mean length of the path is about 3250 m lasting about 400 s, and the approximate flight altitude is 140 m above the ground. The recordings have an unknown but consistent camera heading offset relative to the direction of the flight due to equipment error. The GNSS ground truth (GT) data is available at about 1 Hz sampling frequency; however, the data is not synchronized with the video recordings. The flight cruising speed was set to 10 m/s on all straight sections of the track.

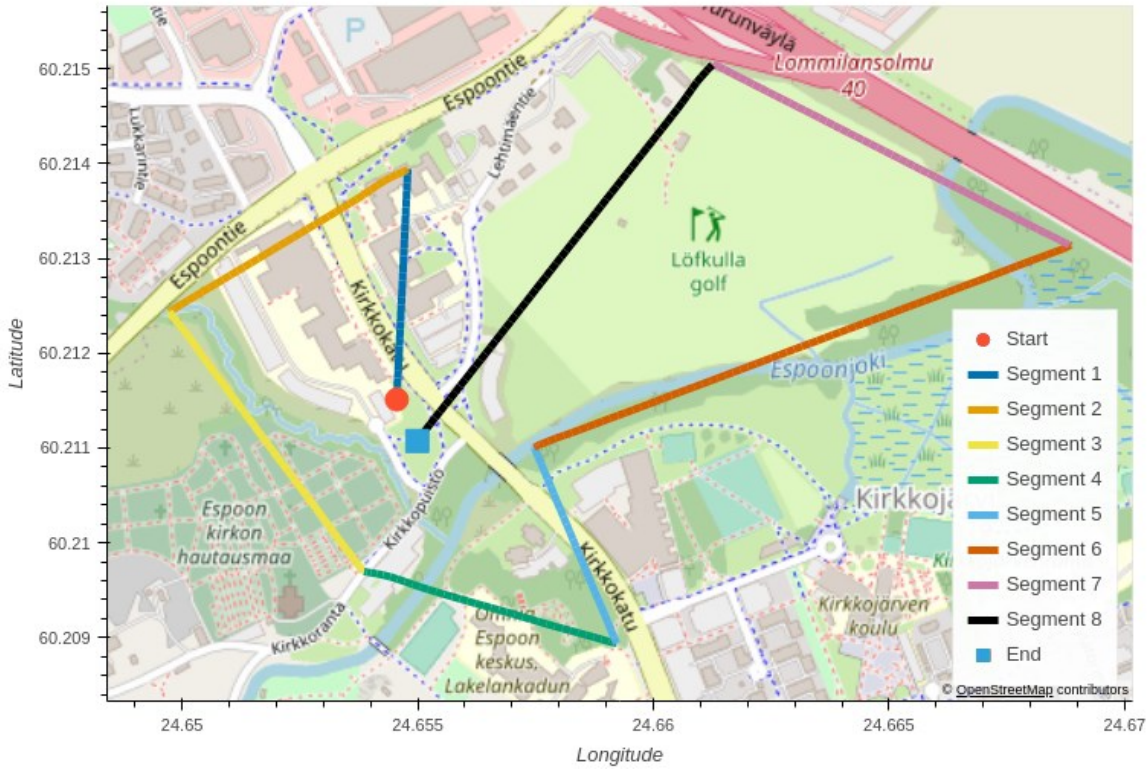


Figure 3. Test data flight path.

Figure 3 illustrates the path the UAV followed during the flight recording missions. It is often convenient to refer to the individual segments of the GT path when commenting on results or other aspects related to the track. The segments are marked by distinct colours in the figure to remove ambiguity. Segments 1, 2, and 5 cover areas containing very distinguishable manmade objects such as buildings, roads, and parking lots. Segment 3 starts by departing from a main road, down along a brook lined with trees and grassland on the side, then traversing over a cemetery. Segment 4 crosses a small river, and contains a playing field, some buildings, and a forest patch on top of a small hill. Segment 6 follows a small river, again lined with trees and a flood area. Segment 7 keeps to a highway ending in a fork but with a monotonous look in the middle section. The final segment flies over a golf course, ending near the take-off location.

Even though the flights happened at different times, many recordings are very similar in their appearance. To simplify the analysis but maintain a wide breadth

of conditions, 6 video recordings were manually selected for evaluation. Sample frames from the chosen recordings can be seen in Figure 4 and Figure 5 along with the associated names of the videos for simpler referring and mnemonic associations. Figure 4 shows the first frames of the selected videos for a comparison of their differences and a more intuitive understanding of their qualities. Figure 5 shows example frames at different moments in the recordings and associates their positions along the flight path over a portion of the reference imagery.

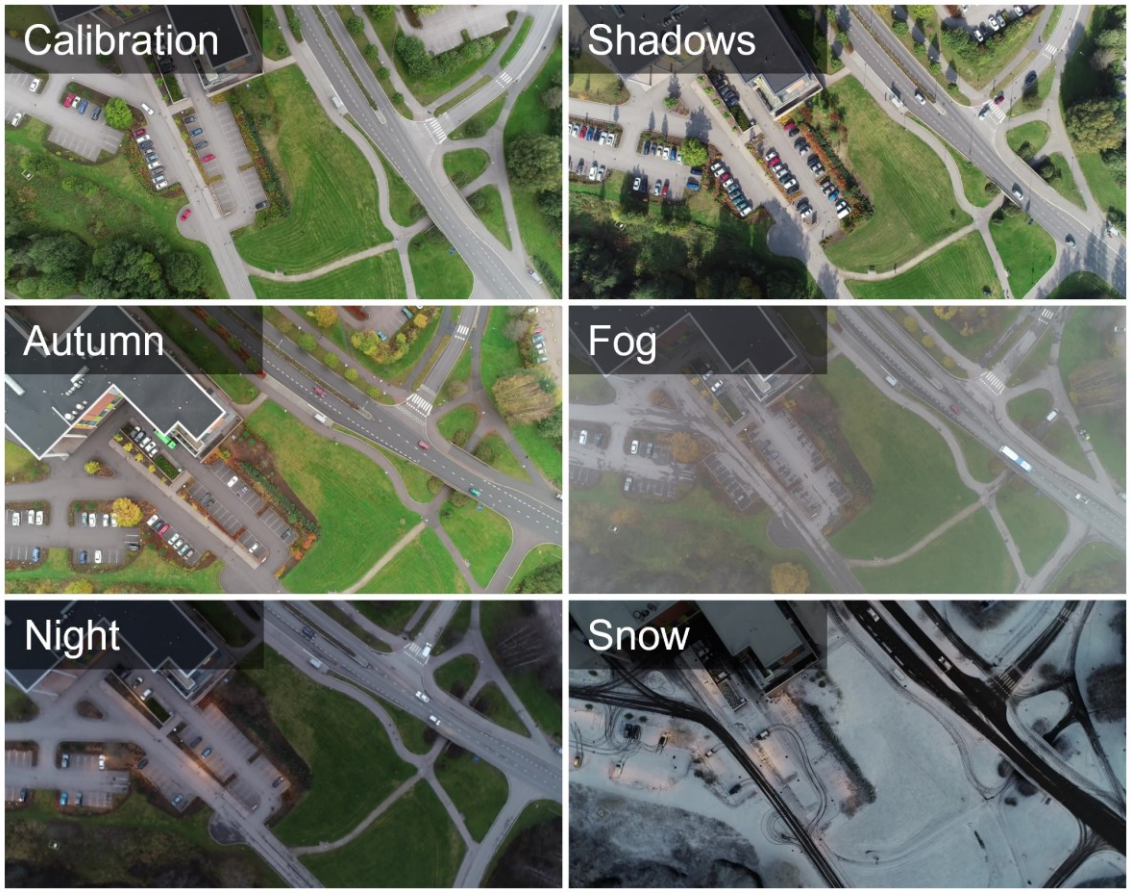


Figure 4. First frames of the selected videos.

The “Calibration” video was used for parameter tuning as described in the previous section and is most similar in appearance to the obtained reference imagery (see Figure 6). The “Shadows” video is characterised by its sharp light and pronounced shadows. In contrast, the “Autumn” video features soft light but maintains high image sharpness and clarity, and as the name suggests, it also

features colourful autumn foliage. The last three videos share much poorer optical properties such as blur and softness due to the low amount of ambient light present. The “Fog” video contains parts with very poor visibility owing to dense fog. The “Night” video was recorded in the late evening hours, the “Snow” video was recorded at dawn, both include street and vehicle lights. The surfaces are covered by up to 5 cm of snow in the “Snow” video.

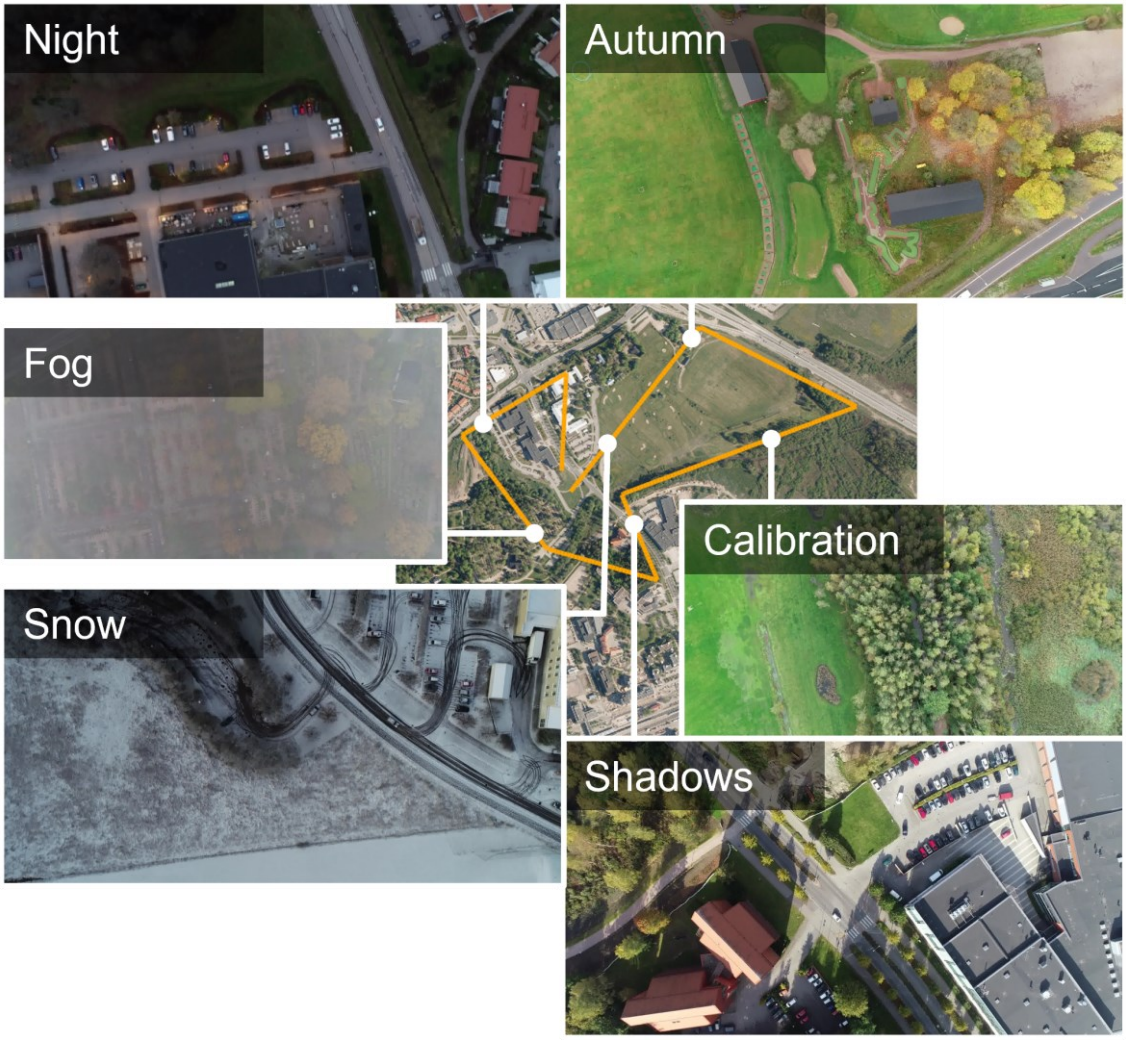


Figure 5. A sample set of frames from the chosen test dataset.

Figure 5 gives a broader overview of the variety of conditions encountered in the dataset and the imagery they are compared against during evaluation. A closer view of the reference imagery is shown in Figure 6.

The underlying reference dataset used in the work is a subset of aerial orthographic imagery from the “NLS orthophotos” dataset produced and published by the National Land Survey of Finland (NLS, 2020). The obtained part of the dataset was published in August 2020. Its resolution is 0.5 m per pixel with an accuracy of 0.5 – 2 m. To communicate the appearance and quality of the dataset a sample view is shown in Figure 6.

Figure 6. Sample view of the reference orthographic imagery. (NLS, 2020)

The example image in Figure 6 covers the flight trajectory and the field of view (FOV) of each frame in the video dataset (compare with Figure 3 and Figure 5).

3 Theoretical background

This chapter briefly covers the main topics and principles used in throughout this work. In the first section, the most relevant and related works are indicated. The second section describes some general aspects of localization and state estimation with a more detailed explanation of particle filtering. Lastly, the third section gives an overview of the computer vision principles used in this study. Additional materials and resources are indicated in the chapter.

3.1 Related works

A wide range of visual localization systems has been explored in the literature and applications. Dozens of research outputs have been published in the context of visual UAV localization alone, with many different approaches taken.

Tabulation and review of works on absolute visual localization for UAVs in general can be found in the work of Couturier and Akhloufi (2021). The survey by Lu et al. (2018) also includes relative-only vision-based systems but is less recent.

The search for related works was performed with the help of a search engine using keywords and progressively by exploring found works and browsing their relevant citations. No works describing an approach like the PLS have been found in the context of absolute UAV localization nor in broader computer vision contexts.

The conceptually closest work found is Mantelli et al. (2019) where the authors build their localization solution similarly to the one presented in this thesis and devise an analogous image matching method where they use a feature descriptor in a non-traditional way. In contrast, their scoring method works on the pixel level with a complex handcrafted and highly specific pipeline intended to increase robustness. The advantage of their method performing at the pixel level seems to be a very high speed of execution, but according to the results, the method suffers from a rapid increase in uncertainty in homogenous areas.

This thesis is also partly inspired by the works of Kinnari et al. (2021; 2022) in which the authors examine different image matching methods using PF for absolute visual-inertial localization for UAVs.

As described earlier, the BLS is based on the work of Jurevičius et al. (2019) where the authors evaluate and compare different score conversion functions for the ZNCC matching score.

3.2 Localization and state estimation

In robotics, general navigation, object tracking and in numerous other fields the problem of physical localization frequently comes up. Many classes of methods for localizing an object exist. Sensor measurements are noisy and there are often various uncertainties involved in localization processes. Therefore, a common trait among localization approaches is the need for a state estimation method. State estimation methods model the dynamics of the localized systems and integrate measurements and control inputs to provide a robust belief about the true state of the system. (Thrun, Wolfram and Fox, 2005; Alkendi, Seneviratne and Zweiri, 2021)

All state estimation methods involve a set of assumptions on their applicability and different performance characteristics. Choosing a particular state estimation method frequently involves a detailed analysis of the system at hand and considerations of the advantages and disadvantages of the applicable methods. However, under some assumptions, optimal methods have already been developed, in such cases, the choice is then straightforward. An example of an optimal state estimation method is the Kalman filter (Kalman, 1960) for linear processes with normally distributed (Gaussian) noise, famously used in the Apollo missions (Mcgee, Schmidt and Schmidt, 1985). Although, like in the case of this work, systems often follow more complex distributions or exhibit non-linear behaviours. In such systems, generalizations, and extensions of the Kalman filter are frequently used but also, for example, grid-based filters or the particle filter. (Thrun, Wolfram and Fox, 2005)

The particle filter (PF) is a stochastic state estimation method that is in principle able to estimate the state of an arbitrarily complex system. Besides its generality, its advantages include flexibility and its conceptual and implementational simplicity. Among its drawbacks are non-deterministic estimation (although some deterministic versions exist), the difficulty of parameter tuning, and potentially great computational costs in high-dimensional problems.

The PF performs state estimation by tracking a set of state-space samples and their associated importance weights. The samples are commonly referred to as particles, hence the name. The particles together with the set of weights represent a discretized, approximate probability distribution of the estimated state. Another way to understand the role of particles is that each particle represents a hypothesis that the system's true state is the particle itself. The associated weight then represents the likelihood of the hypothesis being true. During its operation, a particle filter uses Bayesian inference to improve its estimate and resampling of the state space to explore new hypotheses.

The conceptual explanation of a PF is described in the rest of this section. The formal definition of a PF, its detailed explanation, and design considerations can be found for instance in Elfring et al. (2021) and in Yozevitch et al. (2017). The steps of the PF algorithm are outlined below, with their explanation following subsequently:

1. Initialize a set of particles.

2. Update particle weights to find the posterior.

3. Optionally resample particles.

4. Predict the next state by propagating particles.

5. Continue from step 2 (new iteration).

The initialization of particles in the first step may be done for example at random or exploit prior knowledge. It should be set up so that some particles are likely to be near the true state, for example by densely sampling the search space, otherwise, the process may not converge to the true state.

The second step is often crucial for a good performance of a PF. The prior belief on the state is improved by integrating new evidence, i.e., observations such as sensor measurements. The integration of evidence into a posterior

belief is achieved by calculating the individual probability of each particle encountering the evidence. The calculated probabilities are then assigned as the particle weight and they shall be normalized so that their sum is equal to 1, to form a well-defined discrete probability distribution. It is rarely possible to know the true probability of the evidence occurring, therefore the design of a PF often involves the art of finding a good approximation of a such function.

The resampling step can be viewed as a reinitialization of the particles with good prior knowledge. This step typically samples from the posterior distribution found in the second step. Ideally, the particle resampling would be done from the continuous posterior distribution. But in practice, it is often done by randomly choosing the existing particles proportionally to their weights. Sampling from the existing states has the effect of removing particles with low weights and copying particles with higher weights. However, multiple particles representing the same state do not provide novel information and thus, a small amount of noise is added to the particles after resampling or during the fourth step. Commonly, advanced particle filters modify the number of particles during the resampling step according to convergence estimates to save computational resources. Furthermore, the resampling does not have to be performed at every iteration since it is a potentially computationally expensive operation. Various strategies for deciding when to resample exist.

The resampling algorithm used in this work is stratified resampling. Given N particles, stratified resampling splits the particles into N equally sized bins based on the particle weights (a particle with a weight of more than $1/\mathrm{N}$ is in multiple bins simultaneously). A single particle is then selected from each bin uniformly at random.

In the prediction step, the particle states are propagated according to the system model and inputs. For example, if the state space is tracking the position and velocity of an object, and the object received a command to change its velocity, the prediction step adjusts the velocity component of each

particle according to the (noisy) input together with updating the position according to the motion model and the previous state.

The steps 2 – 4 represent one iteration, or generation, of the particle filter. The particle set resulting from the fourth step becomes the prior for the new iteration. The new iteration continues from the second step upon obtaining new evidence.

An example of the progress of PF posterior states is shown in Figure 7. States from initialization to convergence near the true position are shown. The image grid in the figure is in the left-to-right, then top-to-bottom order. The example was created with the implemented solution and with intentionally poor parameter selection to visually emphasise the state propagation and convergence. Only a subset of the PF iterations is shown in order to capture the main properties. The particle initial positions are sampled from the normal distribution around the true position. The true positions are represented by the orange crosses. The particle heading (direction) is initialized uniformly at random in all directions. In the figure, the particle horizontal positions are represented by blue discs drawn over a map and the colour shades represent the altitude. A particle's diameter is proportional to its (posterior) weight while its heading is depicted by an arrow.

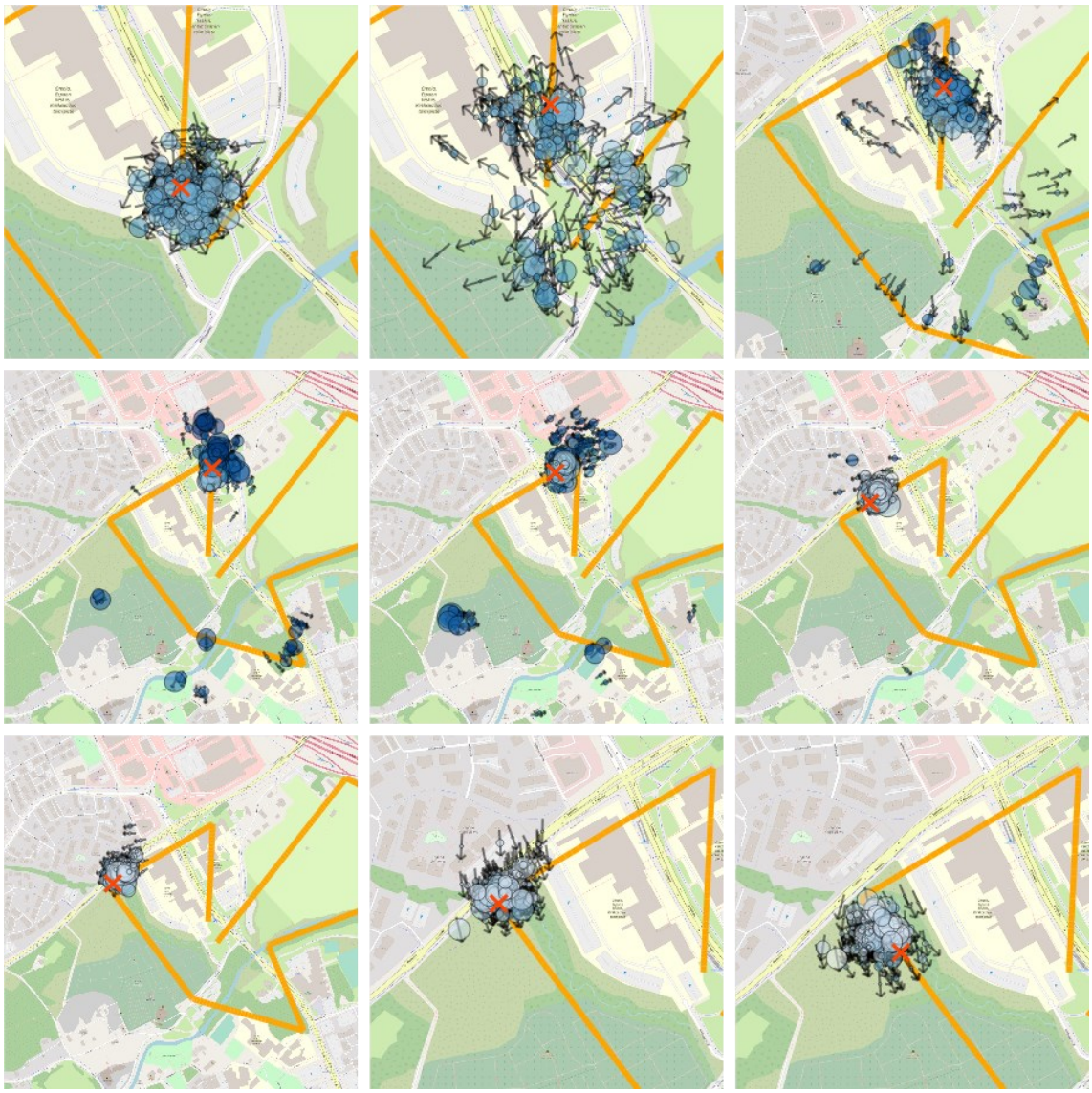


Figure 7. Example progress of posterior states of a particle filter.

Observe how the particles start spreading in almost all directions in the second image in Figure 7 and how it appears that more particles are concentrated around the true position. Some directions were calculated to be highly unlikely and therefore they did not get resampled while the converse applies to states closer to the GT. In the third image, most particles are already concentrated at the GT position, but some still continue spreading away, although with low weights. In the two following images, some of the stray particles scored well in the measurement step and continue surviving. However, after a change in direction of the GT, the stray particles rapidly dissipate. The subsequent images show a converged state near the GT, with a cloud of particles exploring states

further away. Another vital observation throughout the figure is how complex and multimodal the estimated distribution can be.

In this work, the particle filter was chosen for its suitability for the task, its frequent use in similar systems and for the simplicity of implementation.

3.3 Computer vision principles

For thorough coverage of many computer vision topics and methods, it is recommended to read Computer Vision: Algorithms and Applications, Second Edition by Richard Szeliski (Szeliski, 2022). The most relevant book chapters for this study are chapters 2, 7, 8, and 9.

Many calculations that deal with image projections and cameras depend on a mathematical model of the camera and its position and orientation. The camera position and orientation are together referred to as a pose or the extrinsic (camera) parameters. The camera model used in this work is a simple pinhole camera model. The pinhole camera models image formation through a single-point centre of projection. The model has 5 degrees of freedom and depends on the focal length, the image size (resolution) and the centre of projection in the image. The model parameters are referred to as camera intrinsic parameters.

Naturally, the actual image formation is more complex and forms the image through a set of lenses and an aperture. The real process introduces some image artefacts and distortion effects. Camera calibration procedures exist to estimate the distortion model and its parameters. The image can then be undistorted to remain closer to the pinhole camera model. However, the distortion parameters have not been measured and are therefore ignored in this study.

A commonly used concept in this work is that of image features. An image feature represents a certain trait of an image or its part. Though the term is

imprecise and may refer to different properties, this thesis uses the term to refer to local regions of an image, such as the patches shown in Figure 2.

Image features are commonly used to find spatial relationships between two or more images. This is done by first identifying the features (feature detection) in the images and reducing their dimensionality (feature description). Subsequently, the features are matched which enables the reconstruction of the spatial relationships. An illustration of a conventional usage of features for finding correspondences is shown in Figure 8 (described later).

A commonly desired quality of feature detectors is to discover features that are repeatably and reliably discoverable even under some viewpoint and appearance changes. Ideally, the same world points would be consistently detected in different images. Detectors often try to find features across many scales, orientations, and positions in the image. The detections are also referred to as (feature) keypoints. This work often represents a keypoint with a local affine frame (LAF) which defines a parallelogram region in the image (Obdržálek and Matas, 2002).

Feature description is typically performed on the pixels in the local neighbourhood of a feature keypoint in the image, according to the scale, orientation, or even more attributes. The description attempts to encode some of the innate properties of the surrounding area with the intention to make different features easily comparable. The result of a feature description is usually a simple, multi-dimensional vector, comparable using either the Euclidean or the Hamming norm.

After feature detection and detection, feature matching can be performed. Normally, this is accomplished with some nearest-neighbour search method used to find the closest pairs of feature descriptors between the descriptor sets. However, such matching typically produces many false matches. To filter out the false matches, descriptor- and problem-specific filtering techniques are applied. A simple cross-check that the nearest-neighbour correspondence is bi-

directional or a ratio criterion that the nearest-neighbour is at most a fraction of the distance away than the second nearest-neighbour is, is commonly a good first filtering stage (Lowe, 2004). The final stage of a feature matching pipeline typically involves a robust method for estimating the spatial relationships. An adaptation of the random sample consensus algorithm, also known as RANSAC, is generally employed (Fischler and Bolles, 1981). The algorithm performs model regression robust to a very high proportion of outliers. However, with many features and many outliers, it can be computationally too expensive. Furthermore, it is not guaranteed to find the inlier subset due to its stochastic nature.

Many dozens of handcrafted and machine-learning methods for feature description and feature detection exist. There are even hybrid pipelines which perform joint detection, description, and matching or pipelines that find the correspondences in unusual ways. However, none of the approaches can be completely relied upon, owing to the difficulty and ambiguity of the problem, i.e., perspective, modal, temporal changes, homogeneous regions, etc.

An example of feature-based matching between images is shown in Figure 8. The locations of the detected features are visualized as blue points, other properties of the keypoints are ignored in the visualization but considered for matching. The found true feature correspondences are paired with yellow line segments and the false correspondences are paired with pink segments. The location and orientation of the query image (right) is marked with a black rectangle in the reference image (left).

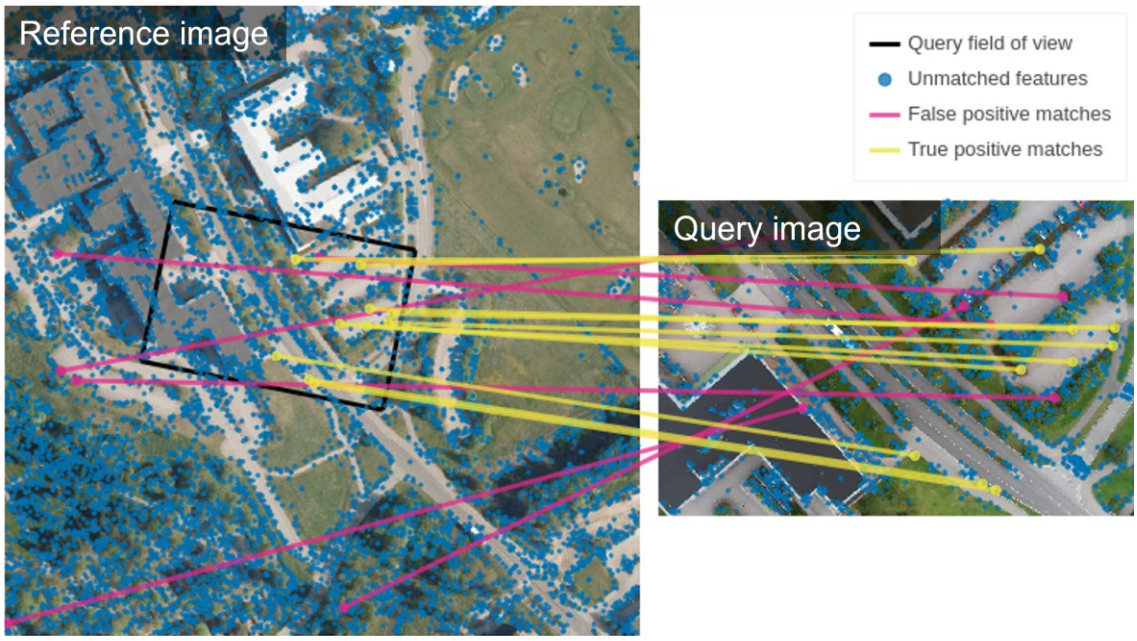


Figure 8. An example of feature matching between two images.

The used feature detection and description method in Figure 8 is SIFT developed by Lowe (2004). The features were first matched with the nearest-neighbour search and pre-filtered with the ratio criterion. Then the matches were filtered with the random sample consensus method while estimating the similarity transform between the images. The right image is taken from the “Calibration” video and the left image is part of the NLS dataset.

Notice how no keypoints appear in the homogeneous areas of the images in Figure 8, e.g., patches of grass or roofs. Also, observe that many features present in both images are unmatched after the initial filtering and that some very poor false matches passed through the filtering. However, the false correspondences were correctly rejected as outliers.

4 Implementation

This chapter presents a detailed description of the final implementation of both solutions. Important implementation insights are presented, and design choices are stated or explained. However, most implementation details and optimizations are left out for the sake of clarity.

Everything regarding the code in implementation, evaluation, and analysis was written in Python with the help of and thanks to the availability of many applicable libraries. The architecture and implementation of the solutions is straightforward since the execution time was not a significant factor in this work. Consequently, all steps in the process are run sequentially, except for some delegated computations which are automatically parallelized. A combination of functional and object-oriented programming principles was applied in the architecture.

A set of helper methods to execute each high-level PF step was defined: initialization, particle propagation from visual odometry, measurement step, and resampling step with jitter (noise). These are called sequentially on the PF with the input video frames and evaluation parameters given as arguments. The PF state is archived for later analysis after every measurement step.

4.1 Particle filter representation

There are 5 values that need to be tracked for each particle: 3 dimensions for the position, 1 for orientation (heading), and 1 for representing the particle weight. Hence, it is natural to represent a particle as a 5-dimensional vector in an array of particles, i.e., a $5 \times N$ matrix where N is the number of particles.

To simplify analysis and work in an intuitive coordinate system, a local projected coordinate reference system (CRS) is created at the approximate origin of the evaluation dataset. Specifically, the transverse Mercator projection (Snyder, 1993) centred at 24° 39' 16.75" east longitude and 60° 12' 41.51" north latitude with the GRS80 (Moritz, 1980) geodetic reference system used.

4.2 Visual Odometry

The original work of Jurevičius et al. (2019) on which the BLS is based, uses a solution for visual odometry by Forster et al. (2014). However, due to integration

issues, their solution was replaced with a simplistic VO solution for the particle propagation (prediction) step.

The implemented VO solution estimates the 2D Euclidean similarity transform between a pair of consecutive images. The similarity transform is then decomposed into the constituent angle of rotation (heading change), scale factor (altitude change), and a horizontal translation vector. The decomposed scale factor and translation vector are obtained in pixel units and must be combined and scaled to meters. But because of not using IMU data or another source of scale, assumptions about the relative orientation between the terrain and camera must be made to estimate the scale of translation. Assuming the camera is perpendicular to and looking at the ground, and given intrinsic camera parameters, the scale of the translation (both vertical and horizontal) can be determined from the altitude individually for each particle. However, the resulting translation remains in the camera frame of reference and must be rotated according to each particle's heading and the newly estimated rotation. Finally, the position and heading of each particle are updated accordingly.

The similarity transform is estimated using a feature-based approach with the help of OpenCV functionality. Namely, the OpenCV implementations of the ORB detector and the Beblid feature descriptor (with the Beblid scale of 0.9 and considering at most 2000 features). The extracted features are matched using an exhaustive nearest-neighbour search. The model regression and outlier rejection are then performed using the random sample consensus algorithm.

4.3 Particle weighting

The particle weighting process consists of scoring each particle and a joint normalization of the scores into probability masses. To accommodate both solutions with the same architecture, the particle scoring is delegated to a scoring function. For each particle, the scoring function receives the current query image, the camera intrinsic parameters and the particle's pose hypothesis.

In the initial iterations of the implementation, the score normalization methods described by Jurevičius et al. (2019) were tried with the BLS. However, the convergence properties of the PF were poor on the “Calibration” video. In an analysis of scores produced with the BLS’s scoring (described in the following section), a substantial overlap between the scores of random and true corresponding images was found, a similar finding was also noted by Kinnari et al. (2021). There was a tendency for good correspondences to have marginally higher values than poorer correspondences. However, the level of separation was floating, and its mean also depended on the image contents. These findings led to a formulation of the score conversion function that performs a joint conversion over all particles and includes weight normalization:

$$
\begin{aligned}
p_{10\%} &= \mathrm{percentile}_{10\%}\ \bar{s} \\
p_{90\%} &= \mathrm{percentile}_{90\%}\ \bar{s} \\
s_i^n &= \frac{s_i - p_{10\%}}{p_{90\%} - p_{10\%}} \\
s_i^c &= \mathrm{clamp}_{[0,1]}\ s_i^n \\
s_i^e &= \exp 2 s_i^c \\
w_i &= \frac{s_i^e}{\sum_{i=0}^{N} s_i^e}
\end{aligned}
$$

Where $\bar{s}$ represents a vector composed of the matching scores, s_i refers to the score of the i-th particle, and w_i to its normalized weight. The $\mathrm{percentile}_{n\%}\ \bar{x}$ function returns the n-th percentile of the vector $\bar{x}$. The $\mathrm{clamp}_{[0,1]}\ x$ restricts the value of x to the $[0,1]$ interval. Notice, how this is related to the softmax normalization used by Jurevičius et al. (2019) but it performs an adaptive normalization of the scores based on their current values and tries to be robust to outliers.

4.4 Image matching scoring in the benchmark solution

Calculating the score of a particle in the BLS consists of first retrieving a reference image for the particle's pose hypothesis and then calculating the ZNCC between the reference and the query image.

Retrieving the reference image involves finding the camera field of view (FOV) given the particle pose and the camera intrinsic parameters. The FOV is determined by casting rays from the centre of the projection through the image corners and finding the intersections with the ground plane. Subsequently, an image is requested from the reference imagery dataset at an appropriate scale. The Rasterio library with the GDAL backend is used as an interface for reading the reference dataset source, and its virtual warping functionality is used to transparently reproject the data to the local CRS (Gillies, 2013; GDAL/OGR contributors, 2022). Finally, the retrieved image needs to be warped to match the orientation of the FOV.

The query and the reference images are converted to grayscale. Then the correlation coefficient between the images is measured and returned. The output range of the ZNCC is $[-1, 1]$.

In attempts to improve the convergence of the BLS, different image augmentation techniques were also applied, for example, edge detection and sharpening, gaussian blurring, and histogram equalization. However, their benefits and drawbacks were often inconclusive, and the solution would further deviate from the work of Jurevičius et al. (2019). Hence these were omitted in the final implementation.

4.5 Image matching scoring in the proposed solution

A FOV associated with a given particle is found analogously as in the BLS. Instead of requesting a reference image like in the BLS scoring, a spatial

feature database is queried for features found within the FOV. The implementation of the database is described in the next section.

The spatial database returns a set of features – feature descriptors and the areas they represent in the underlying reference map. Among the found features, 64 are randomly chosen. The choice is done with replacement to avoid edge cases when there are fewer than 64 features found.

The points representing each feature's ground area are projected into the image space of the query image according to the pose proposal by the given particle. And local affine frames (LAFs) are constructed from the projected points. The Kornia library's pyramidal patch extraction method is used to extract the image patches corresponding to each LAF (Riba et al., 2020).

Each image patch is then described using the same descriptor as used in reference feature extraction. All the 64 corresponding vector pairs are multiplied using the dot product, thus providing the cosine similarity measure, assuming that the feature descriptor vectors are L2-norm normalized.

Finally, the 80-th percentile of the cosine similarities is chosen as the representative value and is returned as the particle score. Note that this score has the same output range of $[-1, 1]$ as ZNCC in the benchmark solution. The 80-th percentile was chosen as an educated guess after a brief exploration of histograms and covariance matrices of several compared features.

4.6 Reference feature database

The purpose of the reference feature database is to avoid the need to repeatedly request reference images and extract their features because both processes are resource intensive.

The database contains feature descriptors and the definitions of areas that they describe. The areas are represented by a triplet of 3D world points. The coordinates of the points are selected in a way that makes it straightforward to

determine the corresponding LAF of the feature in their image projection. Notice, that the representation can accommodate different elevations of each point, thus in future, non-planar terrain topology could be exploited without changes to the database.

The database performs the spatial search in two stages to increase efficiency: a coarse search of large groupings of features, and a fine search in each grouping. The feature groupings cover larger areas and are realized as partially overlapping tiles in a regular grid, they will be referred to as tiles in the rest of the section. The coarse stage is responsible for finding tiles overlapping with the FOV, but also for dynamically loading, caching, and creating the feature tiles. The fine search finds individual features within relevant feature tiles. An R-tree implementation by PyGEOS was chosen for both search stages (Guttman, 1984; Wel et al., 2022).

A tile is created by determining the tile's ground extent and fetching its reference image using the same backend and reference dataset as in the BLS. Feature keypoints are then identified and described, the feature descriptors and their corresponding georeferenced area definition (explained earlier) are saved into a uniquely identified file.

Atypically, no keypoint detector was chosen for finding keypoints in the reference tiles. Originally, it was intended to find and use a detector that can detect features approximately uniformly across the image. Many trained detectors may exhibit such behaviour, for example, the detector component described by Revaud et al. (2019) with a 0 reliability threshold. The intuition was to provide some response for the PF even in homogenous regions. However, the implementation of the reference feature database was first tested with a placeholder "detector" which simply tiled the reference image into square image patches with a width of 32 pixels (16 m on the ground) and an overlap of 16 pixels (8 m on the ground). Testing revealed that even this simplistic approach caused particles in the PF to converge quickly. Considering this finding interesting, this implementation was kept for the final evaluation.

The HardNet descriptor was selected for the feature patch description, The reasons for the choice are HardNet's availability in the used Kornia library, considerable familiarity in the community, and good performance across many benchmarks. (Mishchuk et al., 2017; Jin et al., 2020)

4.7 Particle resampling

No other resampling algorithm than stratified resampling was tried due to this being an overlooked factor in the implementation, testing, and tuning phases. According to results presented by Elfring et al. (2021), this may have a significant impact on the performance of the PF, especially if the particle weights are not sufficiently separated.

If a particle's estimated altitude was outside the range of 30 – 250 m, or if the weight measurement failed for the particle, the particle was drawn randomly in the search area.

Normally distributed, zero-mean noise with zero covariance was added to the resampled particles to all 4 estimated dimensions. The standard deviation of the noise for each dimension is discussed in section 5.1.

5 Results and analysis

This chapter begins with detailing the parameter choices used for evaluation and the process of obtaining the results, then it proceeds to explain what additional steps were performed for analysis, and finally, it presents the findings in various levels of detail. However, since there were in total 72 individual algorithm evaluations performed (6 videos, 6 evaluations, 2 solutions), most visualisations of the results are attached in Appendices 1 and 2 as described later.

5.1 Parameter setup

The selection of parameters was manually explored on the “Calibration” video and Google Maps satellite imagery until suitable parameters were found. The parameter calibration was first performed extensively for the BLS, then briefly for the PLS with only 5 different configurations. The selected PLS configuration was one that performed best among those with a shorter execution time than the final BLS configuration. The manual tuning for BLS was done until the mean localization error was below 30 m and by visually confirming the presence of a particle cluster at the GT throughout the whole trajectory.

For all evaluations, the PF was always initialized using the normal distribution with a standard deviation of 20 m in the horizontal position and 10 m in the vertical position around the position of $(0,0,135)$ in the local CRS. The UAV’s true position at the beginning of the video recordings was up to 11 m away from the selected initialization location. The heading of the particles was initialized uniformly through the whole range of $[0,2\pi)$ radians.

Since no camera calibration was performed for the evaluation dataset, a pinhole camera model was used with the camera focal length equivalent of 24 mm on the 35 mm sensor format, as stated in the UAV’s product information (*Phantom 4 Pro - Product Information - DJI*, no date). The principal point was chosen to be the centre of the image and no distortion rectification was applied. The image size used in BLS evaluation was 512 px wide and 270 px high which preserved enough detail in the image and had a short enough runtime for practical experiments, lower resolutions were also tried. The image resolution is not a significant factor in the execution time of the PLS since the number of processed patches remains constant. Intuitively, the method should benefit from higher resolution and therefore, the image size for the PLS evaluation was increased to 1024 px wide and 540 px high.

During the tuning of the BLS, many combinations of parameters and weight normalization functions were explored. The typical problems were very poor and

slow convergence, rapid divergence, and convergence to local optima from which the PF was not able to recover. Noise levels between 1 m to 40 m in standard deviation in position and 1° – 10° in heading were considered. The number of particles ranged from 500 to 2000 and measurement (weighting) periods ranged between every 5 to 200 frames. The final parameters for the BLS were 20 m standard deviation in every positional coordinate and 4° standard deviation in heading, 1000 particles, and measurements were done every 50 frames (about 2 seconds in the footage). The processing takes about 4 times longer than real-time.

Only some configurations were explored for the PLS not to introduce much bias. The selected configuration is 10 m in standard deviation in the horizontal position, and 5 m in altitude. Only 200 particles are used. The remainder of the parameters remains the same as in the BLS. The processing takes about 2 times longer than real-time, i.e., half the time of the BLS.

Each of the 6 videos (including the “Calibration” video) was evaluated 6 times on both solutions with the NLS orthographic imagery dataset as the reference dataset.

5.2 Analytic choices

According to the study design, the results should be compared against each other and against the ground truth. A representative value for the estimated pose must therefore be found in each iteration of the PF, for example, the mode, mean, or the particle with the highest weight. The weighted mean of particles is used similarly to the works of Jurevičius et al. (2019) and Kinnari et al. (2021).

During the tuning phase, it was observed that the particles often split into 2 or more clusters. To avoid representative values that are outside of the clusters, the clusters are identified using the K-means clustering algorithm (Hartigan and Wong, 1979), and the weighted mean is reported for each cluster individually.

Note that if the weighted version of K-means clustering is used, the cluster centroid found by the algorithm is the sought weighted mean. This approach is suggested by Yozevitch et al. (2017) for particle filters with multimodal distributions.

The number of clusters fluctuates and needs to be estimated for every PF iteration. The number of clusters that maximizes the silhouette score is used (Rousseeuw, 1987). However, the silhouette score is defined only for two or more clusters, to overcome this limitation, the silhouette score for a single cluster is set to 0.5 regardless of the cluster's characteristics.

The final issue to be addressed for analysis is the synchronization of the GNSS ground truth data with the video footage. The GT recordings began before and ended after the video recording by an unknown length of time. Furthermore, there appears to be variability in the video frame or GT frequency or both. To resolve the problem, the GT positions for every video were interpolated to a higher frequency and the start offset, the end offset, and the relative speed coefficient of the GT were found. The GT was interpolated to a 10 Hz frequency to attain a position granularity of 1 m or better (the maximum flight speed was 10 m/s). The start offset, the end offset, and the speed coefficient were found so that they together minimize the error in the distance between the proposed GT alignment and the closest PF centroid. For each video, the time alignment with the lowest error was selected as the GT for all the evaluations on that video. No GT for the heading was determined.

5.3 Overall results

To gain an intuitive understanding of the accuracy and general performance of the solutions, Figure 9 presents the distances (errors) between the estimated centroids and their respective GT positions, i.e., the Euclidean norm of the difference between the GT and centroid positions. If there were more centroids estimated in an iteration, only the closest one to the GT is considered. In practice, solutions that would reliably choose the correct centroid likely exist.

For example, in the case of this work, selecting the cluster with the highest number of particles would coincide with the desired cluster most of the time. Moreover, the use of IMU measurements in real-world applications should further decrease ambiguity. The data is presented per video per solution, such that all the errors of the 6 evaluations in each category are merged and sorted by magnitude. Furthermore, the indices of the sorted errors are normalized to $[0,1]$ range, thus creating a ranking plot where the error value maps directly to its quantile and vice versa. This representation makes it straightforward to compare the general performance of the solutions on the evaluated videos.

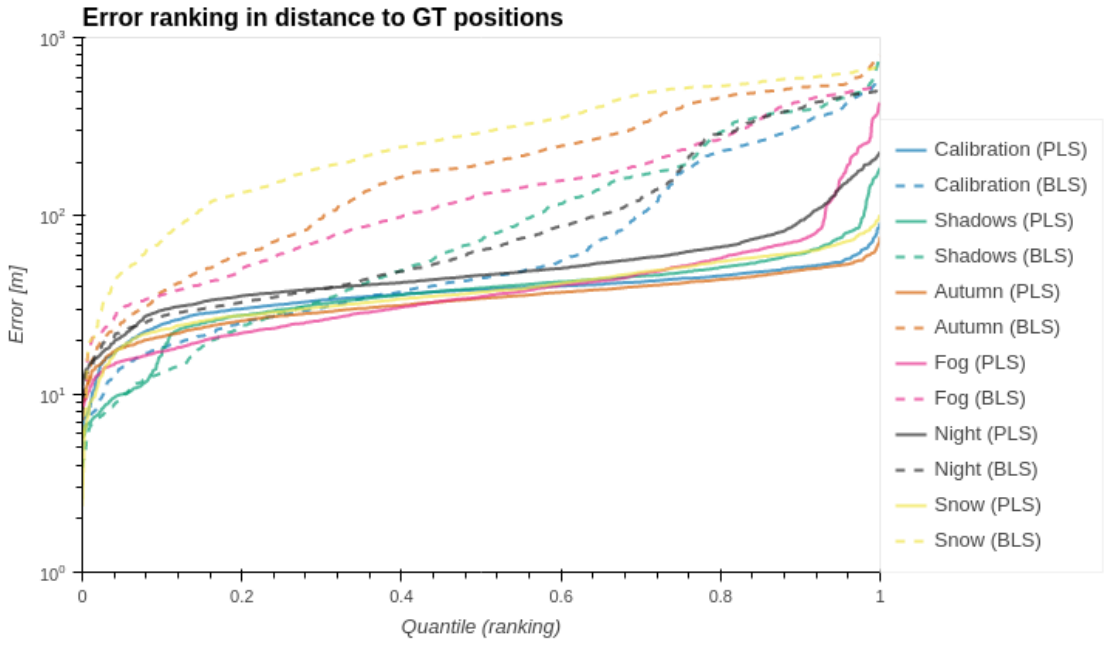


Figure 9. Ranking of distances (errors) between estimated centroids and their corresponding GT position.

Notice, that the errors in Figure 9 are shown in the logarithmic scale and that the ideal performance would be a horizontal line at the bottom of the plot. Figure 9 that the PLS tends to perform better than BLS overall. According to the graph, the “Calibration” and “Shadows” videos contain parts where the BLS has a lower error. However, it is clearly less than half of the samples and the rest of the samples have a rapidly increasing error. Conversely, closer inspection of the data shows that BLS tends to be better in estimating the altitude than the

PLS as shown in Figure 10. Similarly, Figure 11 shows only the horizontal components of the distances to the GT.

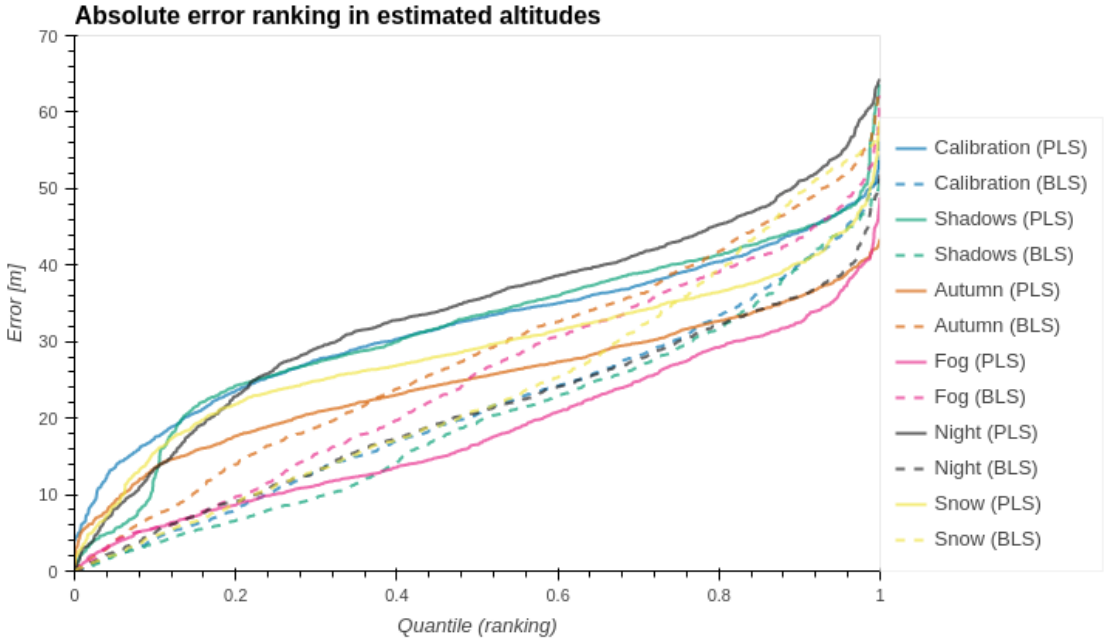


Figure 10. Ranking of absolute errors between estimated and GT altitudes.

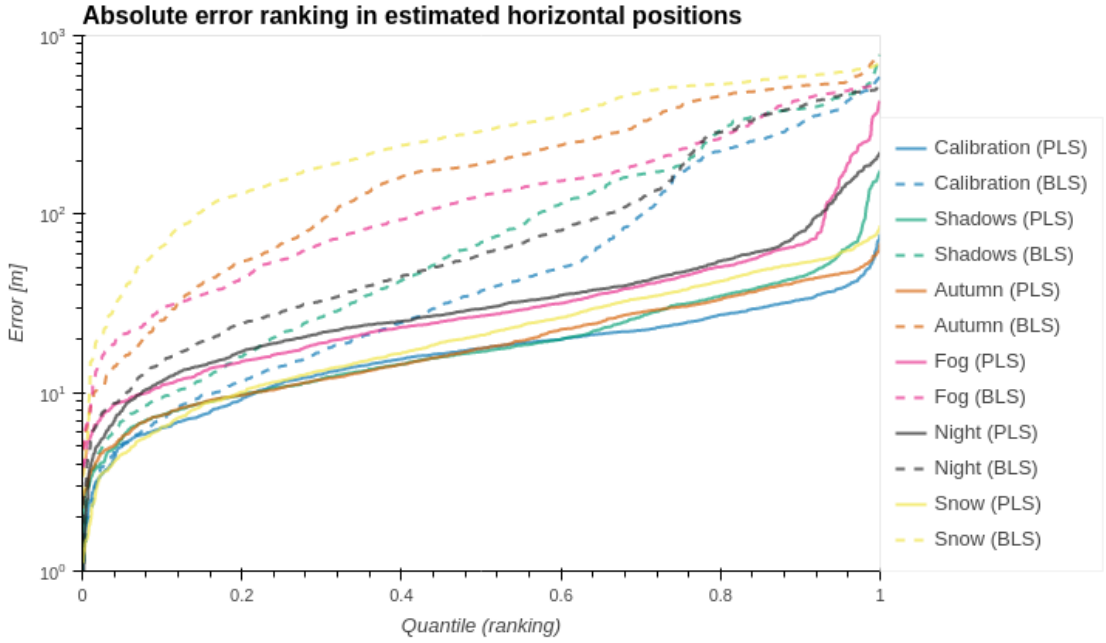


Figure 11. Ranking of horizontal distances between estimated and GT positions.

Figure 10 and Figure 11 in conjunction with Figure 9 show that the error in estimated altitude is a major component of the 3D distance between the estimated and GT positions. When considering only the horizontal error, the PLS compares even more favourably to the BLS.

5.4 Detailed results

Results presented in the previous section ignore the time dimension of the estimation and therefore lack insights necessary for a deeper understanding of the performance and characteristics of the solutions. This section introduces a detailed analysis of the results, including the time dimension, and highlights or explains the findings.

Figure 12 shows a sample of the most evaluation-specific visualization used in this work. Plots for all 72 evaluations are included in Appendix 2. The figure consists of three subplots:

1. The first subplot shows the estimated clusters in time and plane. Each cluster is displayed as a disk over its centroid with its diameter proportional to the standard deviation of distances between the centroid and each particle within the cluster. The weights of the particles are considered when calculating the standard deviation like in the calculation of the centroid.
   The colour of a disk corresponds to its 3D distance from its corresponding GT position up to 100 m. Errors beyond 100 m have their colour clipped to 100 m, this improves the discriminability within non-diverged clusters and simplifies comparisons between different evaluations. In addition, some transparency is added to the disks to make it possible to view overlapping disks, this may compromise the colour to some extent.
   To introduce time associations in the subplot, each disk is connected by a line segment to its GT point; with the PF measurements performed every 50 frames, this translates to the GT points being about 2 seconds

apart. Furthermore, the line segments indicate how many clusters are associated with a given GT location. The GT path is also displayed for reference and better contrast of the line segments.
Finally, there are three concentric disks overlayed on top of each other, each representing a reference size for the intra-cluster standard deviations. There are two different scales used in the plots, the scale used in the plots of BLS evaluations is half the scale used in the PLS plots since the standard deviations in the BLS results tend to be much higher. The coordinate system used in the subplot is defined by the local CRS described earlier, and the axes of the subplot are fixed across all evaluations, focusing on the main area of interest.

2. Only limited information on altitude could be obtained from the first subplot and only with great difficulty (using the distance represented by the colour and finding the planar distance would yield information on the absolute error in altitude). Therefore, the second subplot explicitly shows the altitude components of the estimated centroids. The colour of the points is defined in the same fashion as in the first subplot. Unlike in the first subplot, the size is kept constant.
   The time dimension is shown on the horizontal axis, naturally, using the PF measurement iterations as time steps. Moreover, there are 7 vertical lines in the plot indicating the 8 segments of the path. The GT altitude is also shown in the subplot, but its accuracy is unknown, and it is relative to the UAV's take-off position, not relative to the terrain like the estimated altitude. Also, note that the vertical axis is floating and scaled according to each evaluation plot.

3. The third subplot represents the estimated heading. The heading is represented in radians in the anti-clockwise direction with 0 pointing upwards (North). The same colouring, sizing, and horizontal axis definitions as in the second subplot apply. No GT heading is available.

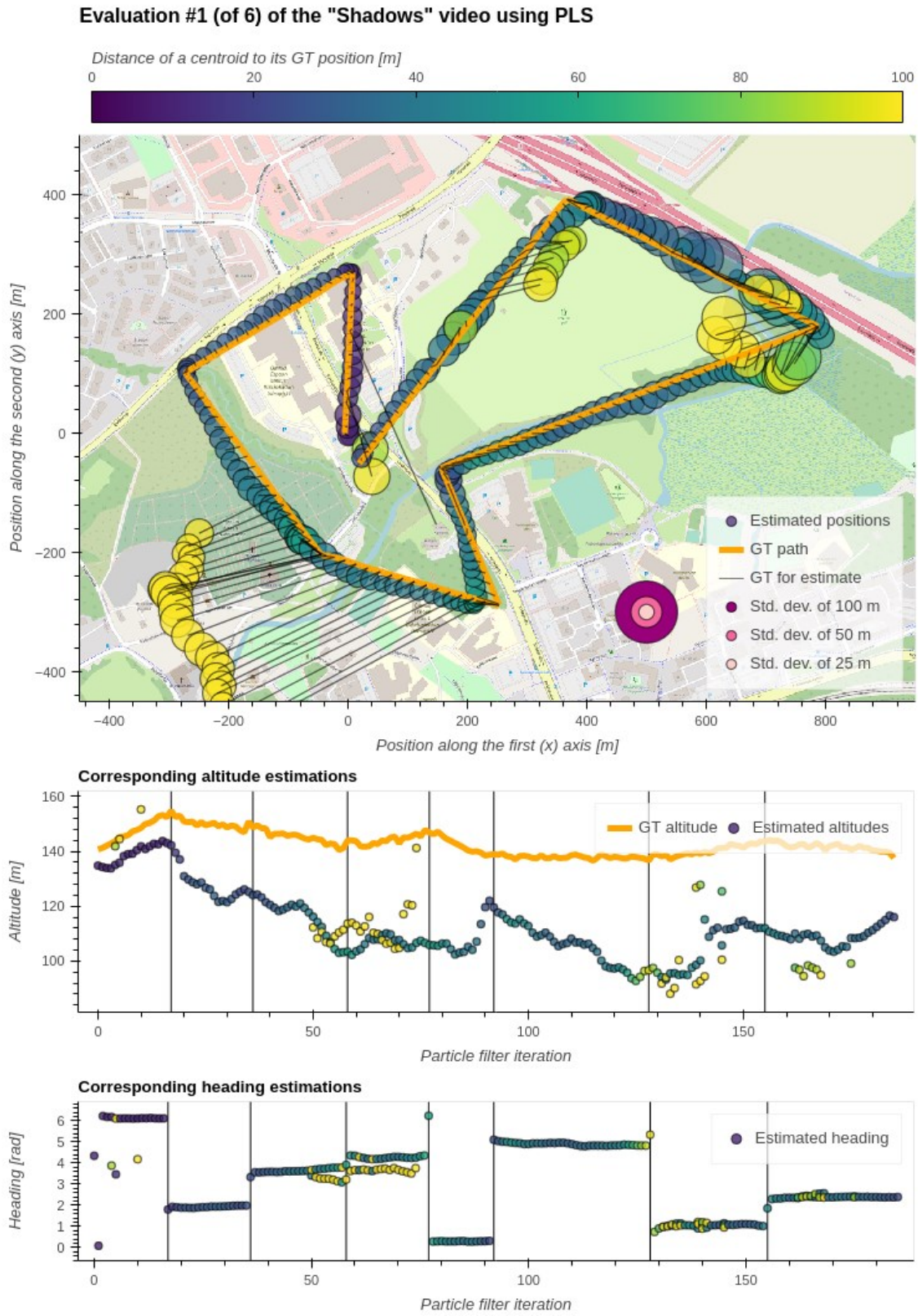


Figure 12. Example of an individual evaluation.

As the title embedded in Figure 12 reads, the figure communicates the details of one of the PLS evaluations on the “Shadows” video. The figure shows that throughout the whole video, there are clusters close to the GT. The performance was best on the first segment of the track (see Figure 3), it decreases somewhat on the second segment due to a larger error in estimated altitude. In the later part, two clusters are tracked simultaneously over the cemetery and beyond, the stray cluster later dissipates when the turn to the fifth segment is encountered. The performance is poorest at the end of the sixth segment over the flood zone. The error quickly decreases when reaching the highway, but along the homogenous-looking section of the highway, the standard deviation increases which could be interpreted as an increase in uncertainty. The final segment displays similar behaviour as the previous segment. Over the whole course of the evaluation, the difference in estimated altitude seems to have the highest impact on error. In addition, observe that the vertical lines indicating different GT path segments do coincide with the changes in the estimated heading.

To demonstrate the need for analysis akin to Figure 12, the first subplot in Figure 12 can be compared with Figure 13. Figure 13 shows the bare particles of the PF without further analysis. Each particle is shown at its planar position as a disk, with size proportional to the particle’s weight and colour depending on the PF iteration. Due to the large number of data points, it is challenging to recognize individual iterations and understand the inherent estimates of the PF. With the higher number of particles used in the BLS evaluations, this would become even more difficult. Nevertheless, some associations between the figures are visible, for example, most of the particles keep close to the GT path, and the set of clusters in the lower-left part of the plot.

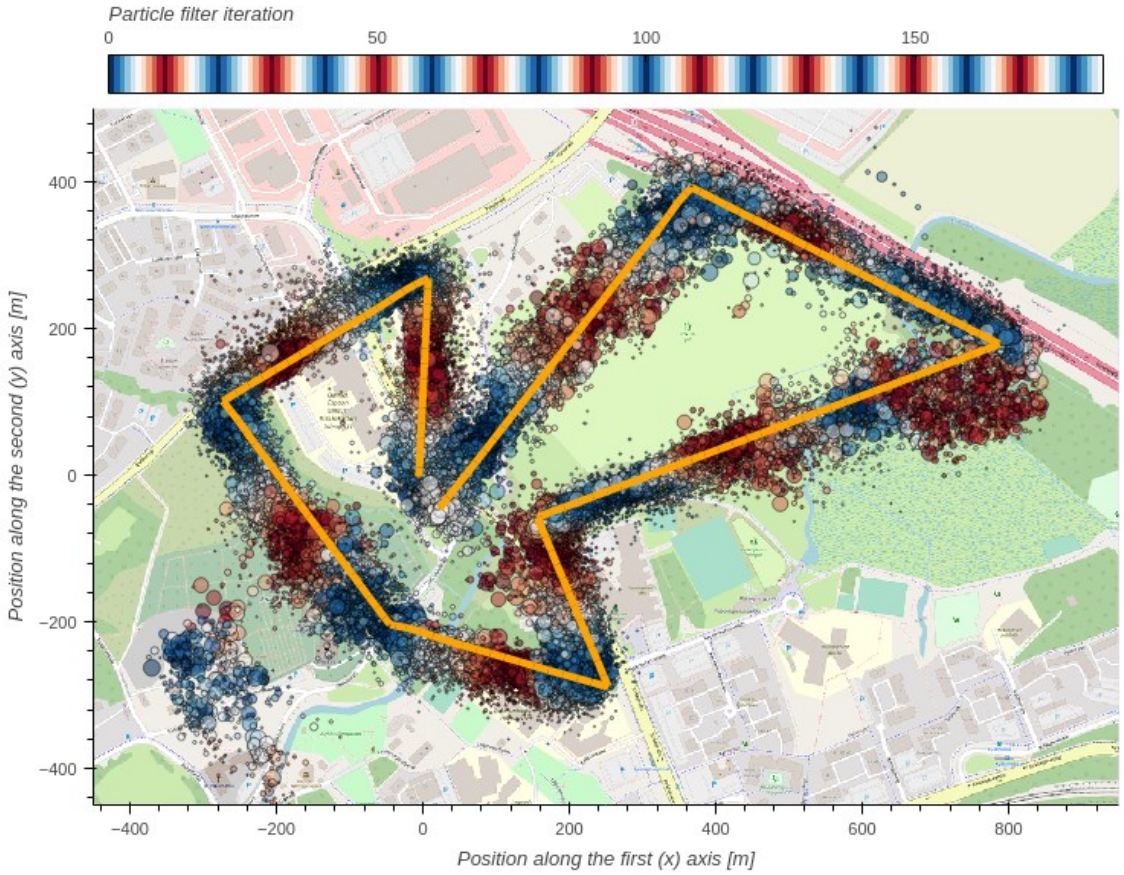


Figure 13. A cloud of all particles from all PF measurement iterations corresponding to the evaluation in Figure 12.

Typically, the larger the collection of results, the better and more general conclusions can be drawn. For this reason, all 12 test evaluations per video are combined and visualized together. To save space the visualizations can be found in Appendix 1 since there were 6 different videos used. Figure 14 serves as an example of one of the visualizations, covering the evaluations performed on the “Shadows” video. In these combined visualizations, shades of blue always depict information PLS evaluations whereas shades of red pertain to BLS evaluations. Like in Figure 12, the visualization consists of three subplots, the figures share a resemblance, but the data is presented in different ways:

1. Similarly to Figure 12, the first subplot in Figure 14 shows the estimated centroids in space. Unlike in the former figure, only the closest estimated clusters to the GT position are shown, the justification for this simplification is explained in the previous section. In addition, displaying

all clusters could result in very cluttered visualizations. The time aspect is instead indicated by connecting consecutive centroids with a line segment. For improved clarity, only simple points with a constant size indicate the centroids. This representation aids in observing and identifying the typical behaviours of the solutions, such as the problematic or well-performing areas, and the degree of their consistency.

2. The second subplot visualizes the absolute errors in estimated altitudes. The time axis is represented as in Figure 12. On the other hand, the GT altitude is not displayed since in an error plot it coincides with the horizontal axis. Different evaluations of the same solution are not distinguished. The absolute error in the altitude is represented by a point for each evaluation and iteration. An envelope of the errors is shown for visual guidance in finding all the associated markers. Lastly, the mean error for each iteration is plotted as a curve, to clarify, it does not represent the rolling mean.

3. The last subplot shows the 3D distance (error) between the estimated centroids and the GT positions. It maintains the same form of presentation as the second subplot, however, due to a larger range of errors, a logarithmic scale is used for the vertical axis.

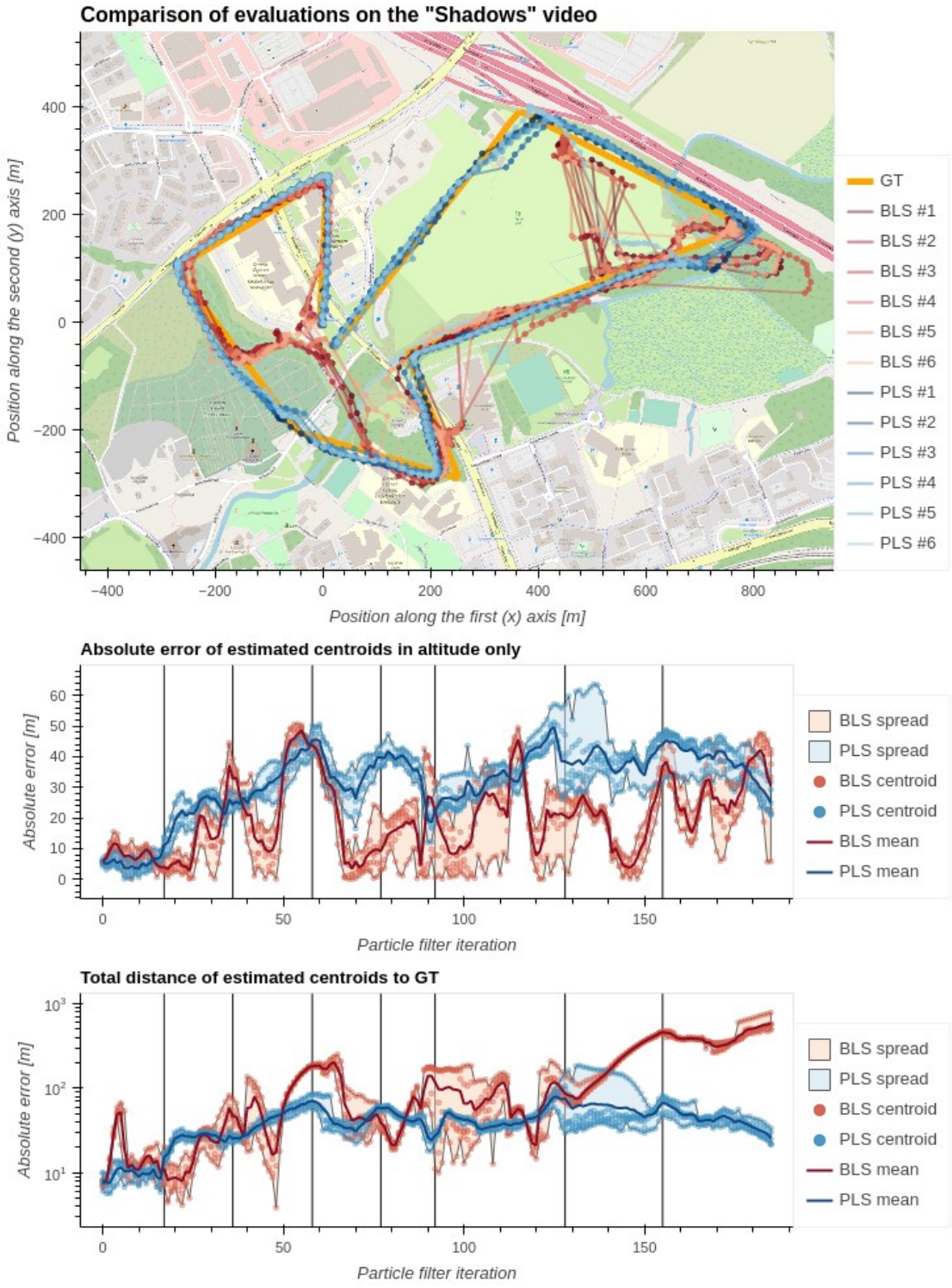


Figure 14. Example of a combined visualization of all evaluations performed on a single video.

By studying Figure 14, it can be concluded that the performance of the PLS is coherent on the “Shadows” footage. All the estimates are close to the GT path,

suggesting a low error in the horizontal position estimation. However, when paying attention to the error subplots, it is apparent that the altitude estimates constitute a large proportion of the distance to the GT. The BLS evaluation also has consistent performance but to a lesser extent. However, this consistency is also evident in the noticeable divergent sections. Initially, both solutions seem to agree on the estimated position, though this changes when reaching the cemetery area in the third GT segment, and then the consensus restores again upon reaching an area with visible buildings in the fourth GT segment. At the end of the sixth segment, the BLS estimates veer off to the right of the trajectory and then to the golf course, the pose is never recovered afterwards.

Considering the rest of the visualizations in the appendices, similar characteristics can be observed about the general PLS performance. Its performance tends to be consistent and most of the time being near its GT. Only two videos (“Fog” and “Night”) contained evaluations that did not have their final estimates converged near the GT. The “Night” footage contains only two converged estimates, the other four diverged in the last segment which seems to be the most challenging part together with the end of the sixth segment. The evaluations on the “Fog” video consistently diverge in the middle of the last segment but otherwise, the overall convergence and errors are good. Upon manual inspection, it was found that visual odometry is defective at the point of divergence and causes all the particles to be reinitialized to a random state.

Regarding the BLS results, under no evaluation was the algorithm able to reliably track the whole trajectory and only in some cases, it happened to reach the final pose by chance. No evaluations got past the seventh segment reliably. Like in the case shown in Figure 14, the cemetery and the area beyond the middle of the sixth segment prove challenging in general. In the case of the “Snow” video, BLS has not been able to converge at all.

5.5 Results summary

The results have been visualized in several different ways in the previous sections and in the appendices. To compactly summarize the results, Table 1 tabulates basic descriptive statistics of the results. The statistics are listed per method and per video with the addition of an overall entry. The shown fields are calculated from the distances between the GT positions and their closest estimated centroids on all the relevant evaluations. That is, the “Mean” column represents the mean distance between the GT and the best estimate. The column is averaged across all iterations and evaluations performed on the given video. Similarly, the “Std. dev.” column denotes the standard deviation in the distances. The last five columns represent the ranking of the distances (see Figure 9 and its surrounding paragraphs for more detail), i.e., the minimum, first quartile, median, third quartile, and the maximum respectively. The units are metres in all numeric columns.

Table 1. Statistics of distance to GT for all evaluations.

Video	**Method**	**Mean**	**Std. dev.**	**Min.**	**25%**	**50%**	**75%**	**Max.**
Calibration	PLS	38.27	11.41	6.06	31.79	38.47	44.42	91.57
	BLS	111.82	131.24	4.86	27.97	44.71	172.99	589.62
Shadows	PLS	41.43	22.80	5.65	29.76	39.36	48.15	185.55
	BLS	143.22	153.14	3.86	28.52	72.97	190.39	783.35
Autumn	PLS	34.84	10.77	7.97	27.47	34.41	41.83	76.27
	BLS	243.77	185.65	5.59	74.90	194.16	402.16	789.77
Fog	PLS	49.62	54.52	8.42	23.91	35.20	52.12	430.75
	BLS	171.90	142.08	9.42	61.29	131.62	233.53	529.69
Night	PLS	56.52	36.06	8.95	37.40	46.41	61.86	226.90
	BLS	136.91	146.40	7.53	35.57	64.26	183.51	528.22
Snow	PLS	40.64	16.33	2.35	28.97	37.35	51.35	100.23
	BLS	323.94	192.26	5.95	156.34	292.56	518.98	717.96

Video	Method	Mean	Std. dev.	Min.	25%	50%	75%	Max.
Overall	PLS	43.43	30.30	2.35	29.58	38.46	48.77	430.75
	BLS	189.09	176.37	3.86	42.68	121.72	297.03	789.77

Table 1 shows that PLS indeed performs substantially better than BLS as suggested by the visualizations described earlier. When considering the median error, the BLS performs closely only on the “Calibration” video. This is likely due to both, the parameter tuning having been performed on the video, and the video being the most similar in appearance to the used reference imagery in the PF measurement step. The mean error is strongly affected by the divergent estimates, naturally, that indicates poor overall performance but bears little information on the accuracy of well-performing parts. Furthermore, horizontal, or vertical error may be more important in certain applications. Hence, Table 2 reveals the mean distance in metres in the horizontal and vertical directions for, both, the whole video, but also for the first half of the video where both solutions appear to perform better. The first half of the path ends approximately at the end of the fifth segment before the UAV rotates to follow the brook in the sixth segment.

Table 2. Summary of mean errors.

Video	Method	Mean distance in direction and video part			
		Horizontal, whole video	Vertical, whole video	Horizontal, first half	Vertical, first half
Calibration	PLS	19.03	31.81	15.23	31.72
	BLS	104.40	21.28	23.17	19.63
Shadows	PLS	24.00	31.63	20.60	26.23
	BLS	139.24	19.92	52.58	17.30

Video	Method	Mean distance in direction and video part			
		Horizontal, whole video	**Vertical, whole video**	**Horizontal, first half**	**Vertical, first half**
Autumn	PLS	21.50	24.89	21.44	22.07
	BLS	239.76	27.97	142.58	26.99
Fog	PLS	42.86	18.48	36.12	21.29
	BLS	167.77	25.08	129.42	25.84
Night	PLS	41.11	34.08	27.13	24.99
	BLS	131.77	20.82	53.44	15.37
Snow	PLS	25.98	28.68	32.06	30.27
	BLS	321.57	23.86	336.06	19.68
Overall	PLS	28.92	28.29	25.32	26.13
	BLS	184.57	23.18	123.07	20.82

According to the data in Table 2, PLS has a lower mean error in horizontal estimates in all cases. On the contrary, BLS has a slightly lower overall mean absolute error in vertical estimates. Additionally, when considering the 3D mean error only for the first half of the flight path, the PLS performs worse than BLS only on the “Calibration” video with the means being 35.87 m for PLS and 33.66 m for BLS. Note that the values listed for horizontal and vertical mean errors cannot be simply averaged with the quadratic mean to get a true mean distance. This is due to the horizontal mean being a nonlinear average of multiple values. However, since the mean distance for the first half of the trajectory was omitted for brevity, the quadratic mean is a still fair estimate if the reader is interested.

6 Conclusion

A robust method for scoring image alignment in the context of aerial localization has been proposed in this thesis. An example localization solution employing the method has been implemented and evaluated on a diverse dataset. Another solution employing a commonly used, established method was also implemented and evaluated for comparison. However, a detailed analysis of the image alignment score analogous to one presented by Kinnari et al. (2022) was not performed since the method can be implemented in numerous different forms and a due analysis would be too extensive.

It was shown that the proposed approach is feasible for UAV visual self-localization with modest assumptions and without domain-specific training or engineering. The performance was consistent across the selected dataset and compared very favourably to the benchmark solution, especially in more challenging visual conditions.

Despite the care taken to avoid the introduction of bias and systematic errors, several potentially confounding factors remain, for example, an unconventional, cluster-based analysis of the results, possible parameter overfitting, and the ground truth data synchronization. But also, poor quality of the ground truth data, its alignment with the results, poor motion estimation, no camera calibration, etc.

Adaptations to the original plan of evaluation had to be made due to the empirical discovery of synthetic data not being representative, i.e., the performance of the benchmark solution did not translate from synthetic data to real video footage. Therefore, further modifications to the benchmark solution had to be performed. But also because of integration issues and too broad scope.

It is straightforward to extend the solution into incorporating other sensory data and estimating more parameters. Additional available external information such as the visible surface topology or multiple image reference sources are also

easy to integrate in derivatives of the solution. There are many possible areas for future studies and improvements of the overall proposed solution, including but not limited to optimizing the reference patch selection, patch descriptor, reference database, the strategy and parameters of image alignment scoring, and the representative state selection policy.

References

Alkendi, Y., Seneviratne, L. and Zweiri, Y. (2021) 'State of the Art in Vision-Based Localization Techniques for Autonomous Navigation Systems', *IEEE Access*, 9, pp. 76847–76874. doi: 10.1109/ACCESS.2021.3082778.

Belmonte, L. M., Morales, R. and Fernández-Caballero, A. (2019) 'Computer Vision in Autonomous Unmanned Aerial Vehicles—A Systematic Mapping Study', *Applied Sciences 2019, Vol. 9, Page 3196*, 9(15), p. 3196. doi: 10.3390/APP9153196.

Couturier, A. and Akhloufi, M. A. (2021) 'A review on absolute visual localization for UAV', *Robotics and Autonomous Systems*, 135, p. 103666. doi: 10.1016/J.ROBOT.2020.103666.

Elfring, J., Torta, E. and van de Molengraft, R. (2021) 'Particle Filters: A Hands-On Tutorial', *Sensors 2021, Vol. 21, Page 438*, 21(2), p. 438. doi: 10.3390/S21020438.

Fischler, M. A. and Bolles, R. C. (1981) 'Random sample consensus', *Communications of the ACM*, 24(6), pp. 381–395. doi: 10.1145/358669.358692.

Forster, C., Pizzoli, M. and Scaramuzza, D. (2014) 'SVO: Fast semi-direct monocular visual odometry', *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 15–22. doi: 10.1109/ICRA.2014.6906584.

GDAL/OGR contributors (2022) '{GDAL/OGR} Geospatial Data Abstraction software Library'. doi: 10.5281/zenodo.5884351.

Gillies, S. et al. (2013) 'Rasterio: geospatial raster I/O for {Python} programmers'. Available at: https://github.com/rasterio/rasterio.

Guttman, A. (1984) 'R-trees: A dynamic index structure for spatial searching', *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 47–57. doi: 10.1145/602259.602266.

Hartigan, J. A. and Wong, M. A. (1979) 'Algorithm AS 136: A K-Means Clustering Algorithm', *Applied Statistics*, 28(1), p. 100. doi: 10.2307/2346830.

Jin, Y. *et al.* (2020) 'Image Matching Across Wide Baselines: From Paper to Practice', *International Journal of Computer Vision 2020 129:2*, 129(2), pp. 517–547. doi: 10.1007/S11263-020-01385-0.

Jurevičius, R., Marcinkevičius, V. and Šeibokas, J. (2019) 'Robust GNSS Denied Localization for UAV Using Particle Filter and Visual Odometry'. doi: 10.1007/s00138-019-01046-4.

Kalman, R. E. (1960) ‘A New Approach to Linear Filtering and Prediction Problems’, *Journal of Basic Engineering*, 82(1), pp. 35–45. doi: 10.1115/1.3662552.

Kinnari, J., Verdoja, F. and Kyrki, V. (2021) ‘GNSS-denied geolocalization of UAVs by visual matching of onboard camera images with orthophotos’, *2021 20th International Conference on Advanced Robotics, ICAR 2021*, pp. 555–562. doi: 10.48550/arxiv.2103.14381.

Kinnari, J., Verdoja, F. and Kyrki, V. (2022) ‘Season-Invariant GNSS-Denied Visual Localization for UAVs’, *IEEE Robotics and Automation Letters*, 7(4), pp. 10232–10239. doi: 10.1109/LRA.2022.3191038.

Lowe, D. G. (2004) ‘Distinctive Image Features from Scale-Invariant Keypoints’, *International Journal of Computer Vision 2004 60:2*, 60(2), pp. 91–110. doi: 10.1023/B:VISI.0000029664.99615.94.

Lu, Y. *et al.* (2018) ‘A survey on vision-based UAV navigation’, *Geo-Spatial Information Science*, 21(1), pp. 21–32. doi: 10.1080/10095020.2017.1420509.

Mantelli, M. *et al.* (2019) ‘A novel measurement model based on abBRIEF for global localization of a UAV over satellite images’, *Robotics and Autonomous Systems*, 112, pp. 304–319. doi: 10.1016/J.ROBOT.2018.12.006.

MarketsandMarkets (2022) *Unmanned Aerial Vehicle (UAV) Market Share, Size, Trends - [2022-2027]*. Available at: https://www.marketsandmarkets.com/Market-Reports/unmanned-aerial-vehicles-uav-market-662.html (Accessed: 18 October 2022).

Mcgee, L. A., Schmidt, S. F. and Schmidt, S. F. (1985) *Discovery of the Kalman filter as a practical tool for aerospace and industry*.

Mishchuk, A. *et al.* (2017) ‘Working hard to know your neighbor’s margins: Local descriptor learning loss’, *Advances in Neural Information Processing Systems*, 2017-December, pp. 4827–4838. doi: 10.48550/arxiv.1705.10872.

Moritz, H. (1980) ‘Geodetic reference system 1980’, *Bulletin géodésique 1980 54:3*, 54(3), pp. 395–405. doi: 10.1007/BF02521480.

NLS (2020) ‘National Land Survey of Finland, NLS orthophotos’. Available at: https://www.maanmittauslaitos.fi/en/maps-and-spatial-data/expert-users/product-descriptions/orthophotos (Accessed: 14 September 2020).

Obdržálek and Matas, J. (2002) ‘Local affine frames for image retrieval’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2383, pp. 318–327. doi: 10.1007/3-540-45479-9_34/COVER.

Phantom 4 Pro - Product Information - DJI (no date). Available at: https://www.dji.com/fi/phantom-4-pro/info#specs (Accessed: 20 October 2022).

Revaud, J. *et al.* (2019) ‘R2D2: Reliable and Repeatable Detector and Descriptor’, *Advances in Neural Information Processing Systems*, 32. Available at: https://github.com/naver/r2d2. (Accessed: 19 October 2022).

Riba, E. *et al.* (2020) ‘Kornia: An open source differentiable computer vision library for PyTorch’, *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, pp. 3663–3672. doi: 10.1109/WACV45572.2020.9093363.

Roma, N., Santos-Victor, J. and Tomé, J. (2002) ‘A Comparative Analysis of Cross-Correlation Matching Algorithms Using a Pyramidal Resolution Approach’, pp. 117–142. doi: 10.1142/9789812777423_0006.

Rousseeuw, P. J. (1987) ‘Silhouettes: A graphical aid to the interpretation and validation of cluster analysis’, *Journal of Computational and Applied Mathematics*, 20(C), pp. 53–65. doi: 10.1016/0377-0427(87)90125-7.

Snyder, J. P. (1993) *Flattening the earth : two thousand years of map projections*. University of Chicago Press.

Szeliski, R. (2022) *Computer Vision - Algorithms and Applications, Second Edition*. 2nd edn. Springer (Texts in Computer Science). doi: 10.1007/978-3-030-34372-9.

Thrun, S., Wolfram, B. and Fox, D. (2005) *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, *The Mit Press*. The Mit Press.

Wel, C. van der *et al.* (2022) ‘pygeos/pygeos: 0.13’. doi: 10.5281/ZENODO.7023576.

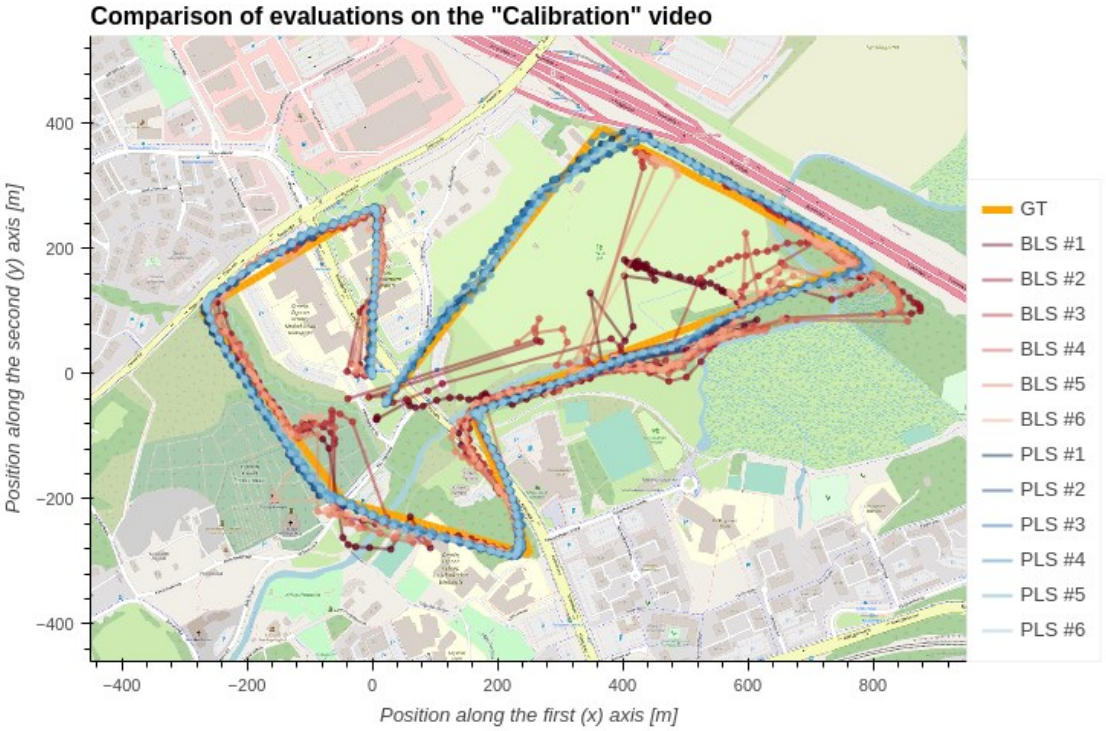
Yozevitch, R. *et al.* (2017) ‘Advanced Particle Filter Methods’, *Heuristics and Hyper-Heuristics - Principles and Applications*. doi: 10.5772/INTECHOPEN.69236.

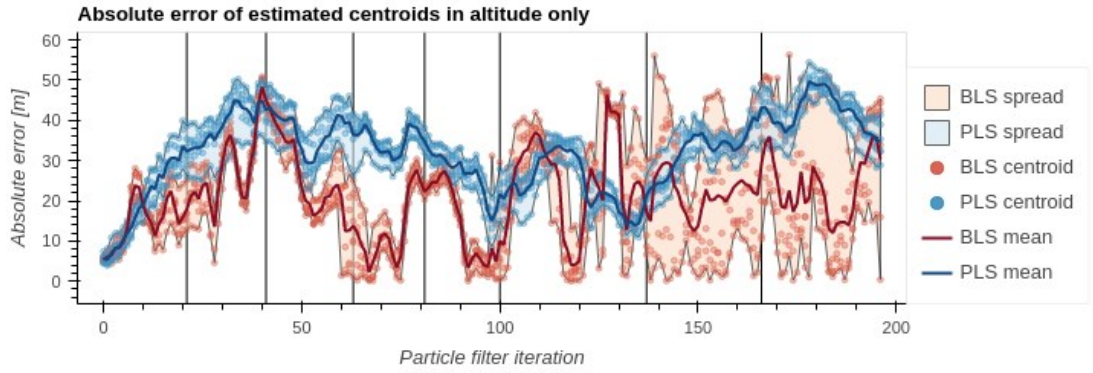
Appendix 1: Grouped results

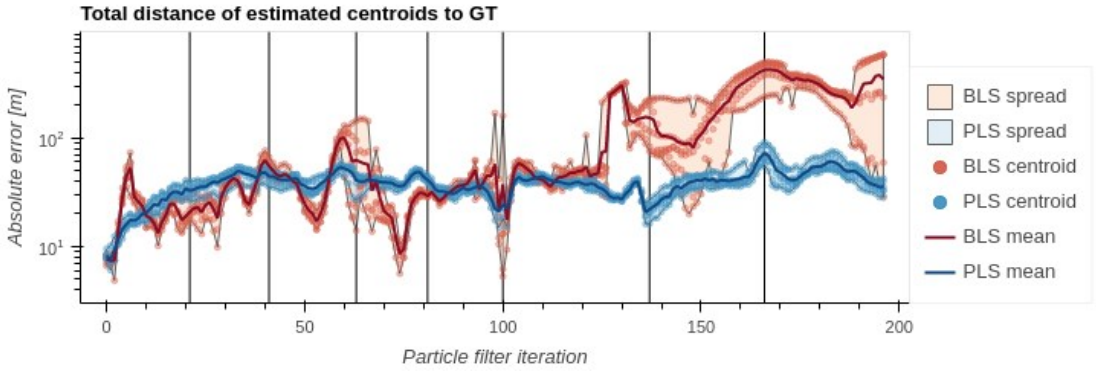
The following 6 figures represent all evaluations performed on each evaluation video. The order of the video evaluations is as follows:

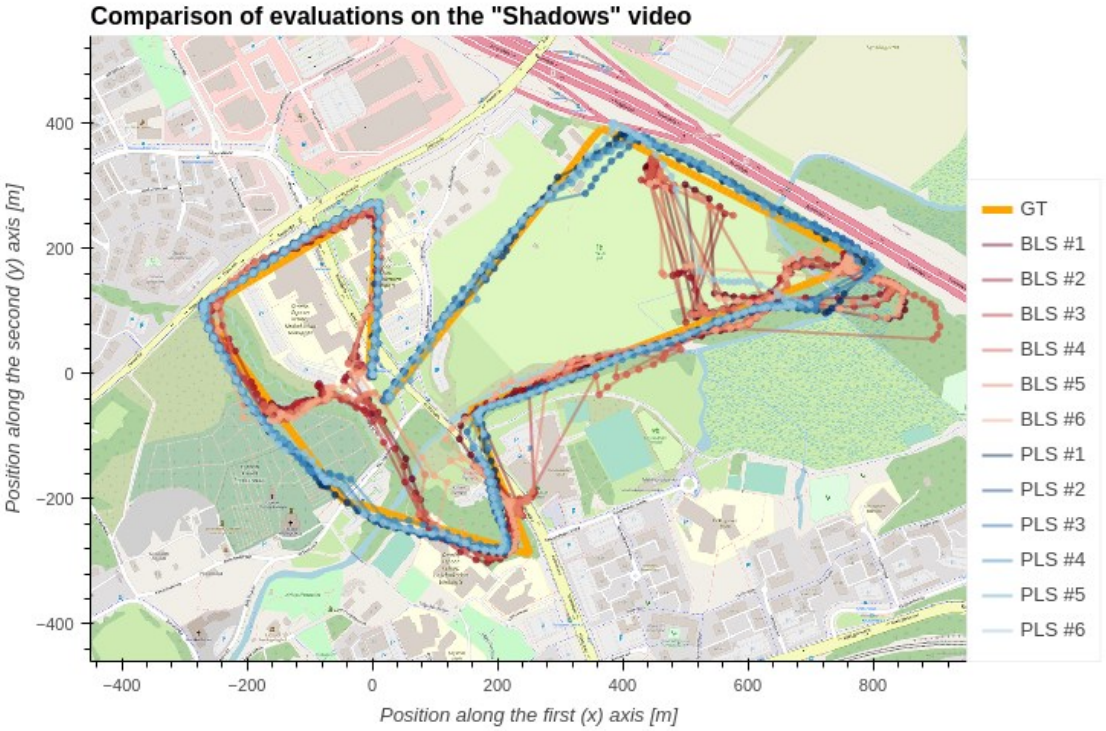
- “Calibration”
- “Shadows”
- “Autumn”
- “Fog”
- “Night”
- “Snow.”

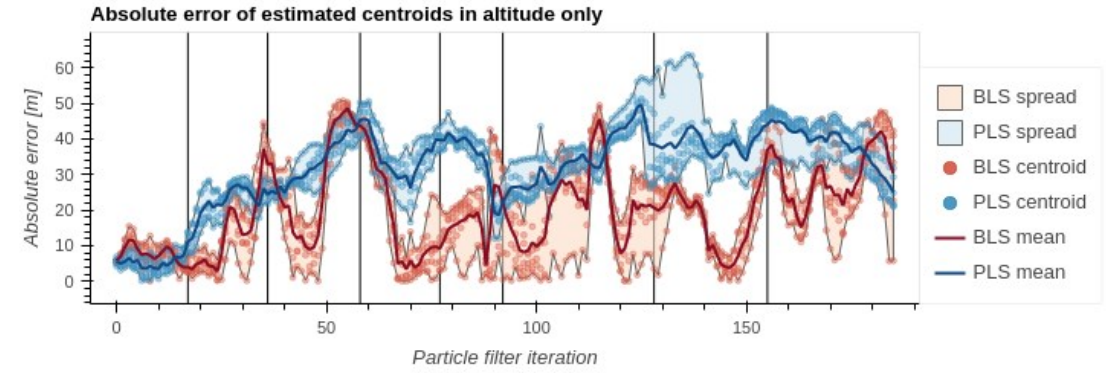
Note that each figure refers to the corresponding evaluation video in its title. The detailed description of the figures can be found in section 5.4.

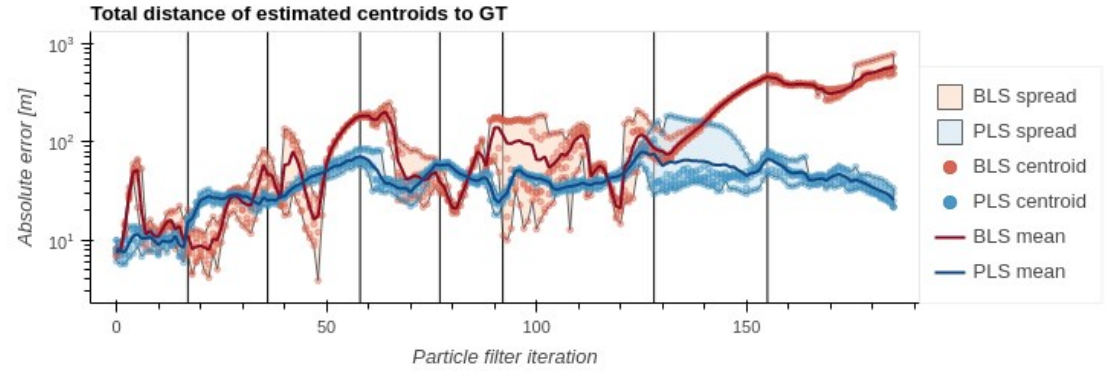

Comparison of evaluations on the "Calibration" video
Position along the second (y) axis [m]
Position along the first (x) axis [m]
GT
BLS #1
BLS #2
BLS #3
BLS #4
BLS #5
BLS #6
PLS #1
PLS #2
PLS #3
PLS #4
PLS #5
PLS #6

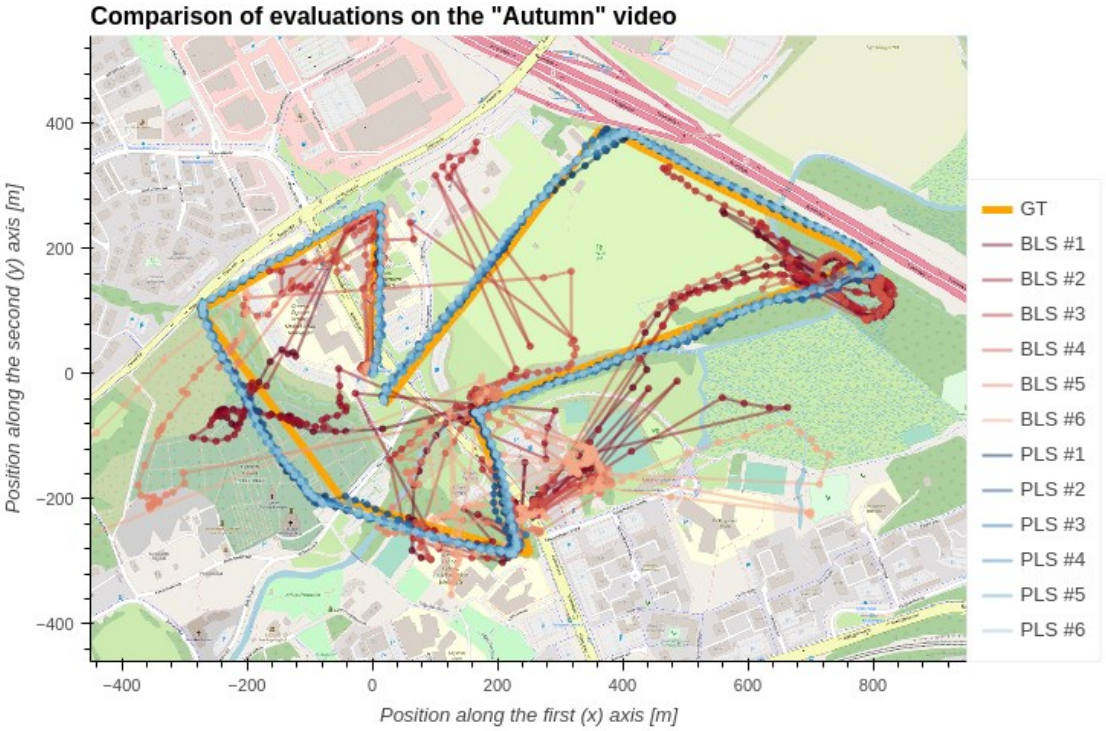

Absolute error of estimated centroids in altitude only
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean

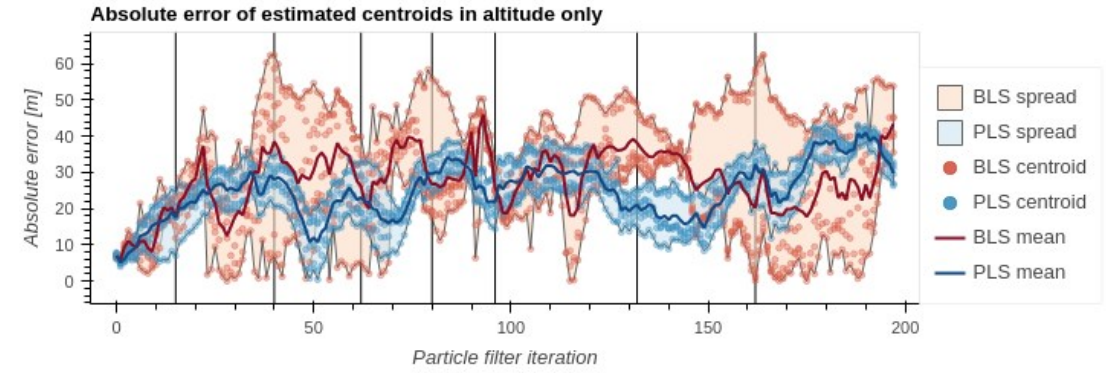

Total distance of estimated centroids to GT
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean

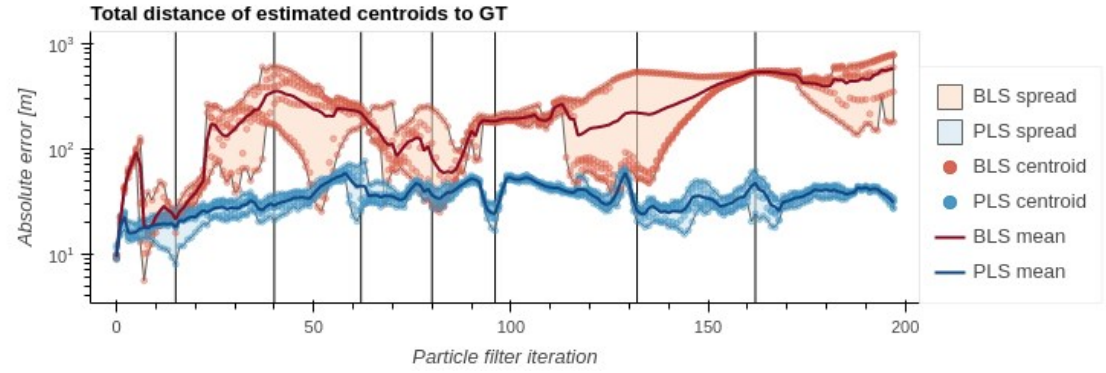

Comparison of evaluations on the "Shadows" video
Position along the second (y) axis [m]
Position along the first (x) axis [m]
GT
BLS #1
BLS #2
BLS #3
BLS #4
BLS #5
BLS #6
PLS #1
PLS #2
PLS #3
PLS #4
PLS #5
PLS #6

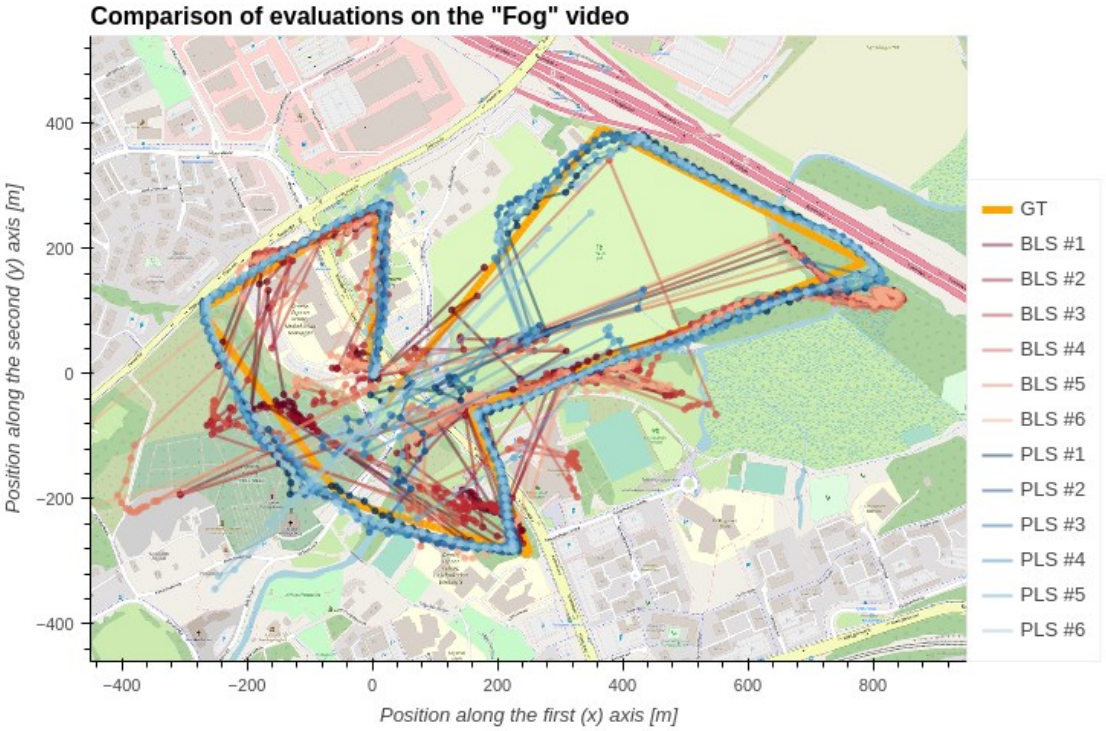

Absolute error of estimated centroids in altitude only
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean

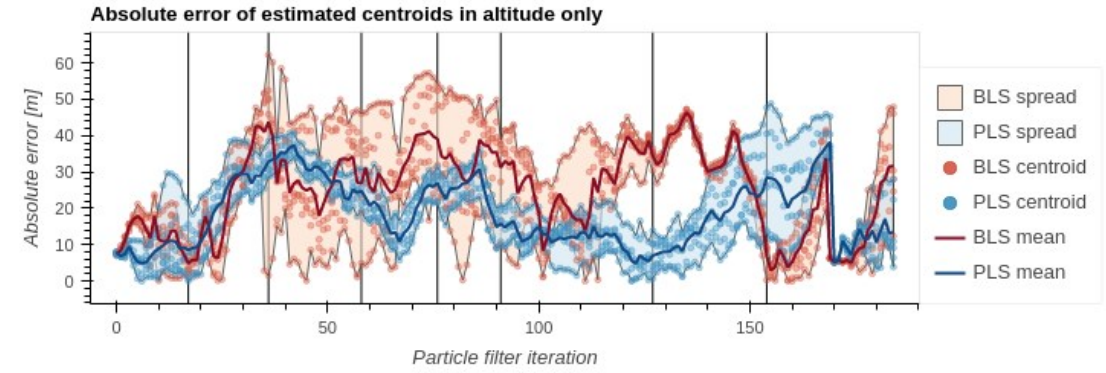

Total distance of estimated centroids to GT
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean

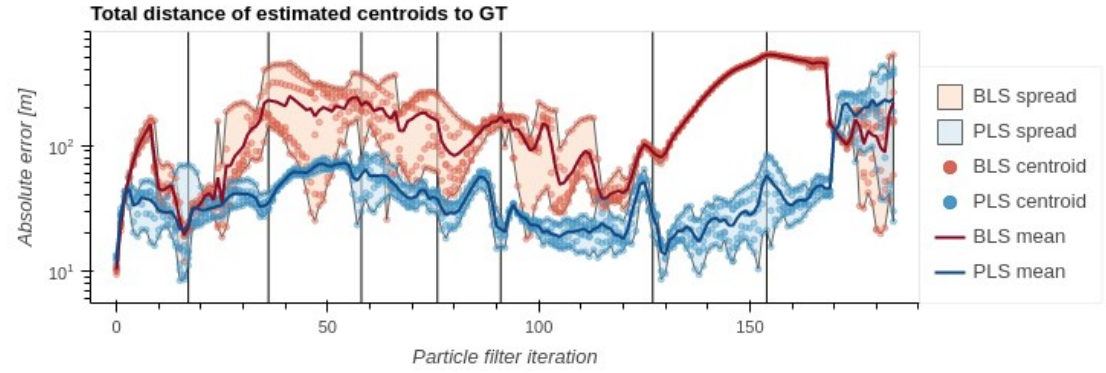

Comparison of evaluations on the "Autumn" video
Position along the second (y) axis [m]
Position along the first (x) axis [m]
GT
BLS #1
BLS #2
BLS #3
BLS #4
BLS #5
BLS #6
PLS #1
PLS #2
PLS #3
PLS #4
PLS #5
PLS #6

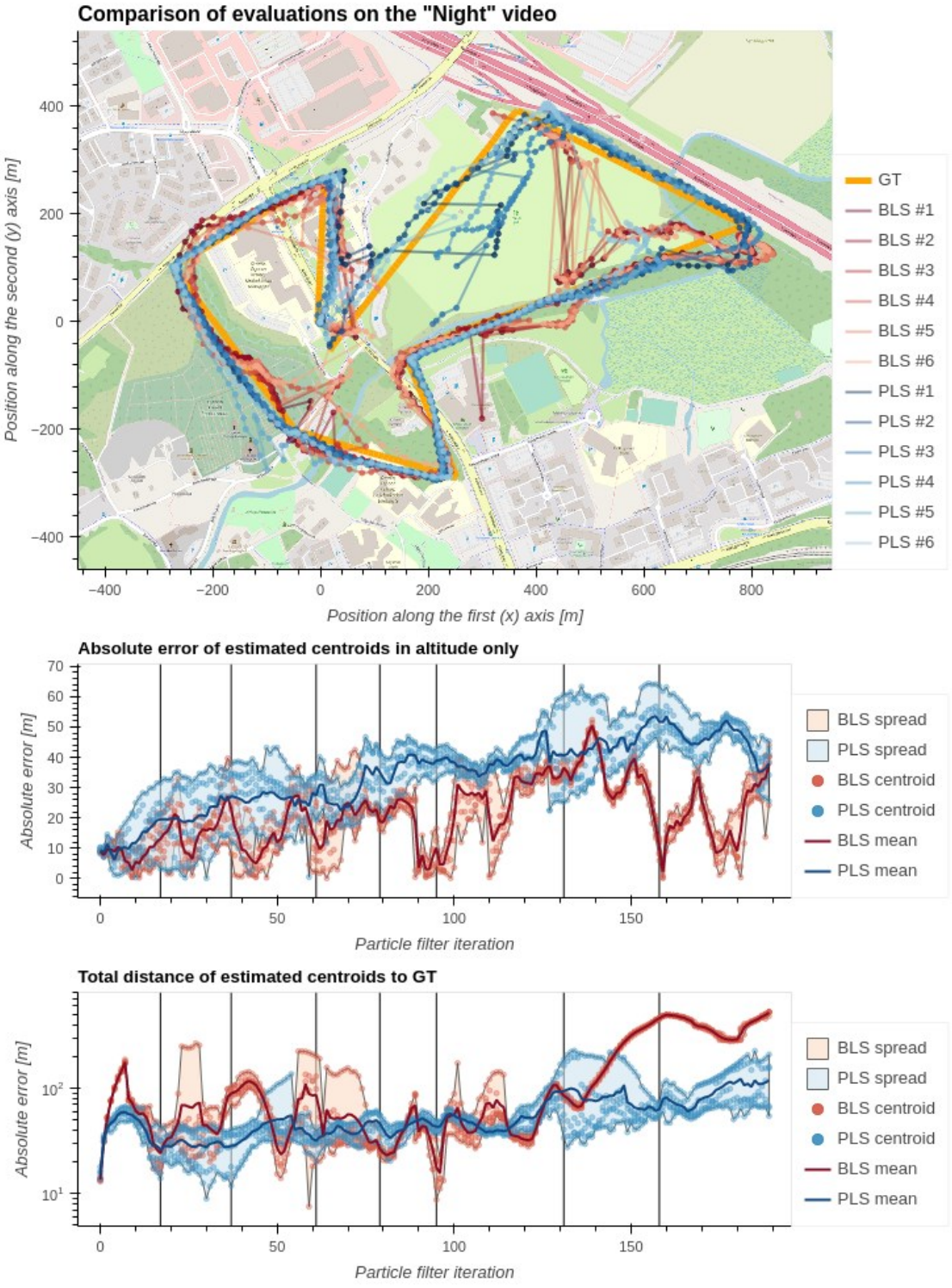

Absolute error of estimated centroids in altitude only
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean

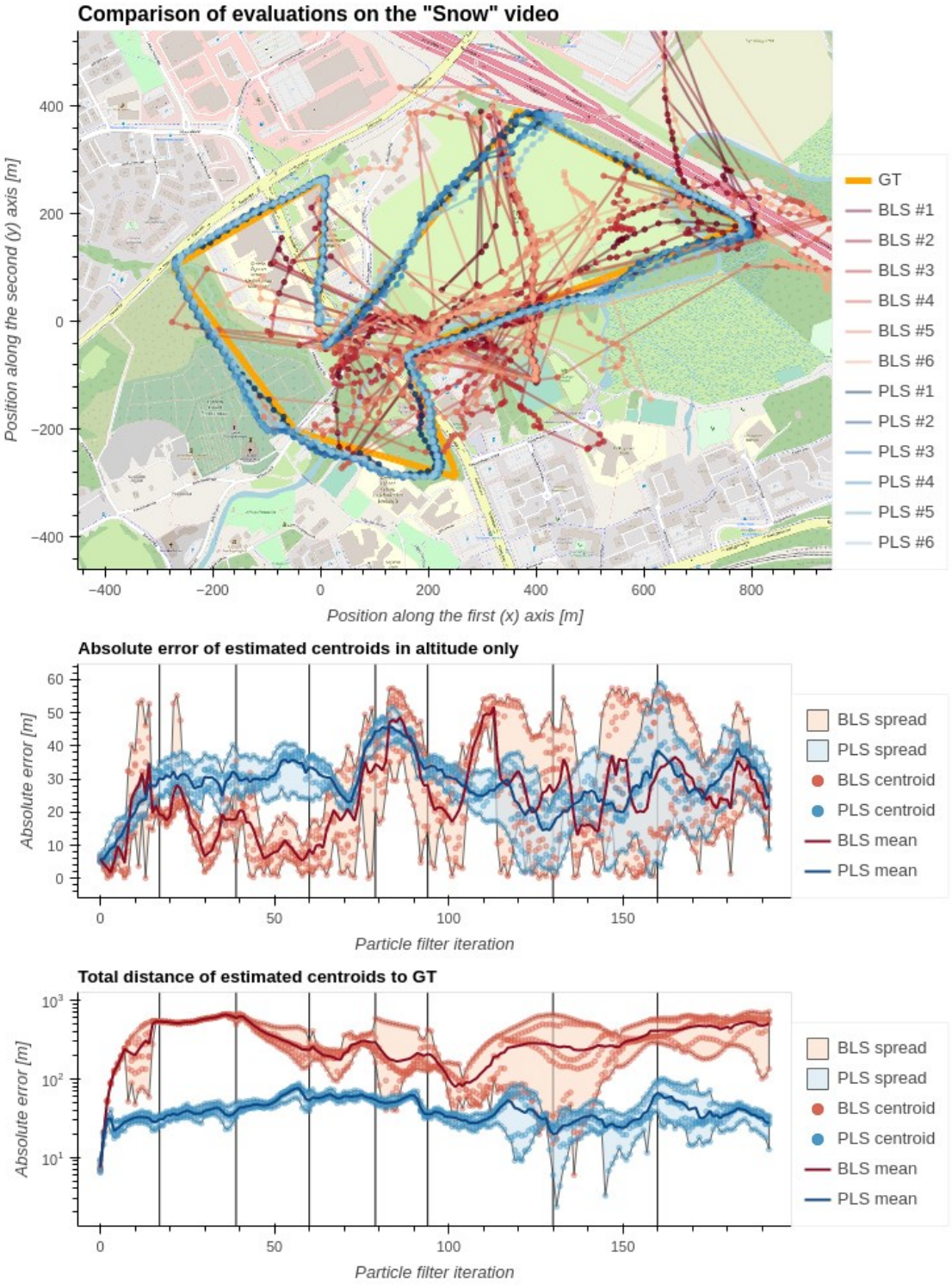

Total distance of estimated centroids to GT
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean


Comparison of evaluations on the "Fog" video
Position along the second (y) axis [m]
Position along the first (x) axis [m]
GT
BLS #1
BLS #2
BLS #3
BLS #4
BLS #5
BLS #6
PLS #1
PLS #2
PLS #3
PLS #4
PLS #5
PLS #6


Absolute error of estimated centroids in altitude only
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean


Total distance of estimated centroids to GT
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean


Comparison of evaluations on the "Night" video
Position along the second (y) axis [m]
Position along the first (x) axis [m]
GT
BLS #1
BLS #2
BLS #3
BLS #4
BLS #5
BLS #6
PLS #1
PLS #2
PLS #3
PLS #4
PLS #5
PLS #6
Absolute error of estimated centroids in altitude only
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean
Total distance of estimated centroids to GT
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean

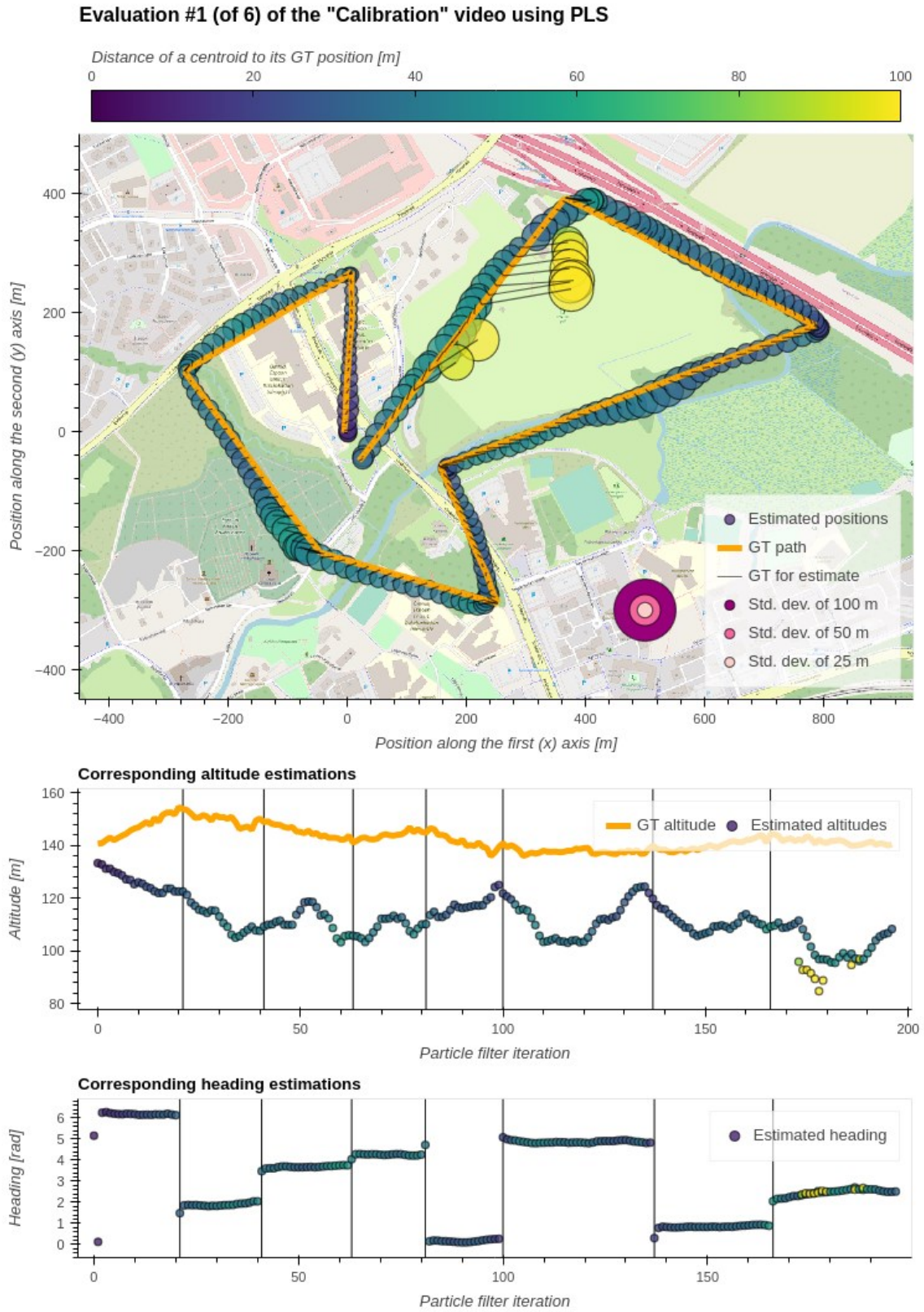

Comparison of evaluations on the "Snow" video
Position along the second (y) axis [m]
Position along the first (x) axis [m]
GT
BLS #1
BLS #2
BLS #3
BLS #4
BLS #5
BLS #6
PLS #1
PLS #2
PLS #3
PLS #4
PLS #5
PLS #6
Absolute error of estimated centroids in altitude only
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean
Total distance of estimated centroids to GT
Absolute error [m]
Particle filter iteration
BLS spread
PLS spread
BLS centroid
PLS centroid
BLS mean
PLS mean

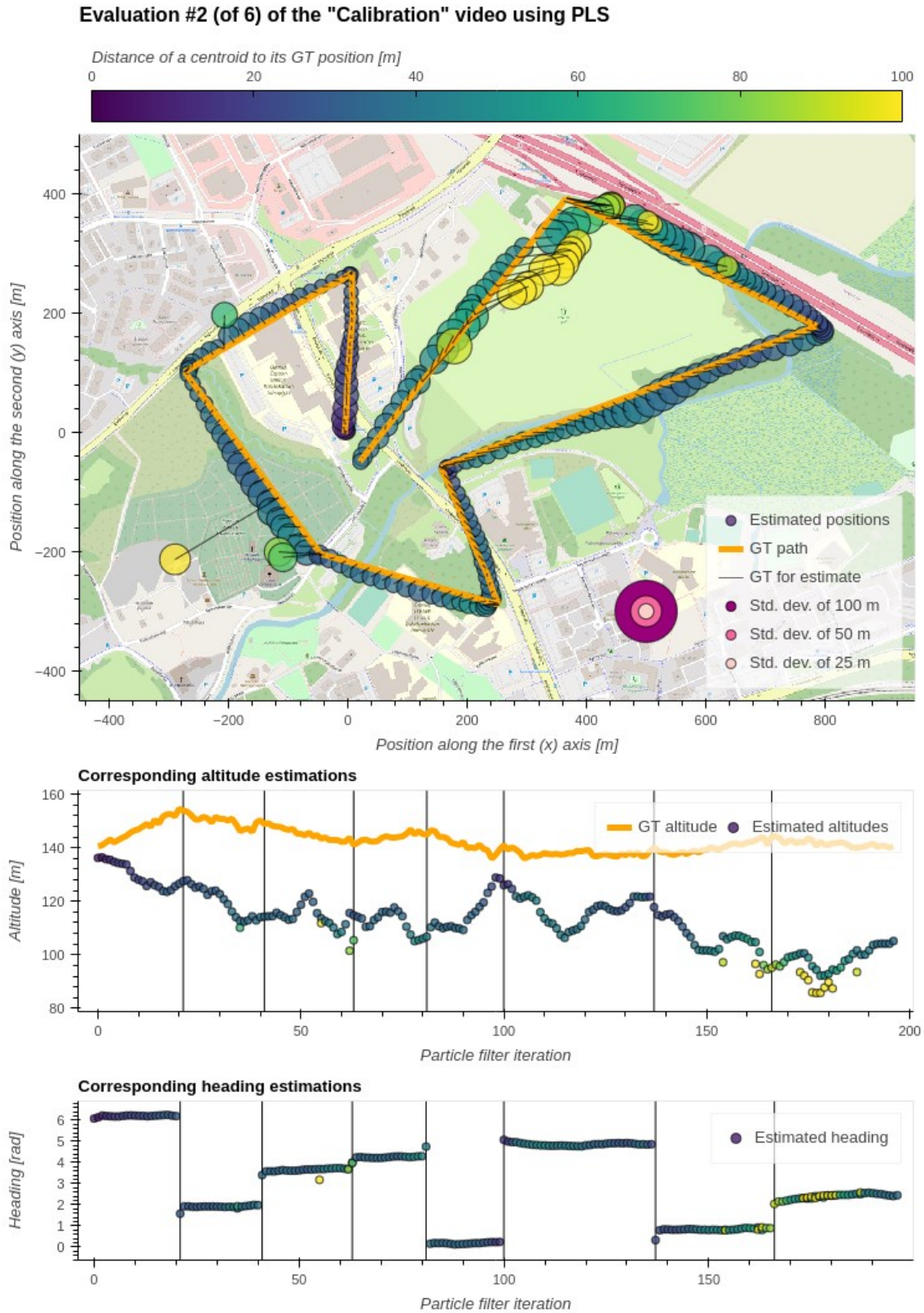
Appendix 2: Individual results

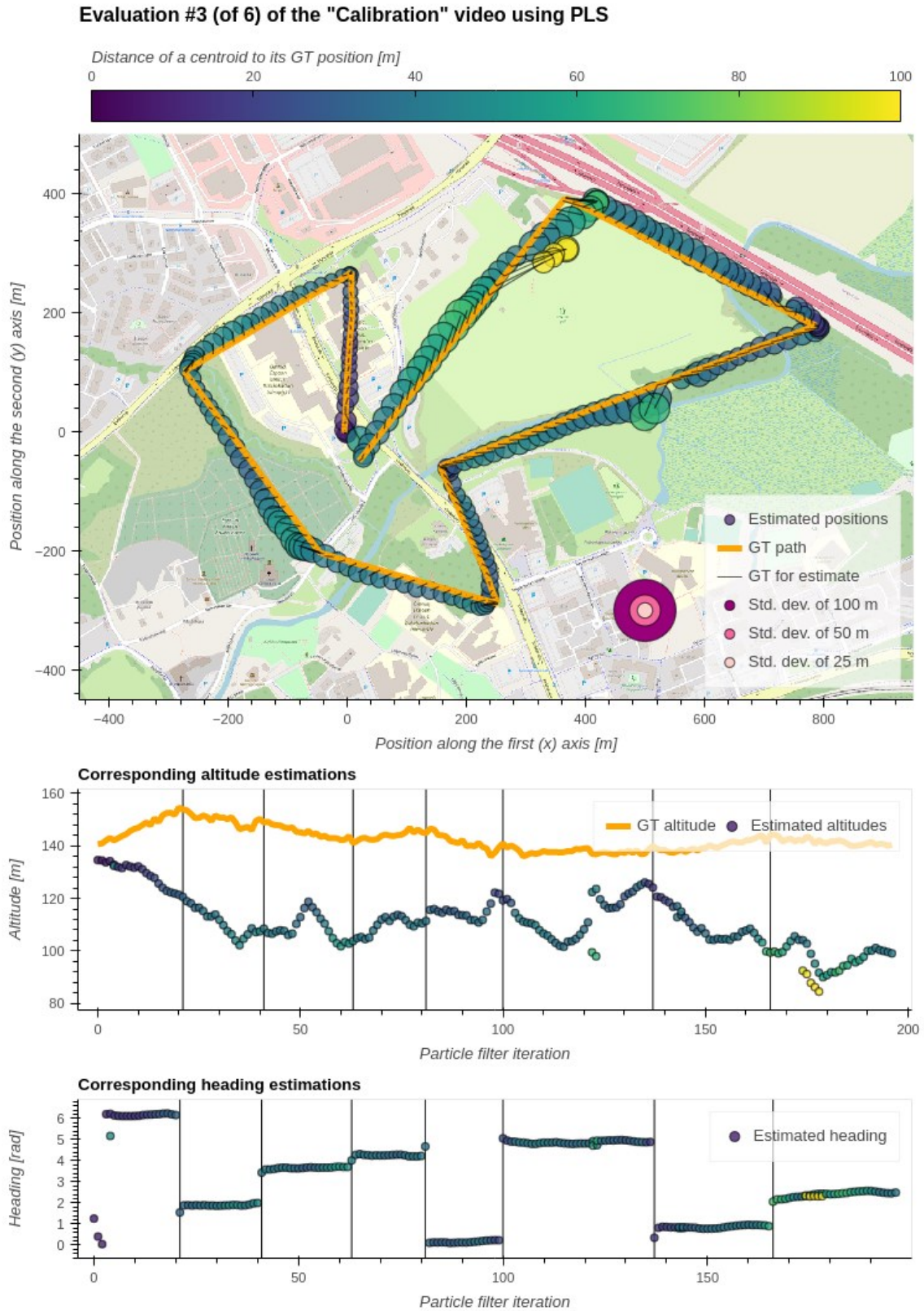
The following 72 figures represent all individual evaluations performed. The order of presentation follows firstly video selection with 6 figures based on PLS evaluation results and subsequently 6 figures based on BLS evaluation results. The order of the video evaluations is as follows:

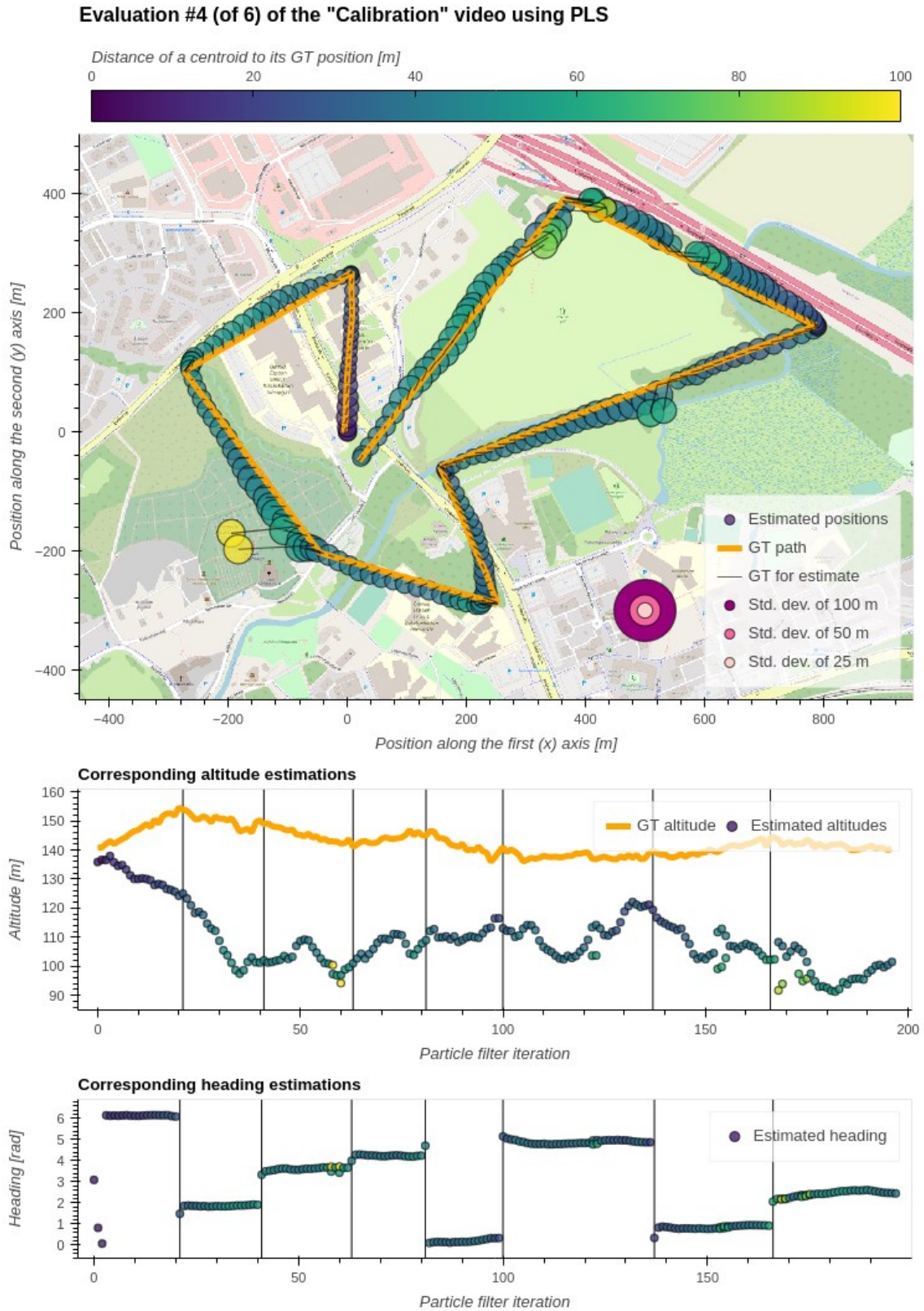
- “Calibration”
- “Shadows”
- “Autumn”
- “Fog”
- “Night”
- “Snow.”

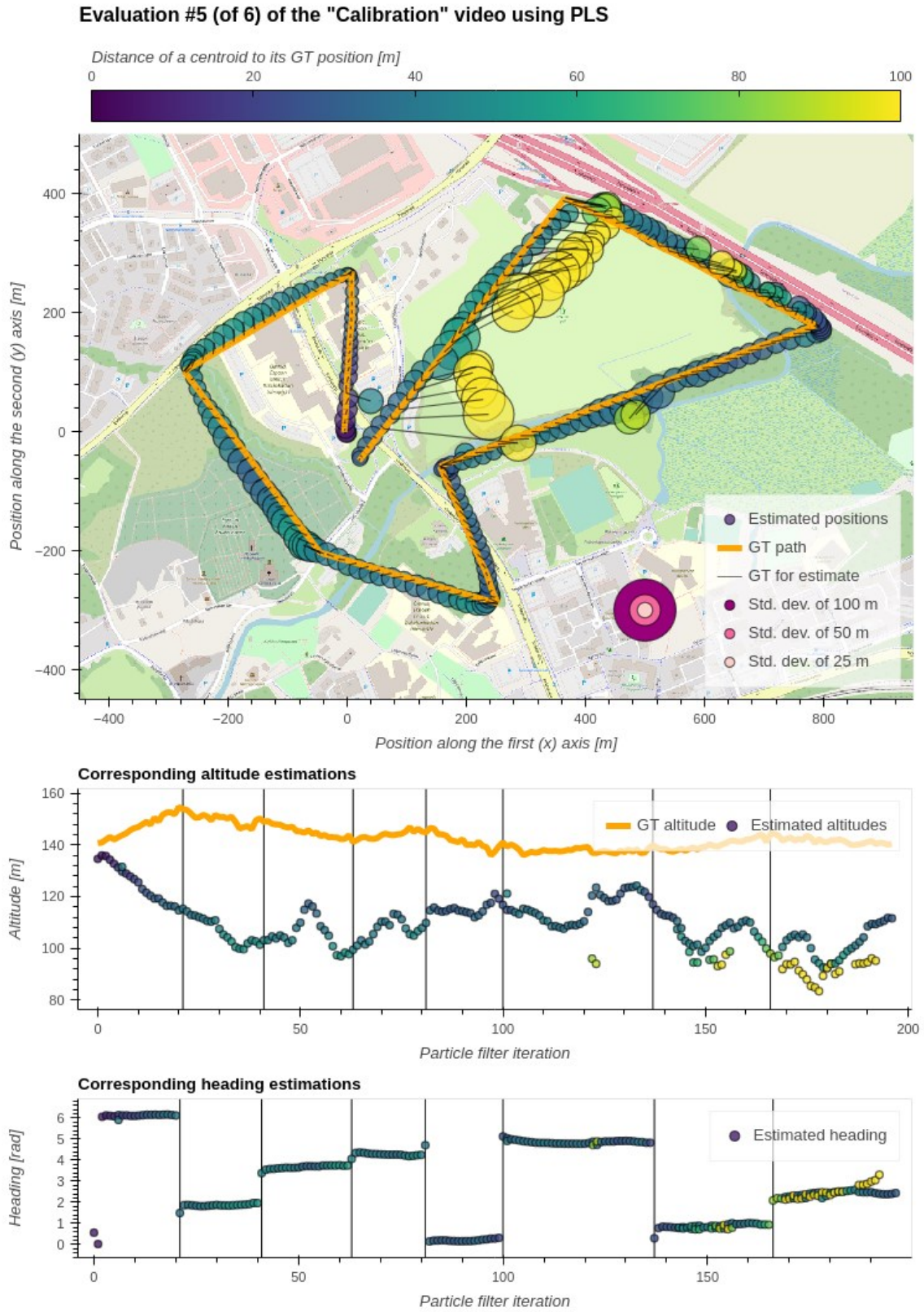
Note that each figure denotes information in its title about the evaluation method and the data it was evaluated on. The detailed description of the figures can be found in section 5.4.

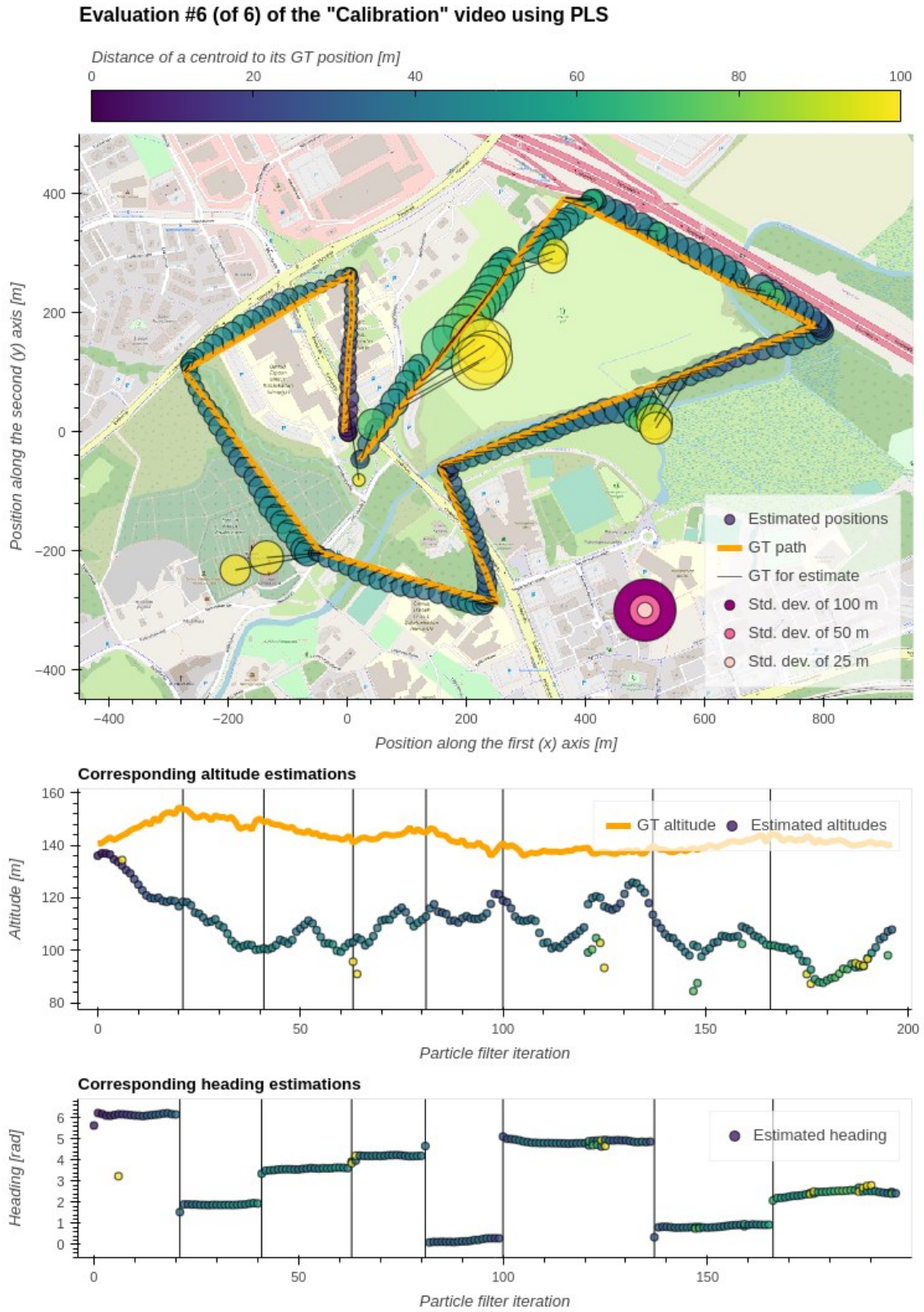

Evaluation #1 (of 6) of the "Calibration" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
200
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

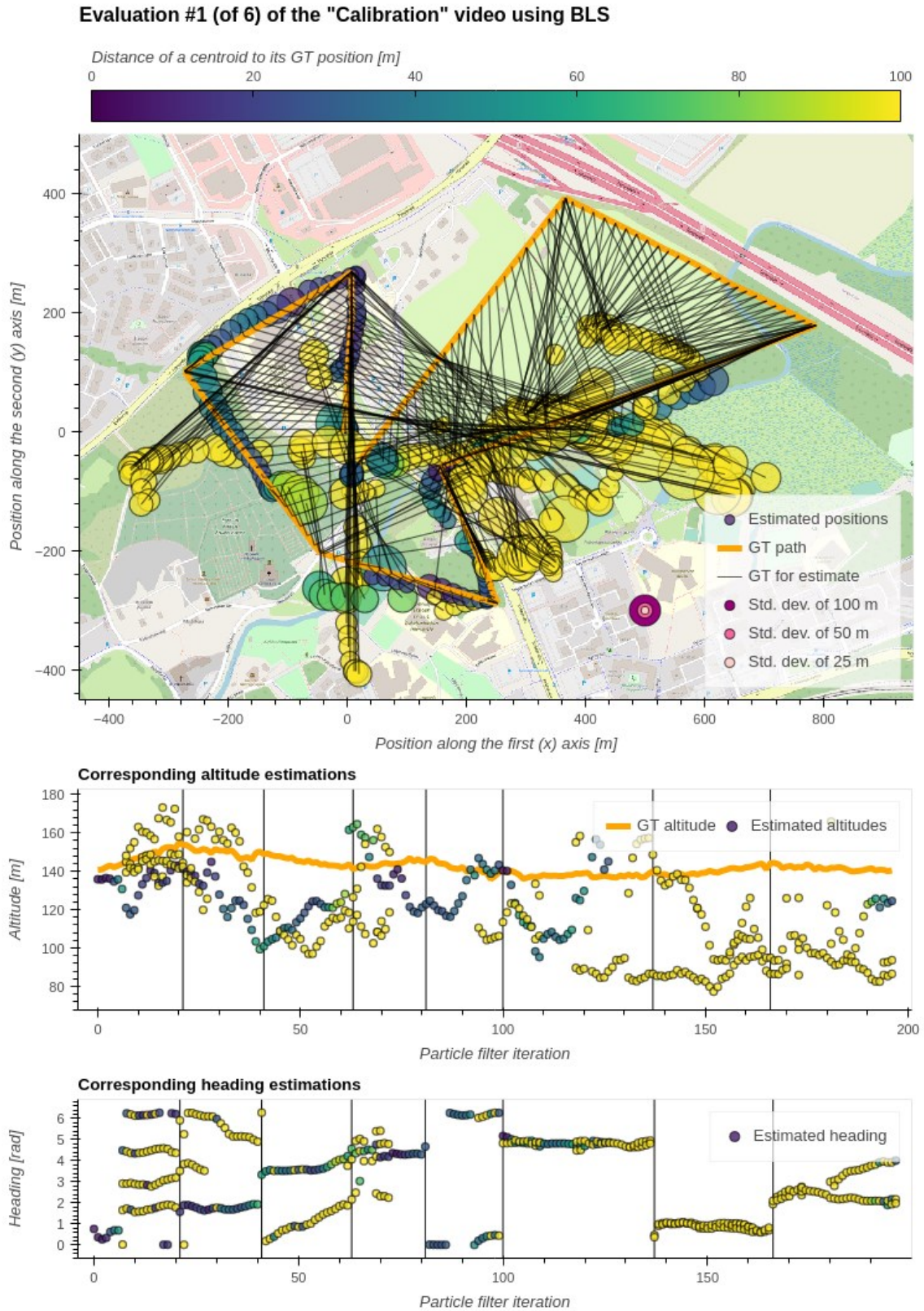

Evaluation #2 (of 6) of the "Calibration" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
200
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

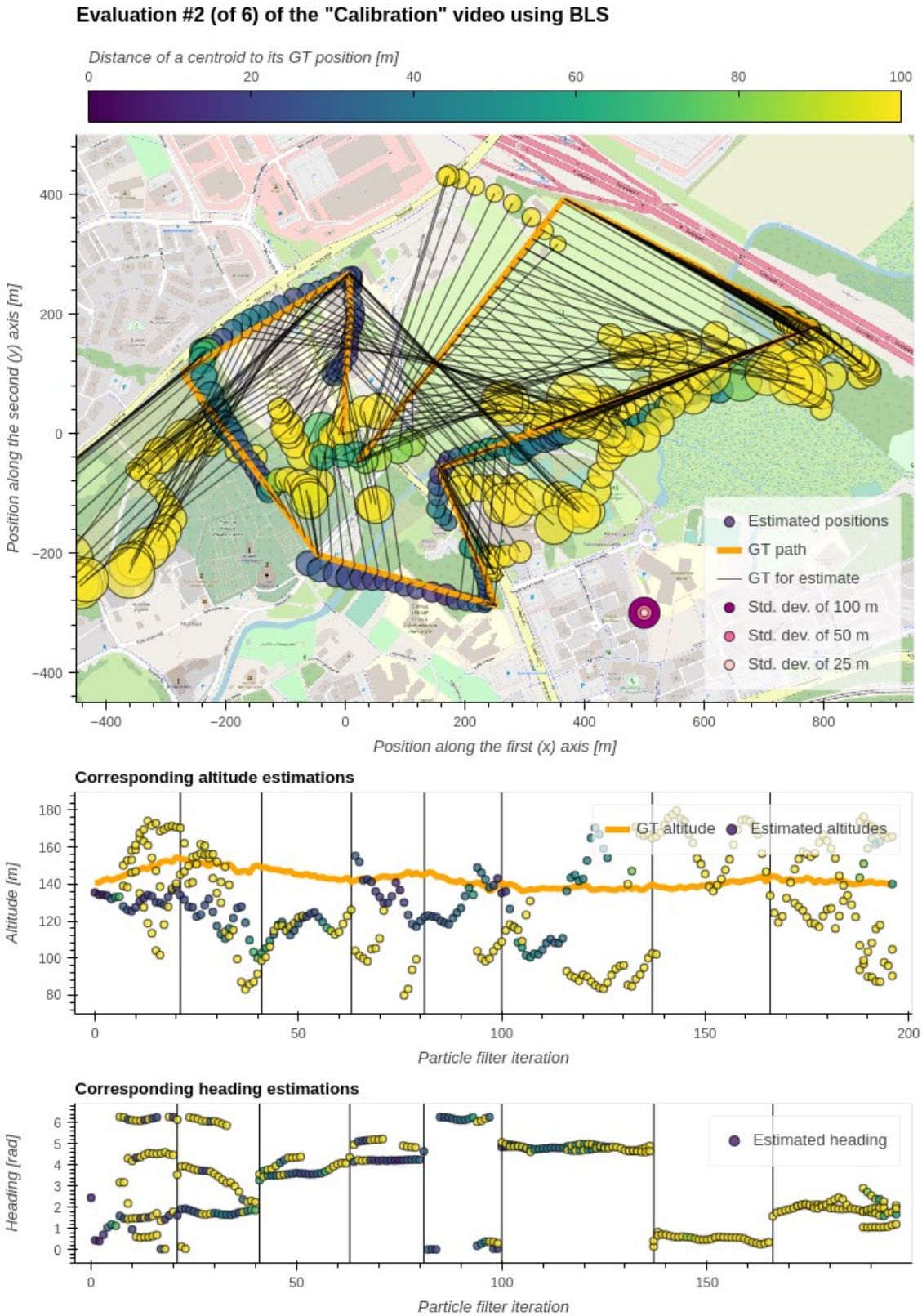

Evaluation #3 (of 6) of the "Calibration" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
200
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

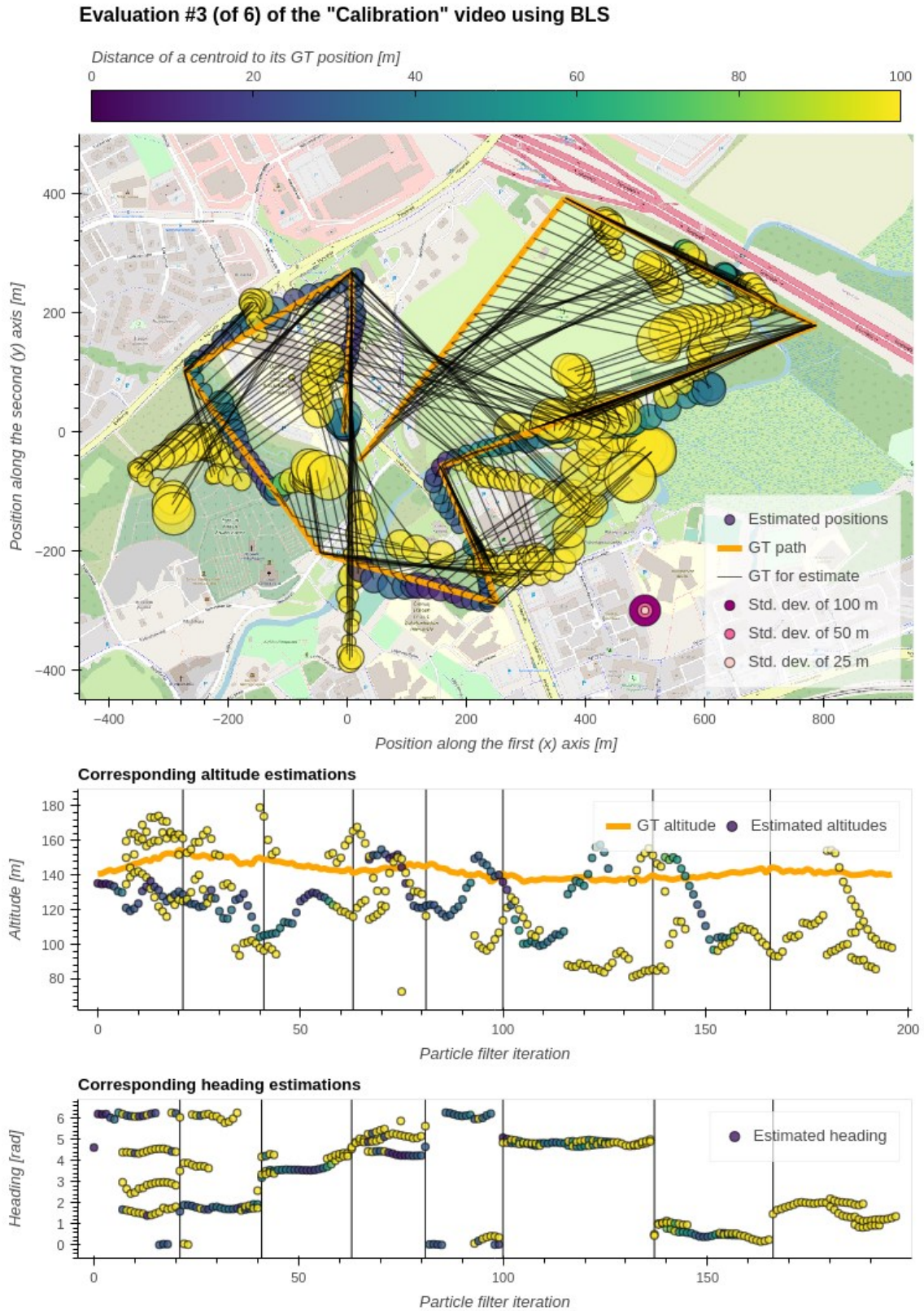

Evaluation #4 (of 6) of the "Calibration" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
150
140
130
120
110
100
90
GT altitude
Estimated altitudes
0
50
100
150
200
Particle filter iteration
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
Estimated heading
0
50
100
150
Particle filter iteration

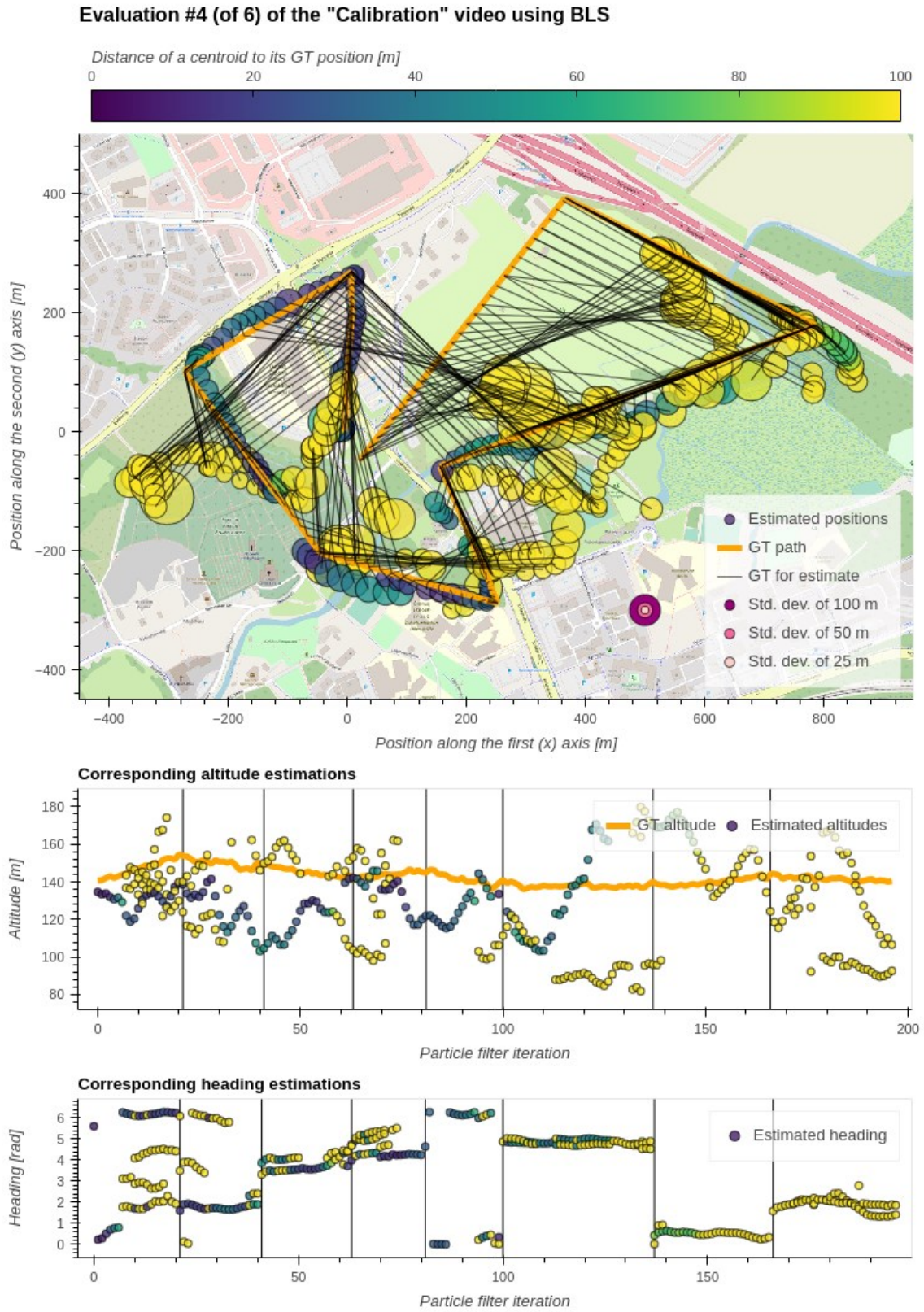

Evaluation #5 (of 6) of the "Calibration" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
200
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

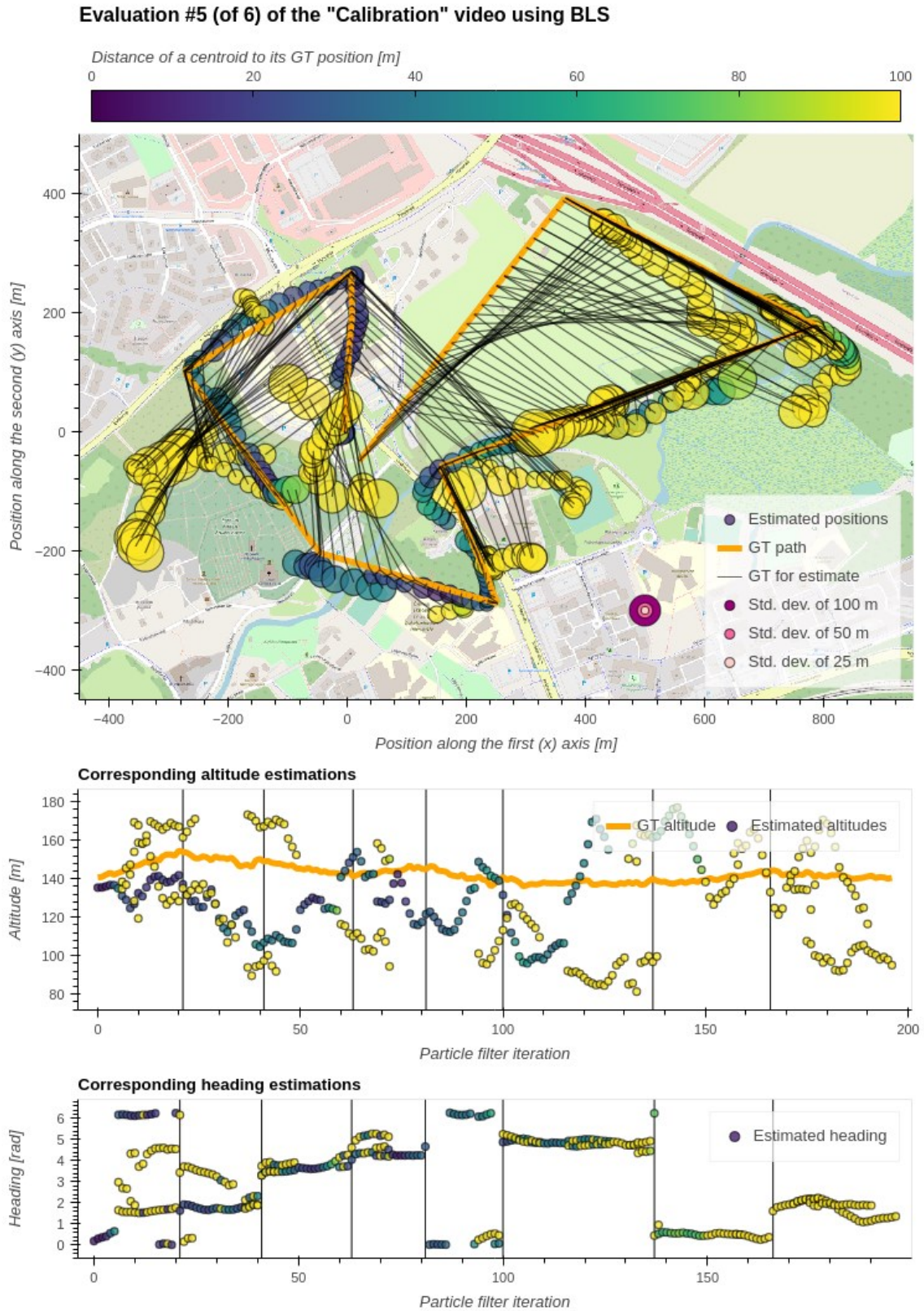

Evaluation #6 (of 6) of the "Calibration" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
200
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

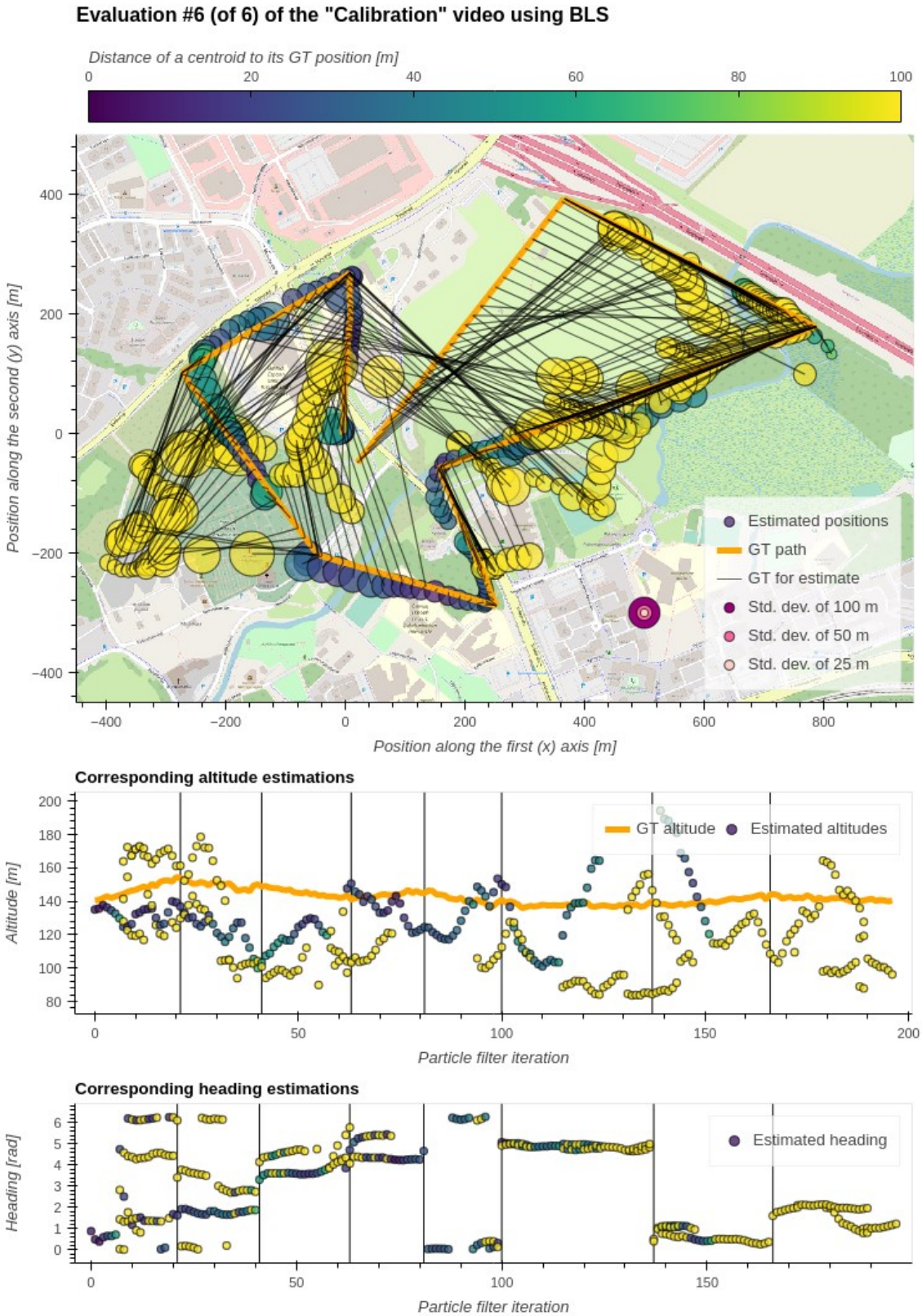

Evaluation #1 (of 6) of the "Calibration" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
200
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

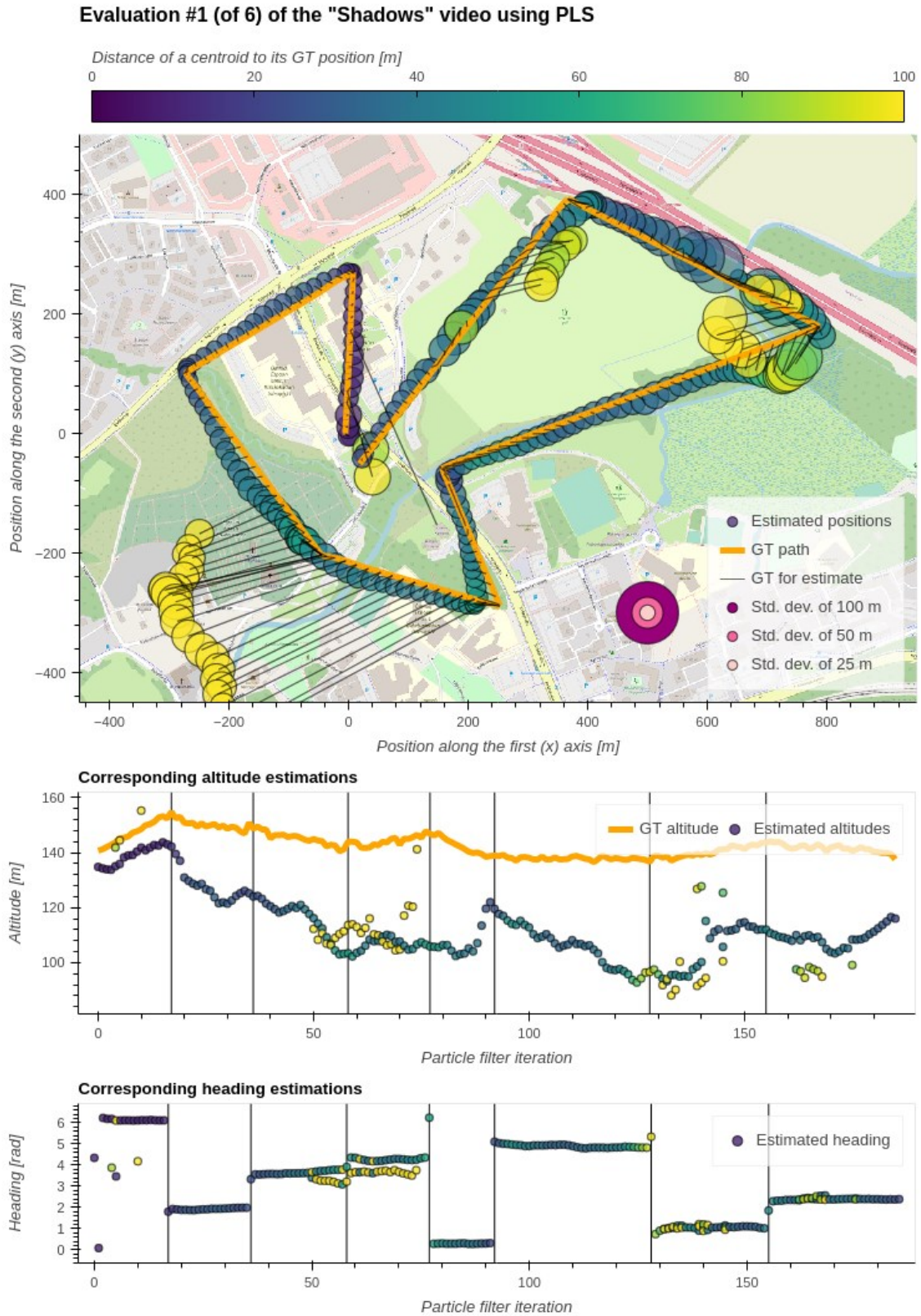

Evaluation #2 (of 6) of the "Calibration" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
200
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

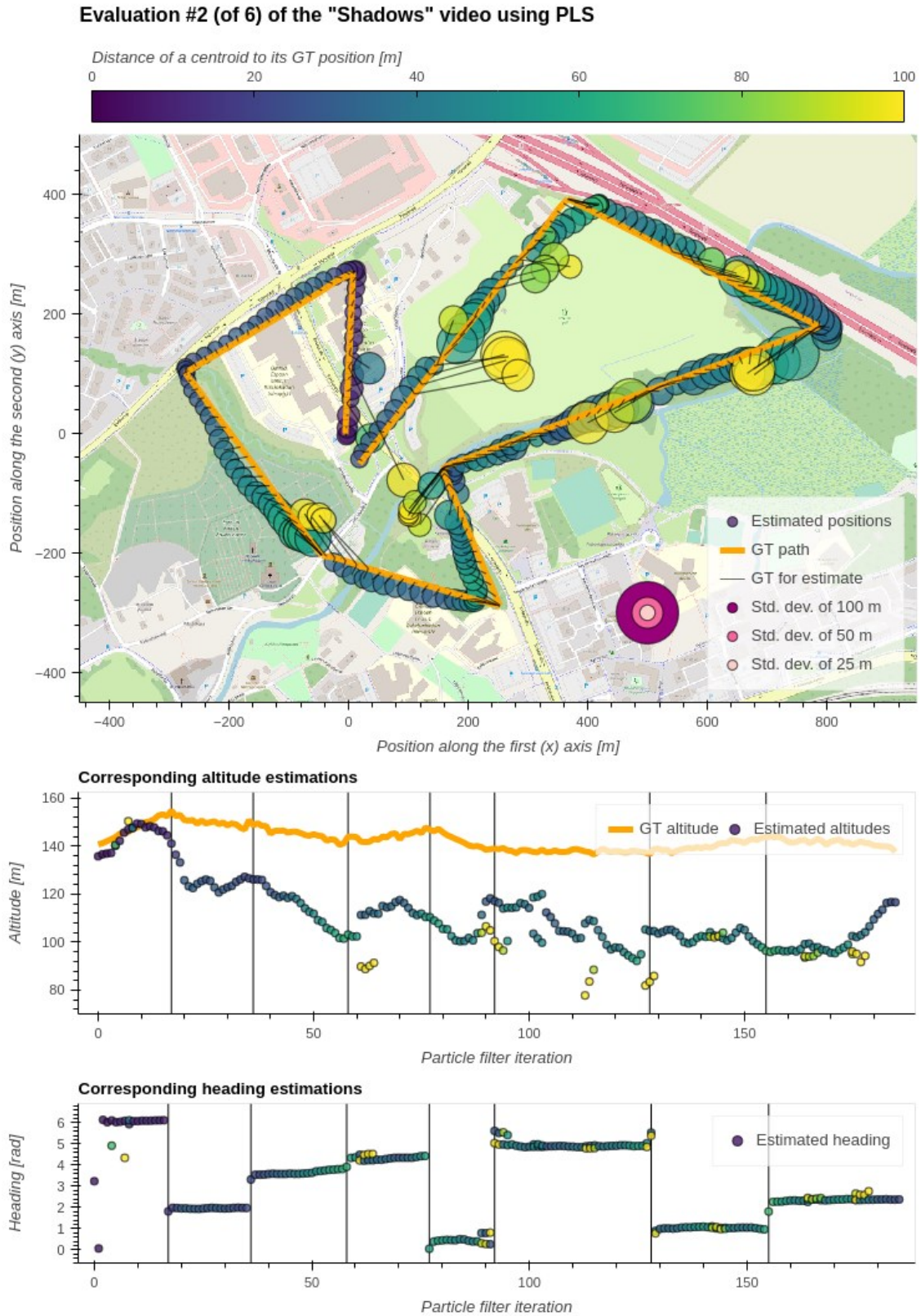

Evaluation #3 (of 6) of the "Calibration" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
200
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

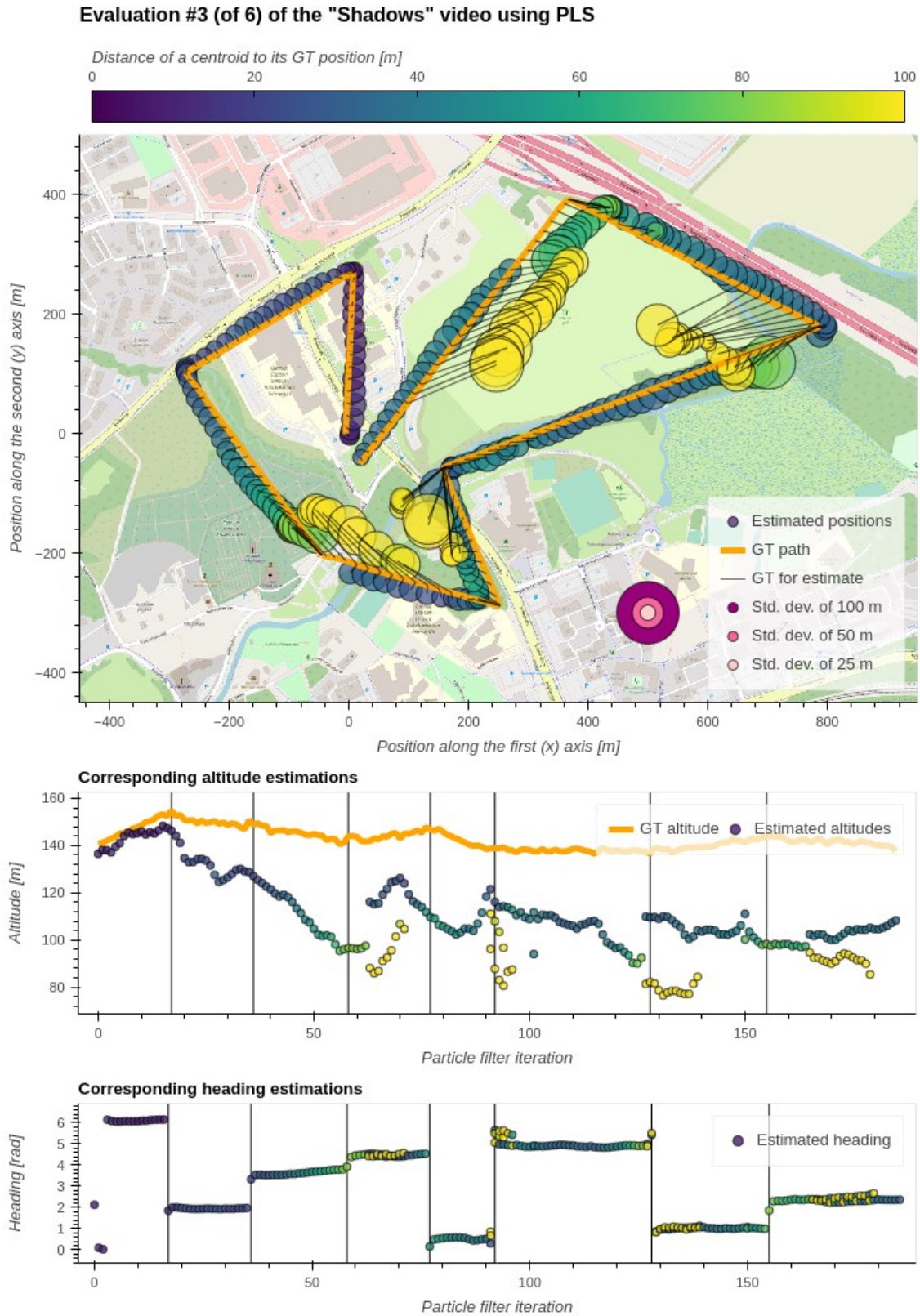

Evaluation #4 (of 6) of the "Calibration" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

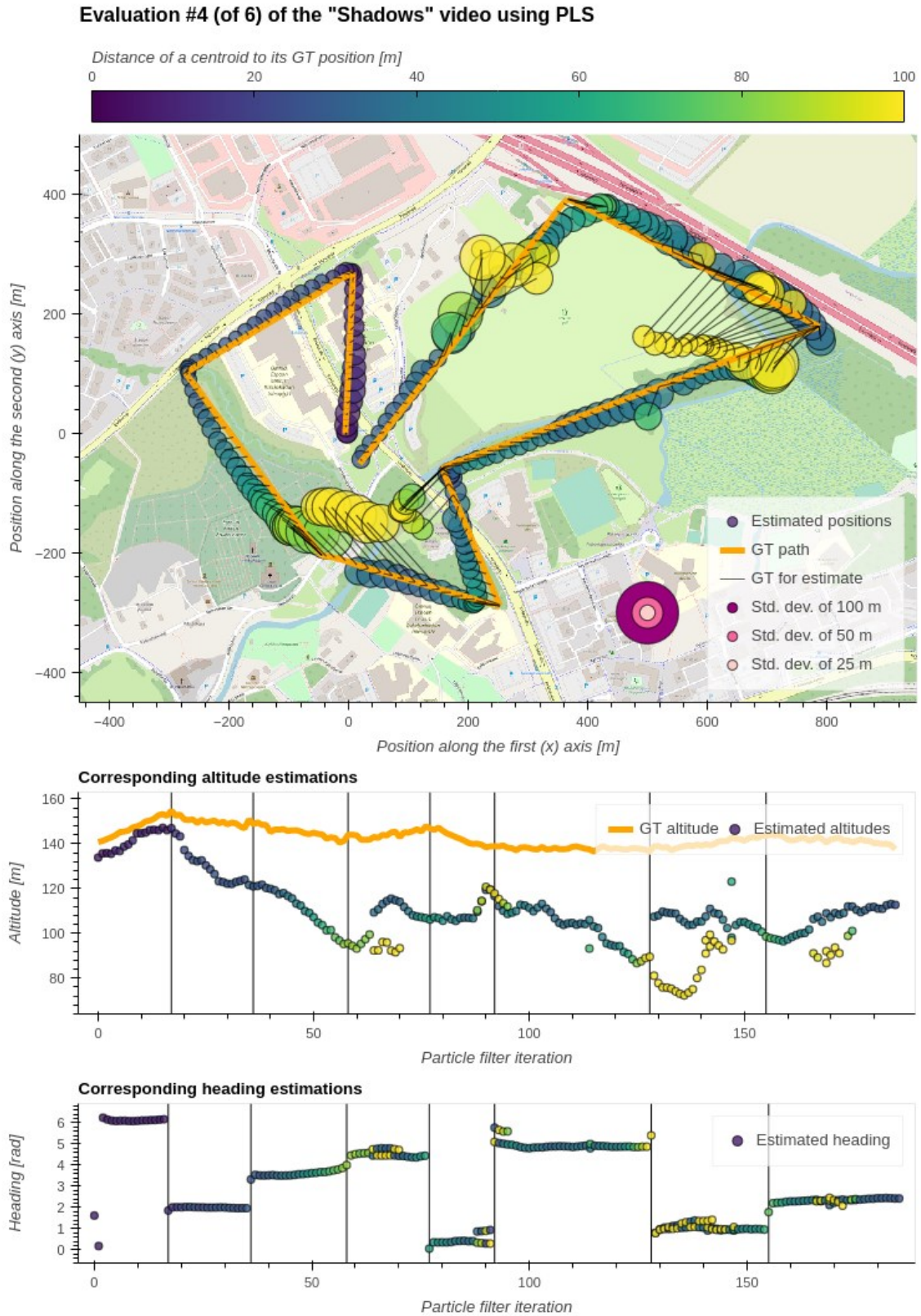

Evaluation #5 (of 6) of the "Calibration" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
200
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

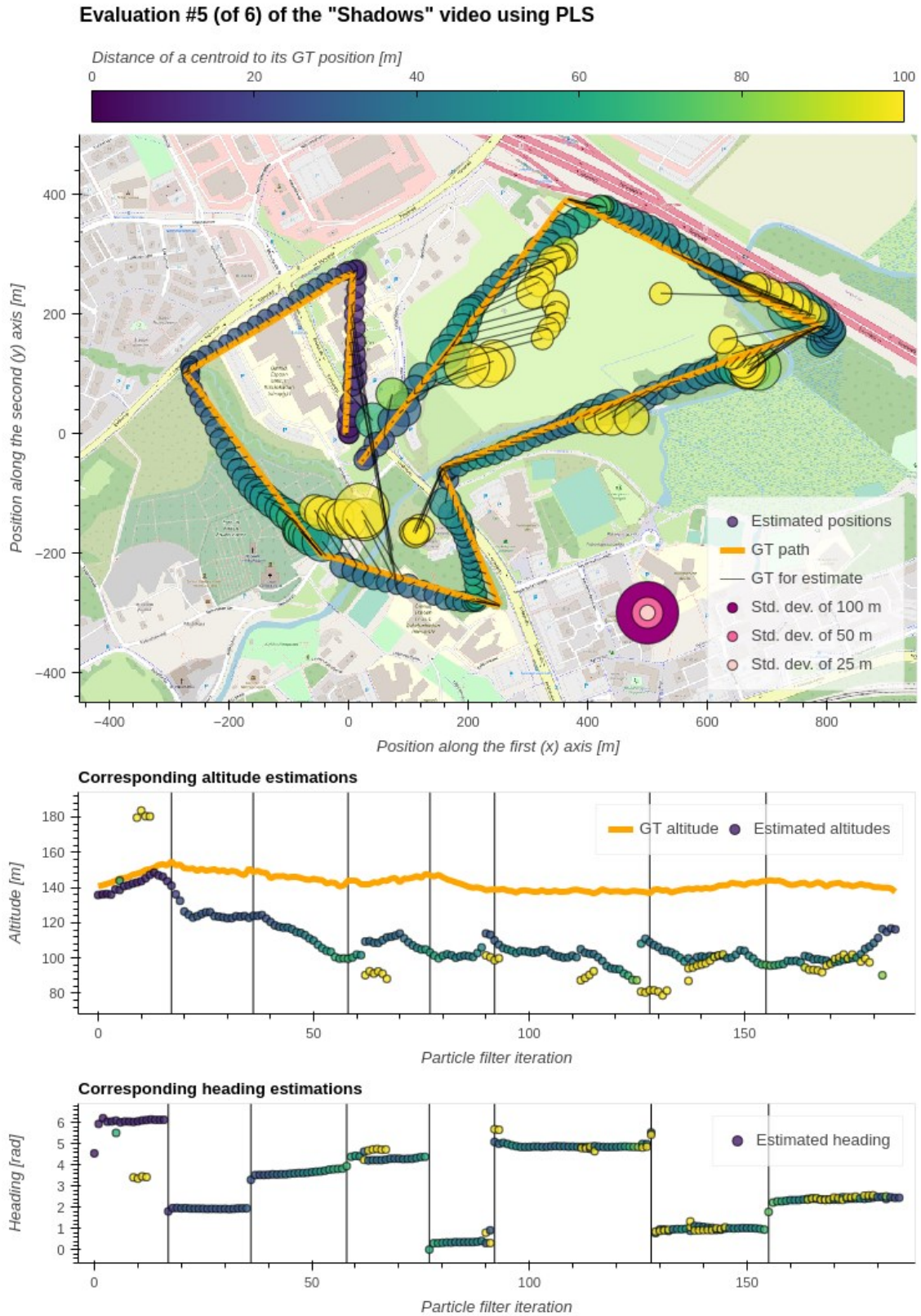

Evaluation #6 (of 6) of the "Calibration" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
200
180
160
140
120
100
80
0
50
100
150
200
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

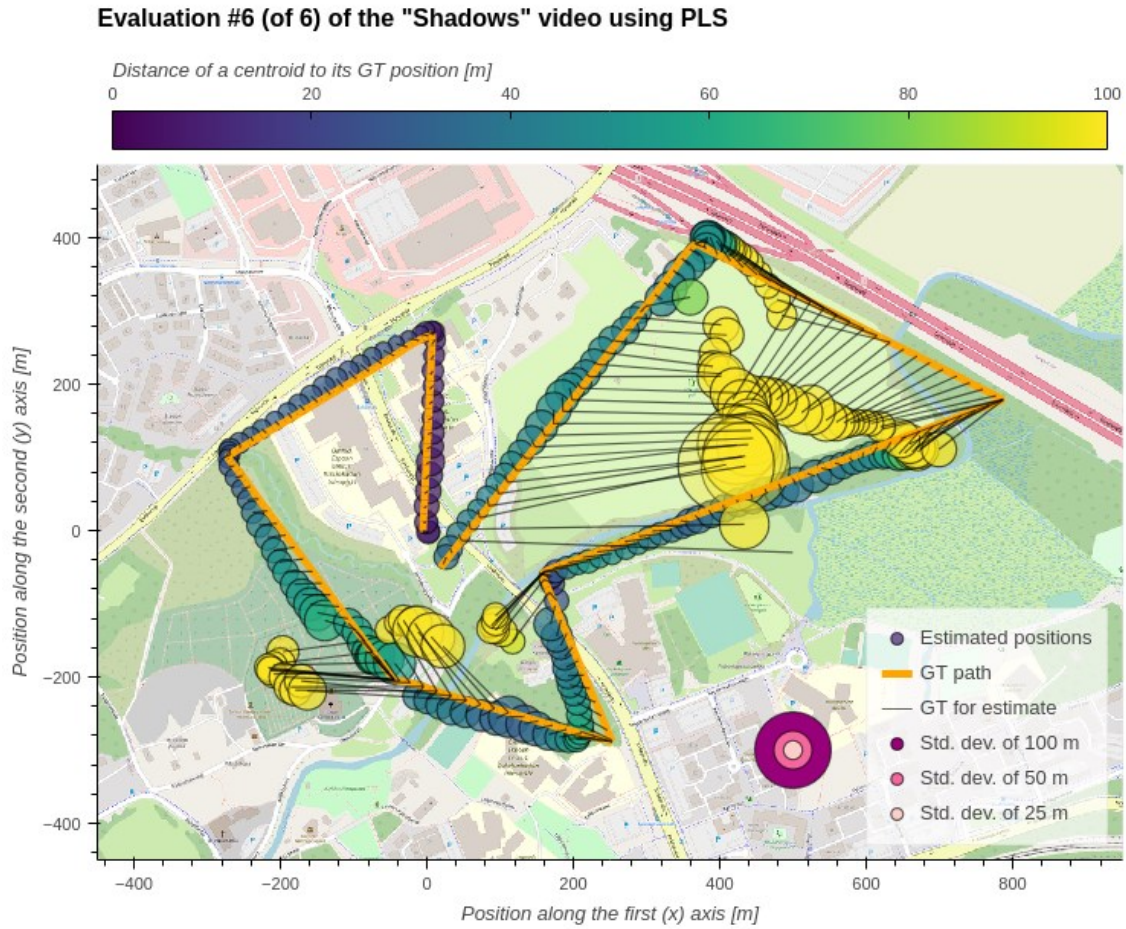

Evaluation #1 (of 6) of the "Shadows" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

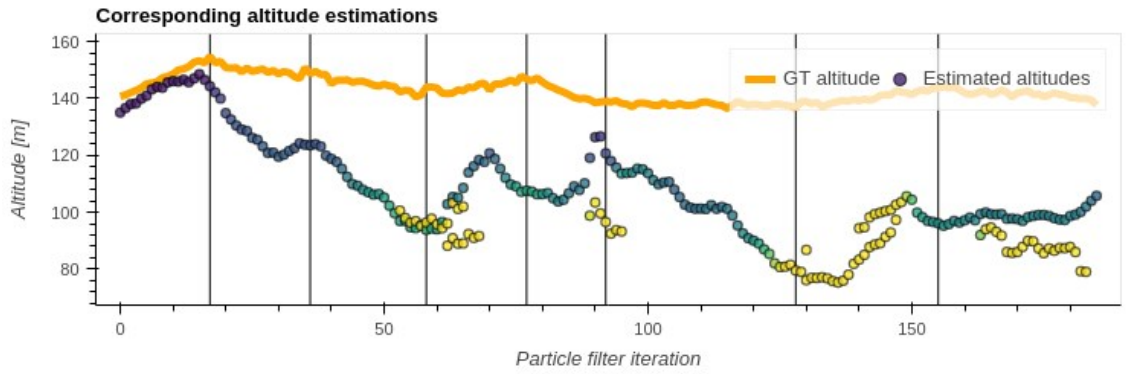

Evaluation #2 (of 6) of the "Shadows" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

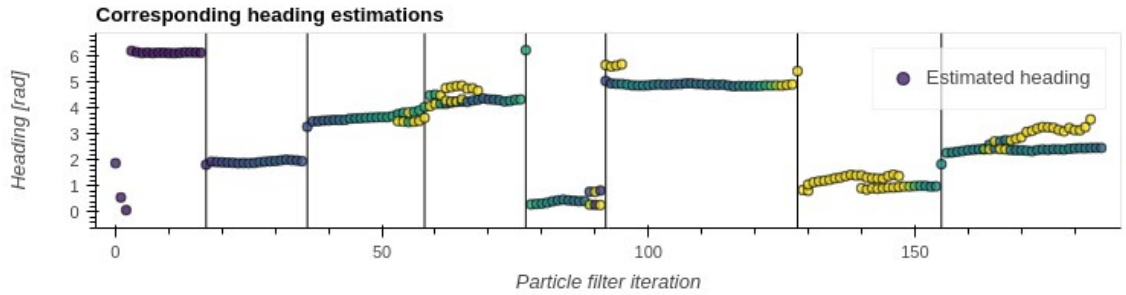

Evaluation #3 (of 6) of the "Shadows" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

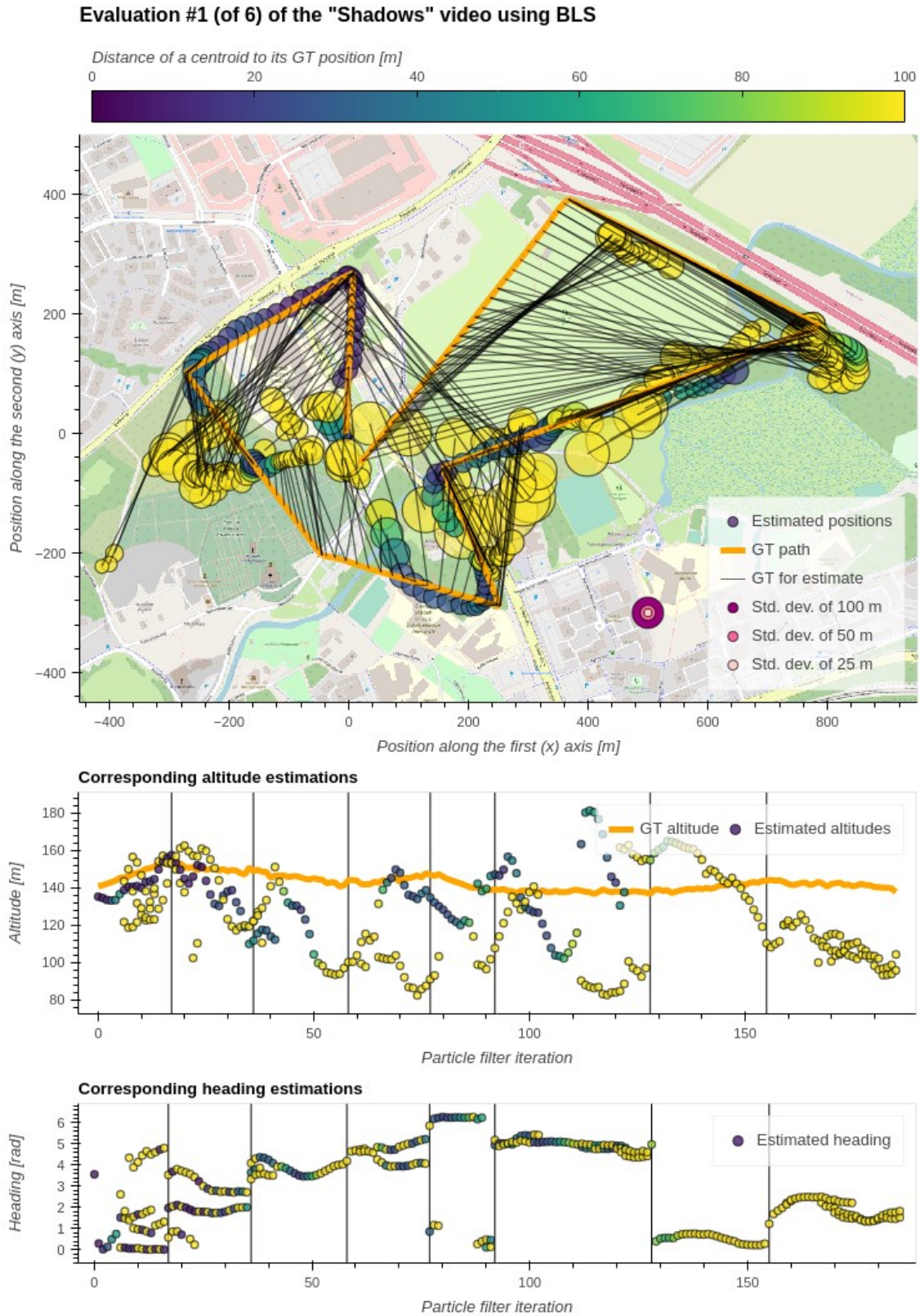

Evaluation #4 (of 6) of the "Shadows" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

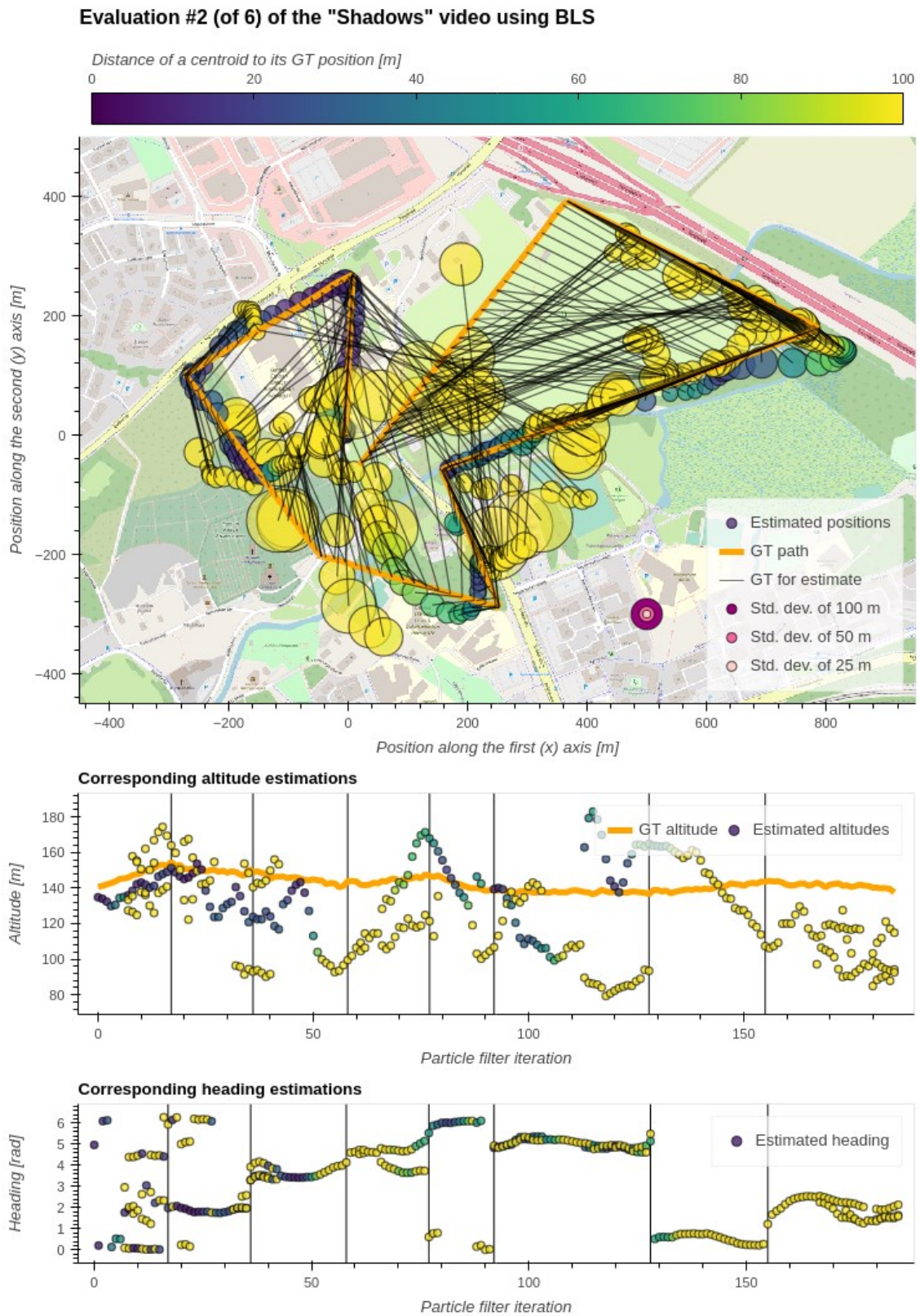

Evaluation #5 (of 6) of the "Shadows" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

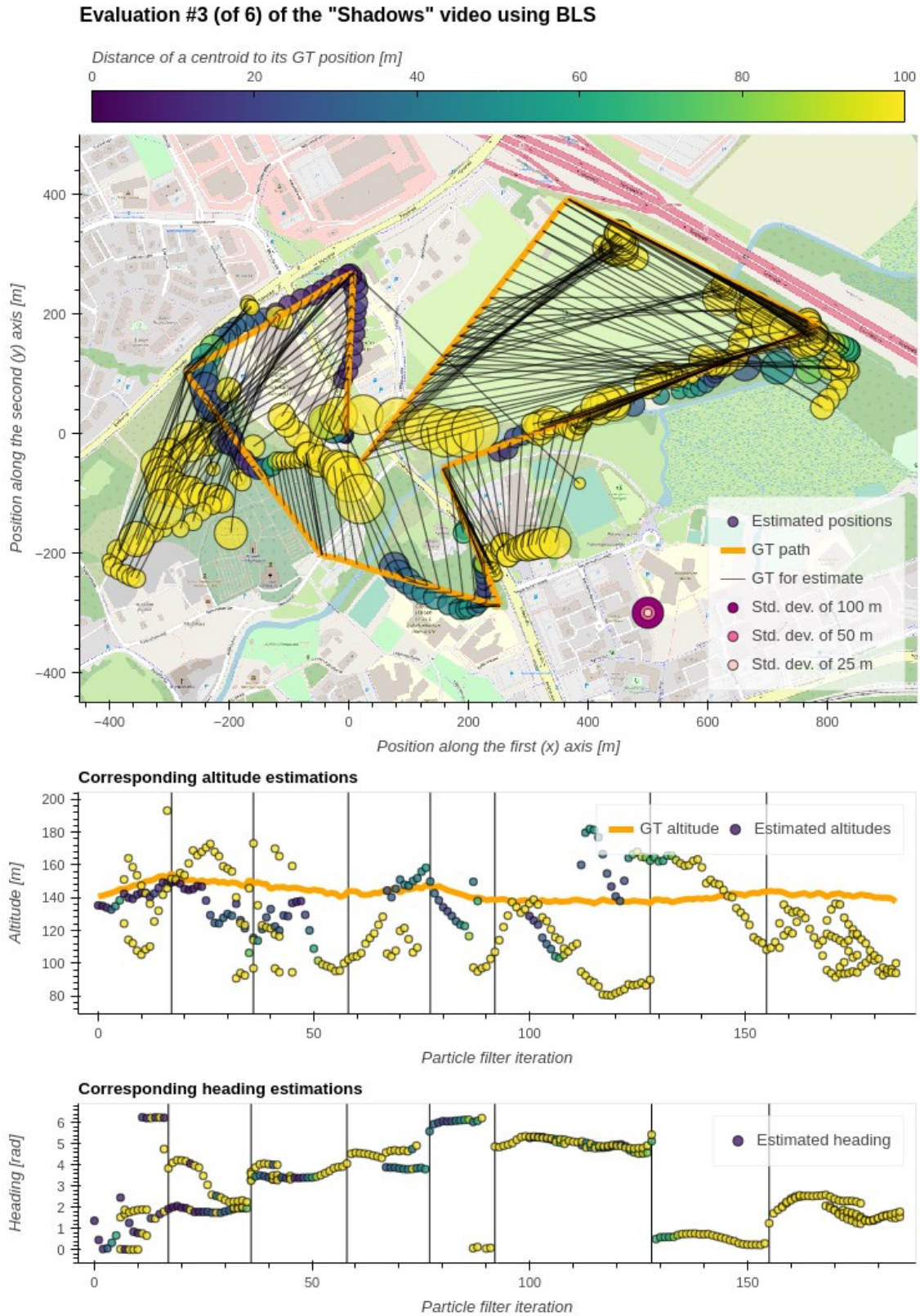

Evaluation #6 (of 6) of the "Shadows" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m

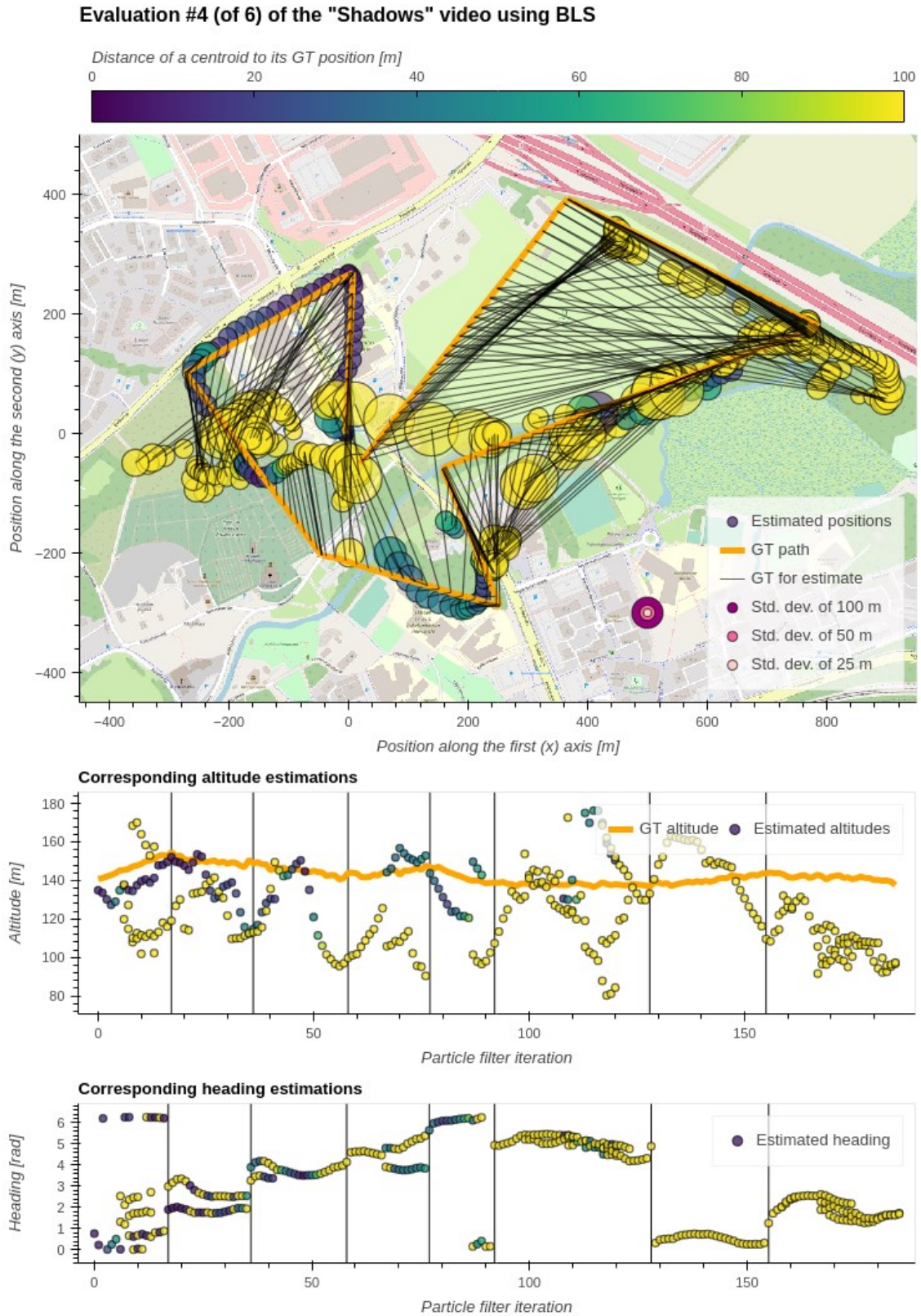

Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes

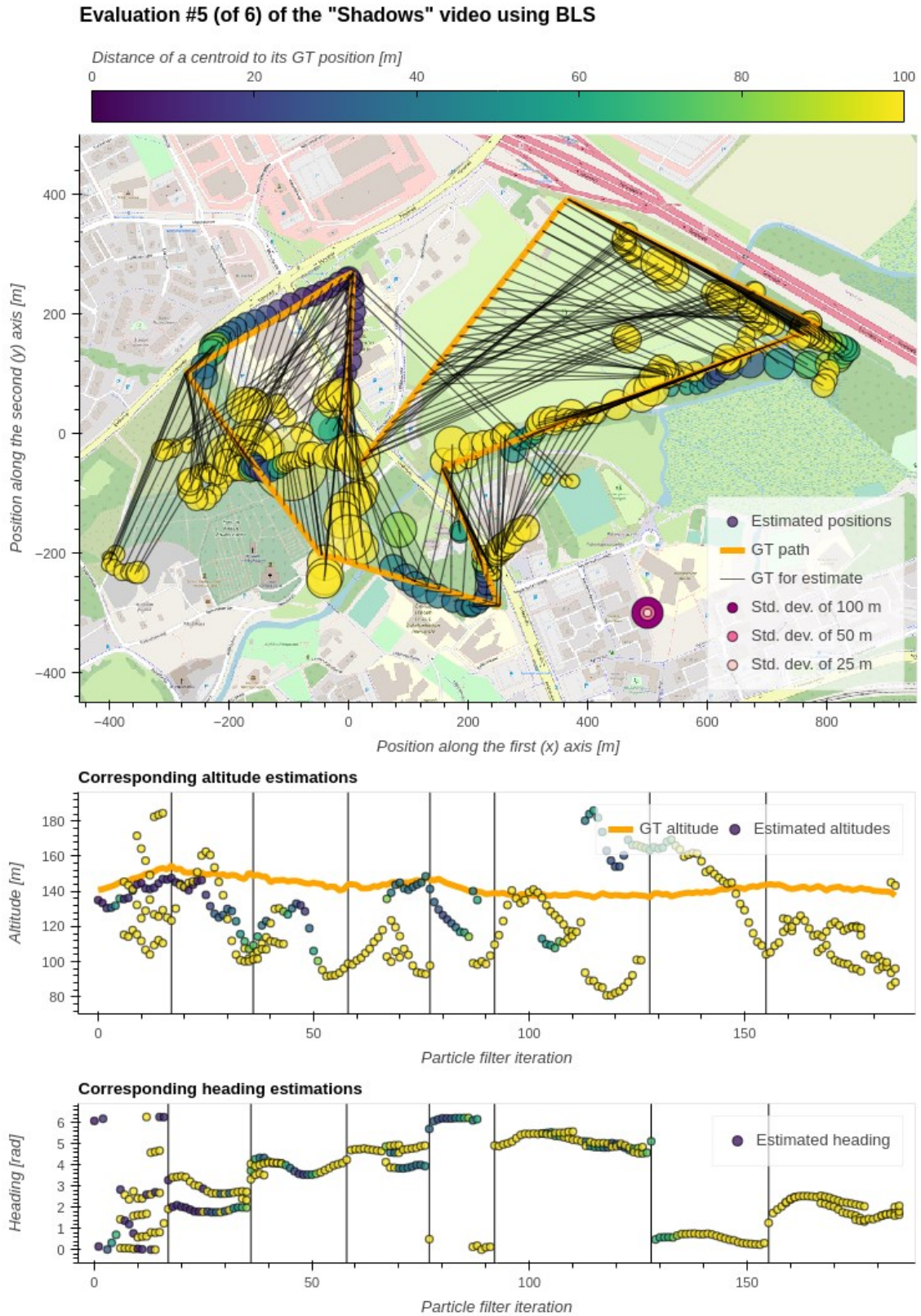

Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

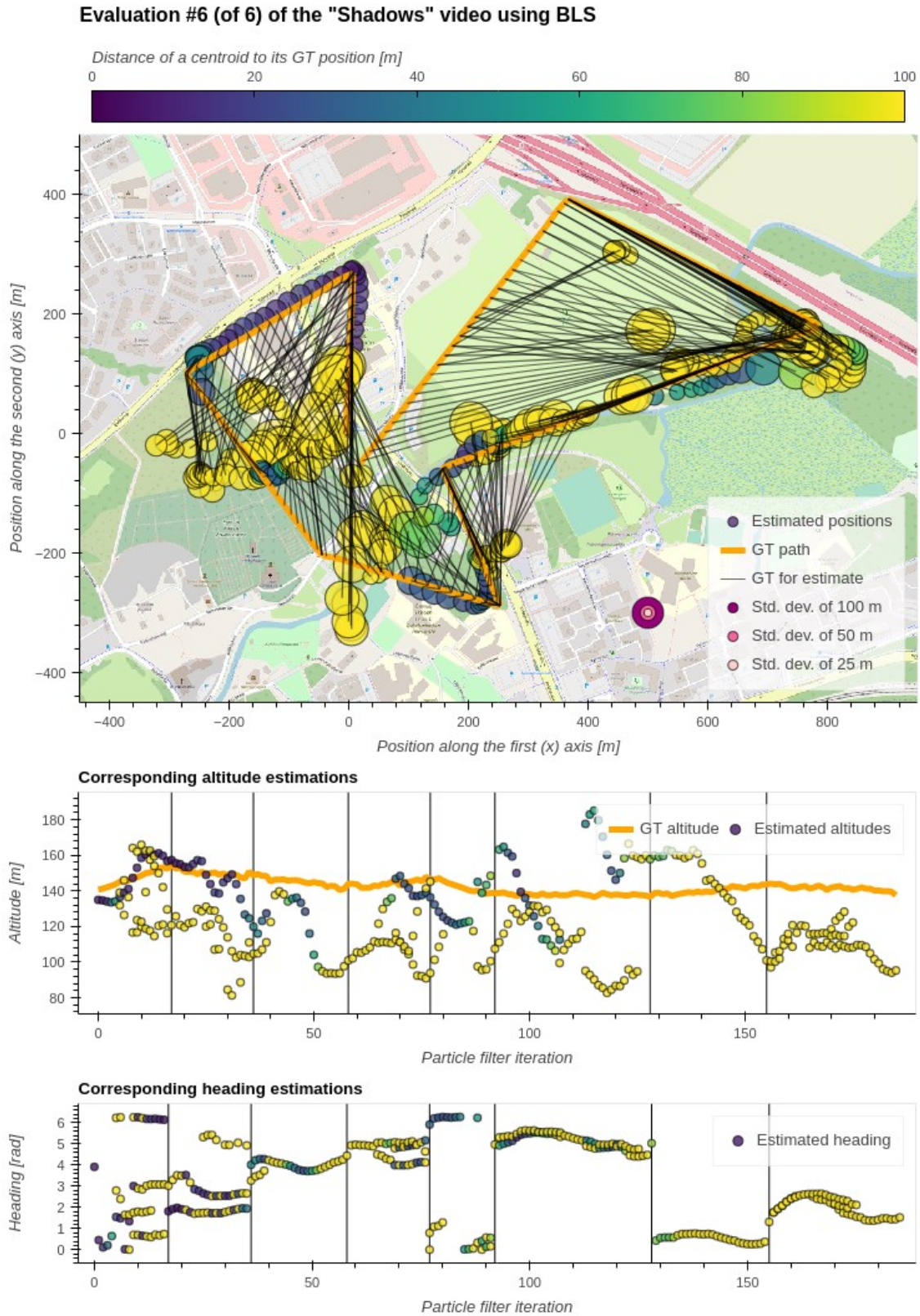

Evaluation #1 (of 6) of the "Shadows" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

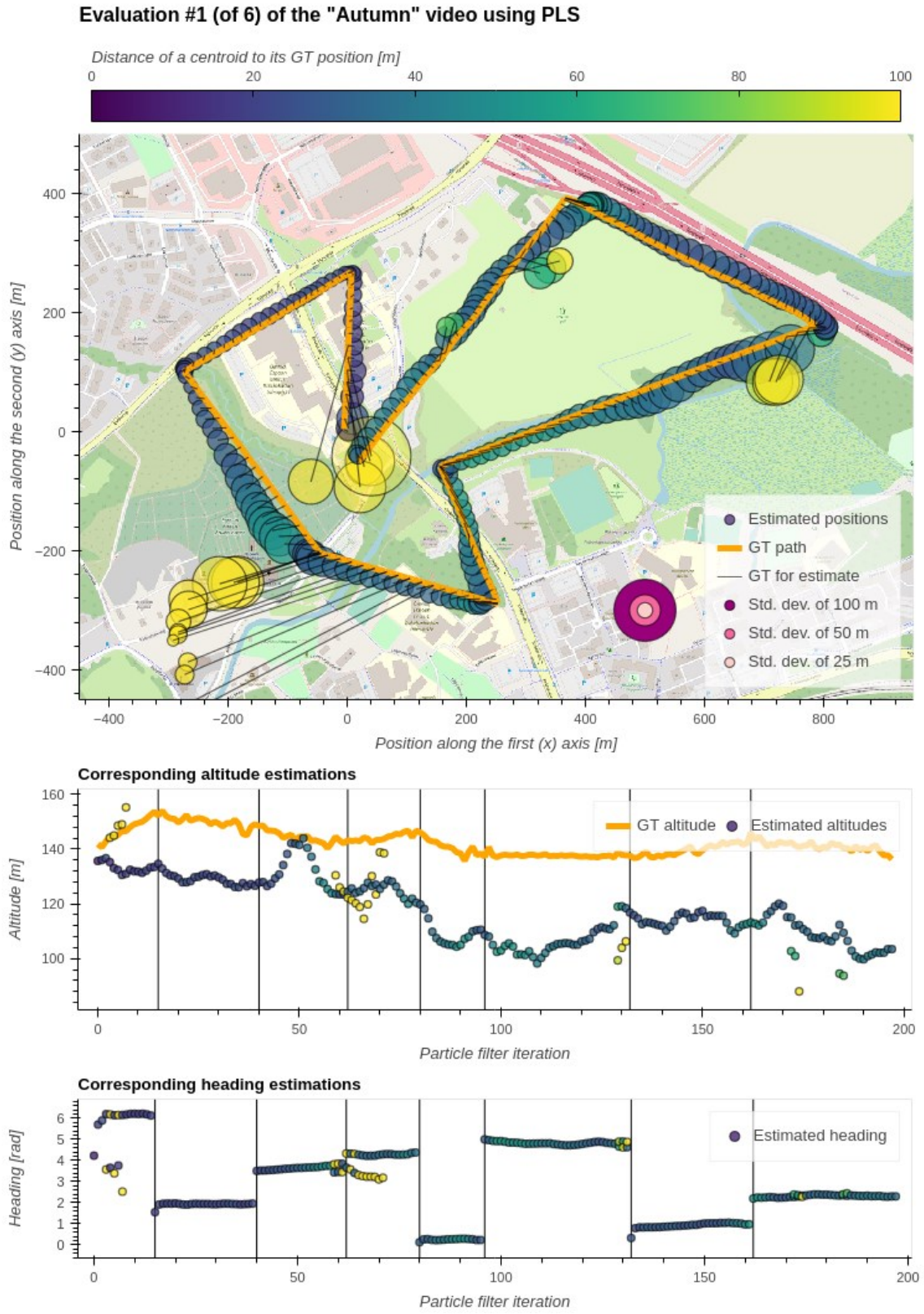

Evaluation #2 (of 6) of the "Shadows" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

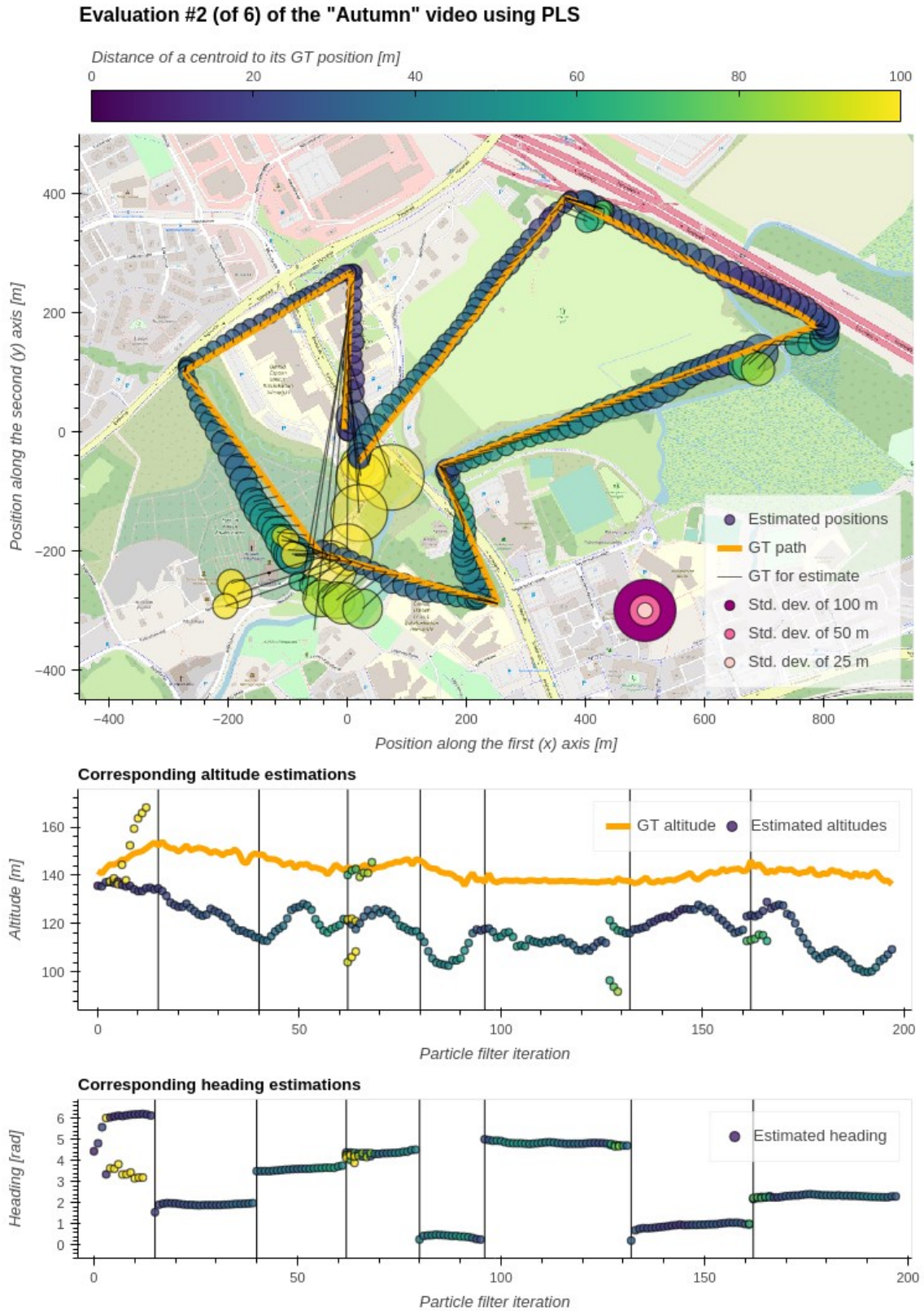

Evaluation #3 (of 6) of the "Shadows" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

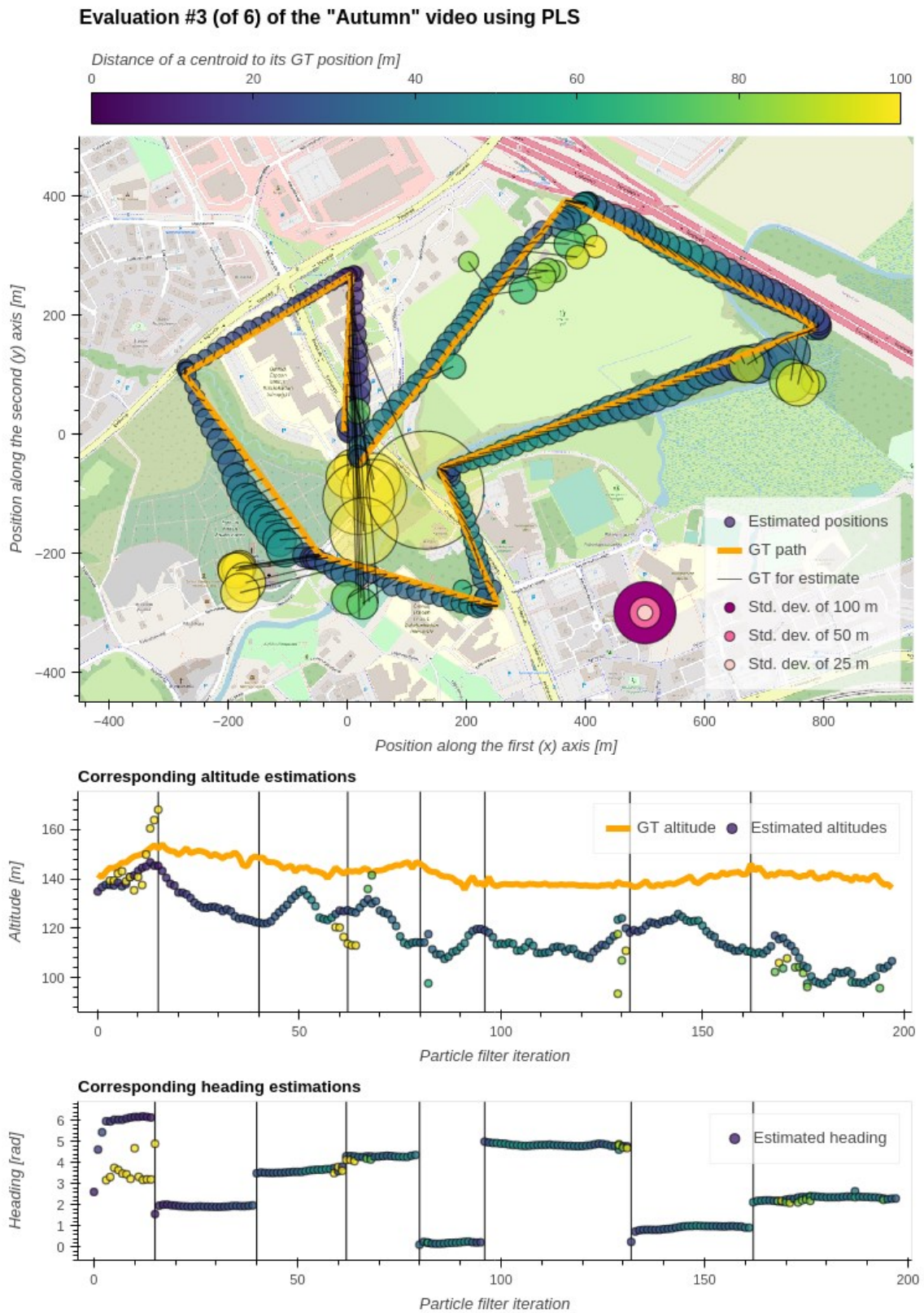

Evaluation #4 (of 6) of the "Shadows" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

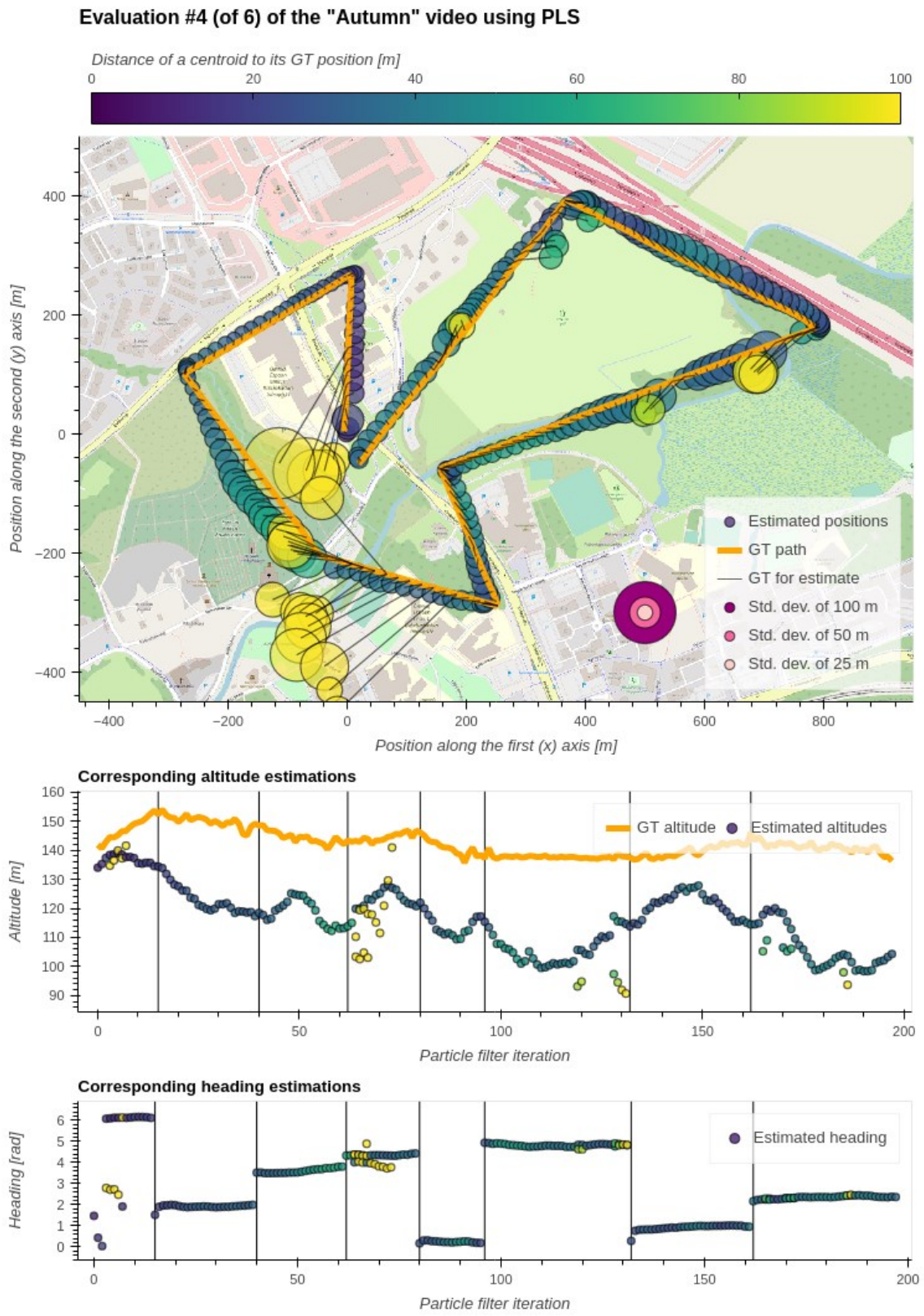

Evaluation #5 (of 6) of the "Shadows" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

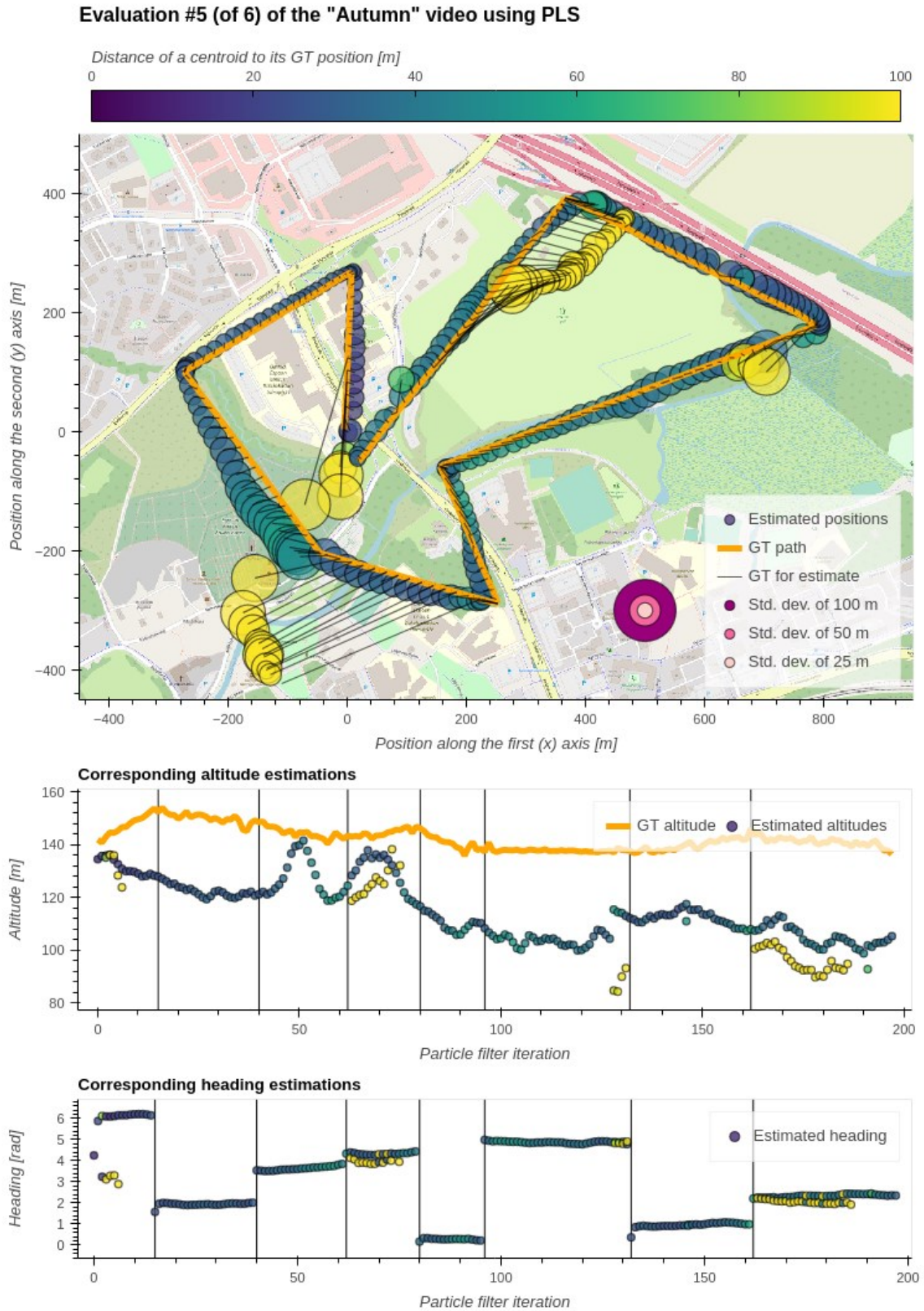

Evaluation #6 (of 6) of the "Shadows" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

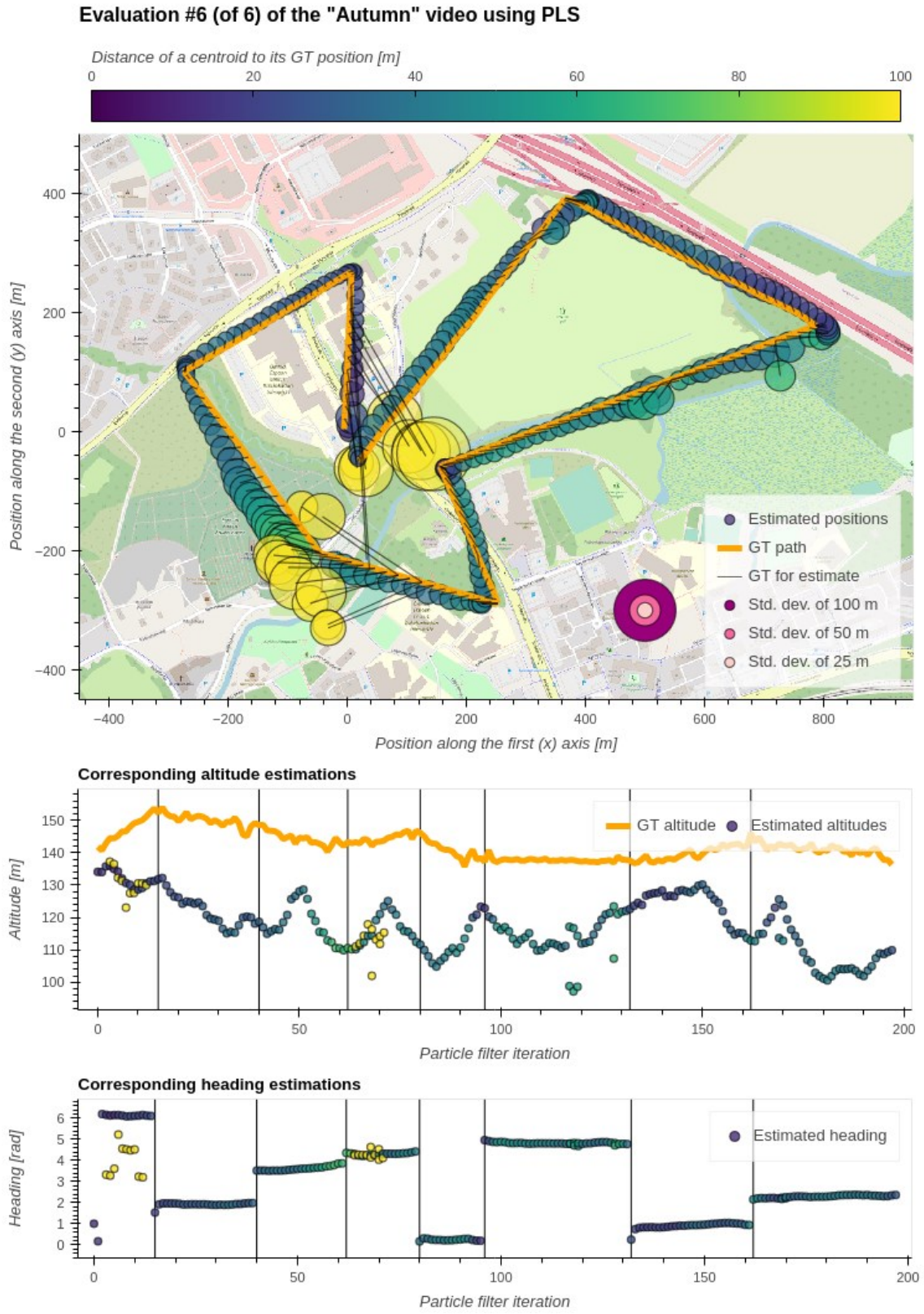

Evaluation #1 (of 6) of the "Autumn" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

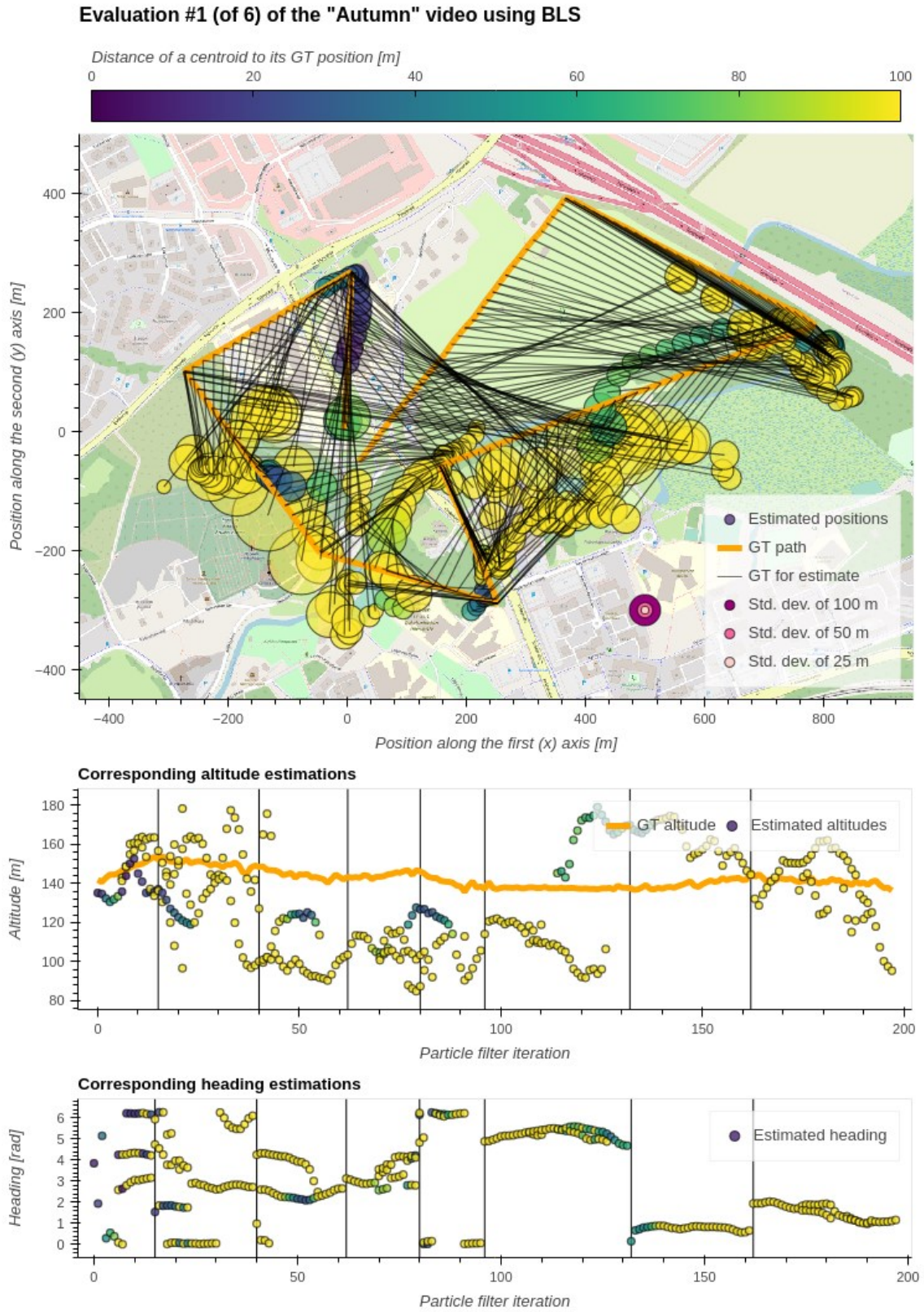

Evaluation #2 (of 6) of the "Autumn" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
GT altitude
Estimated altitudes
Altitude [m]
Particle filter iteration
Corresponding heading estimations
Estimated heading
Heading [rad]
Particle filter iteration

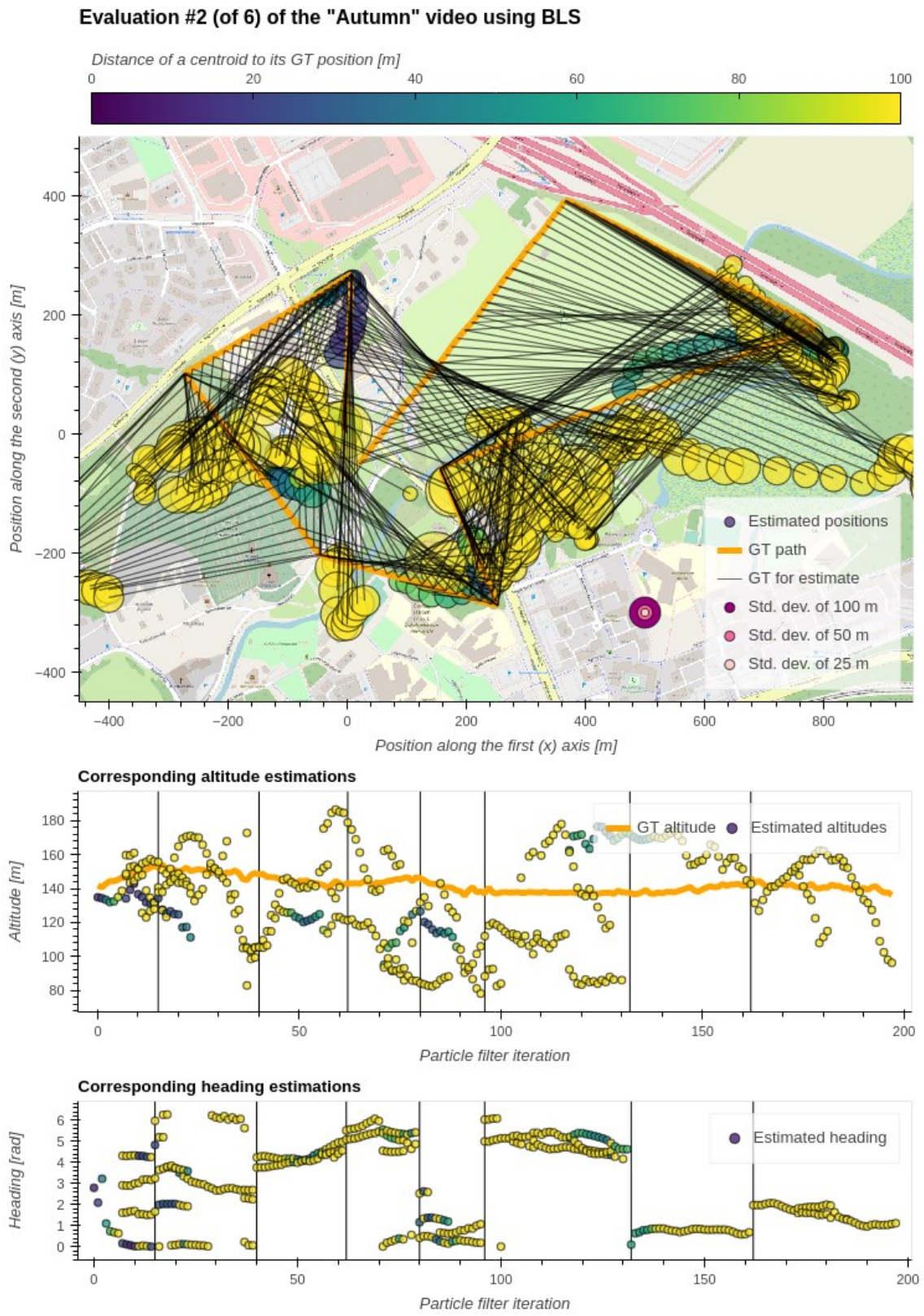

Evaluation #3 (of 6) of the "Autumn" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

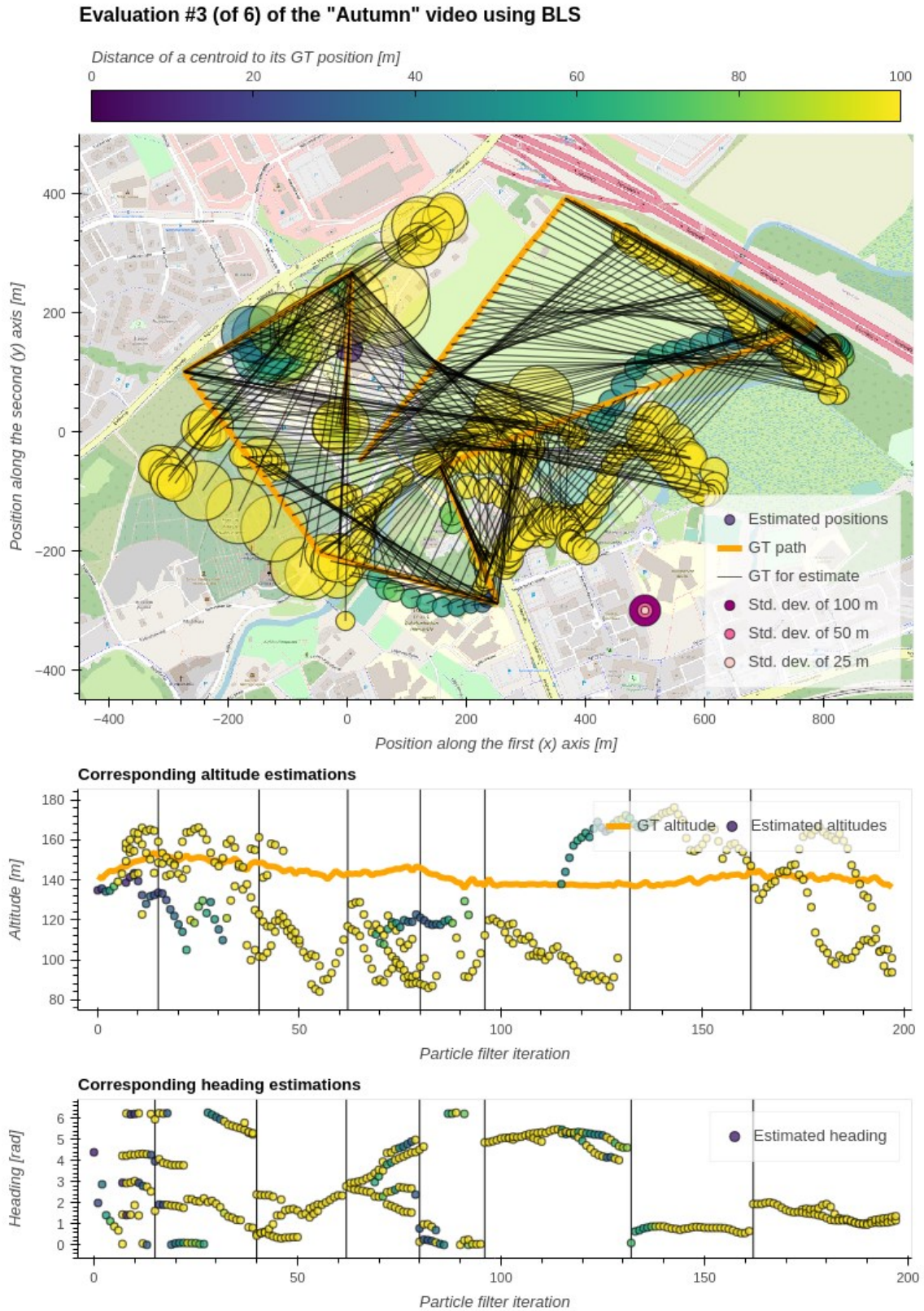

Evaluation #4 (of 6) of the "Autumn" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

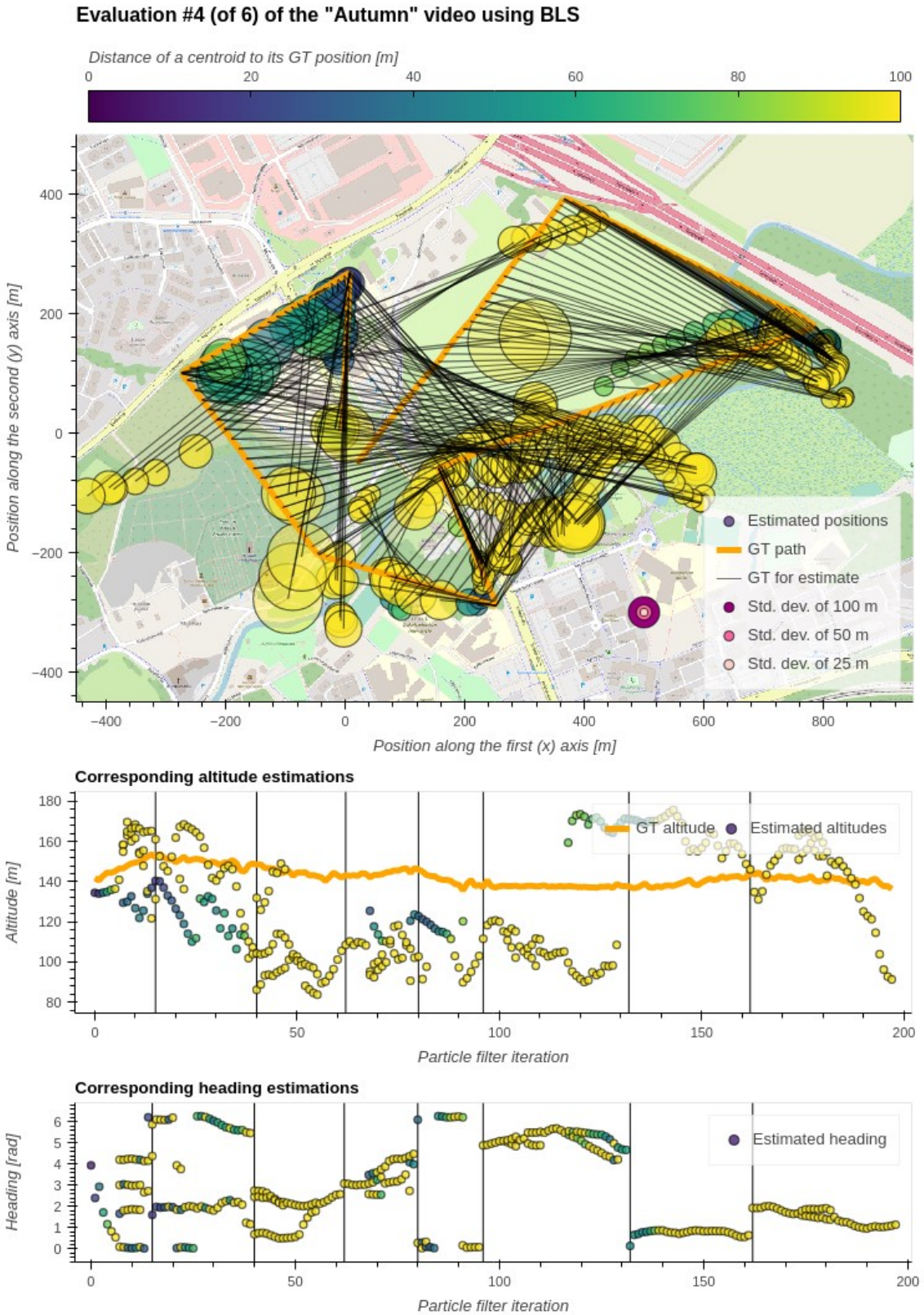

Evaluation #5 (of 6) of the "Autumn" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

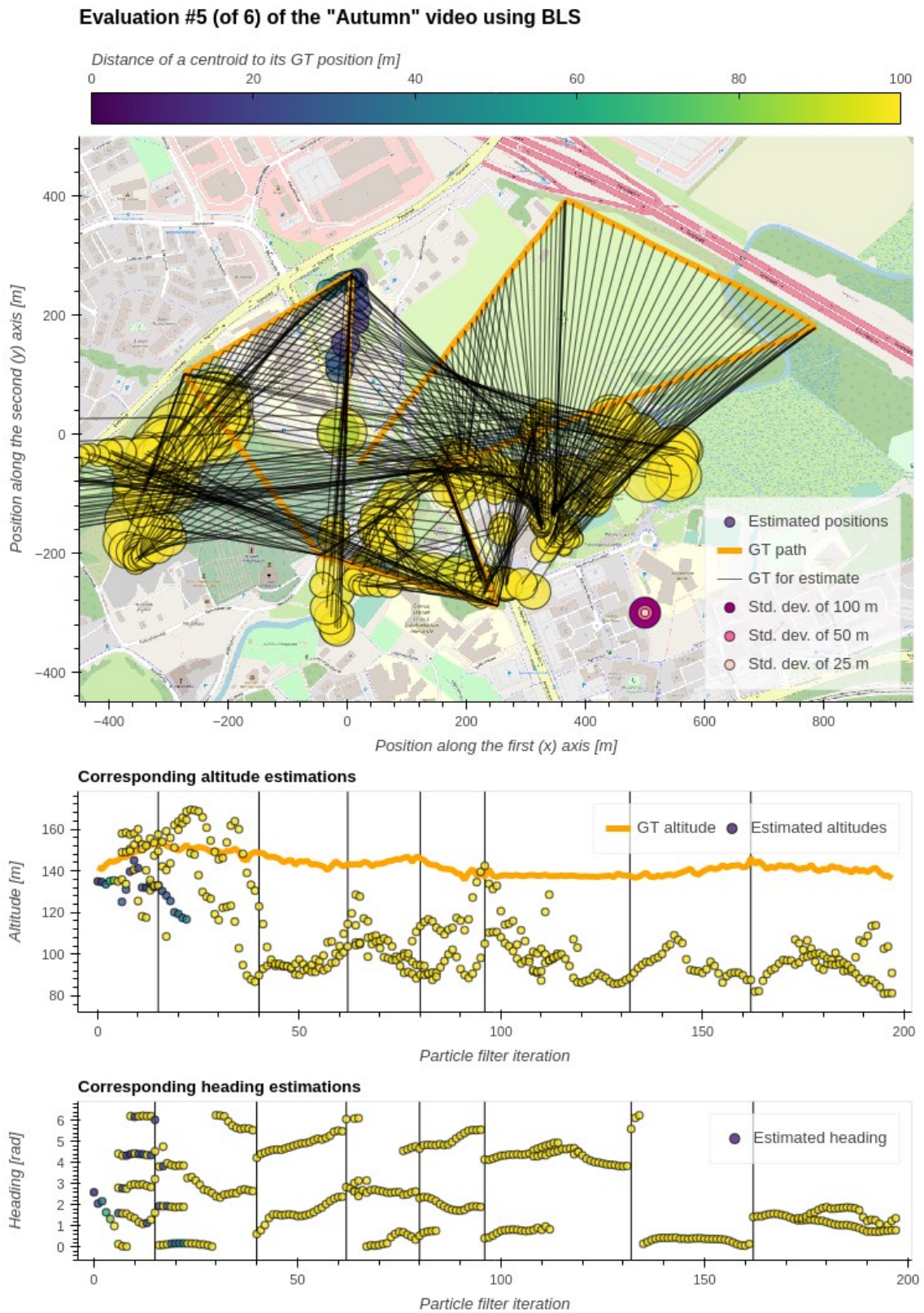

Evaluation #6 (of 6) of the "Autumn" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

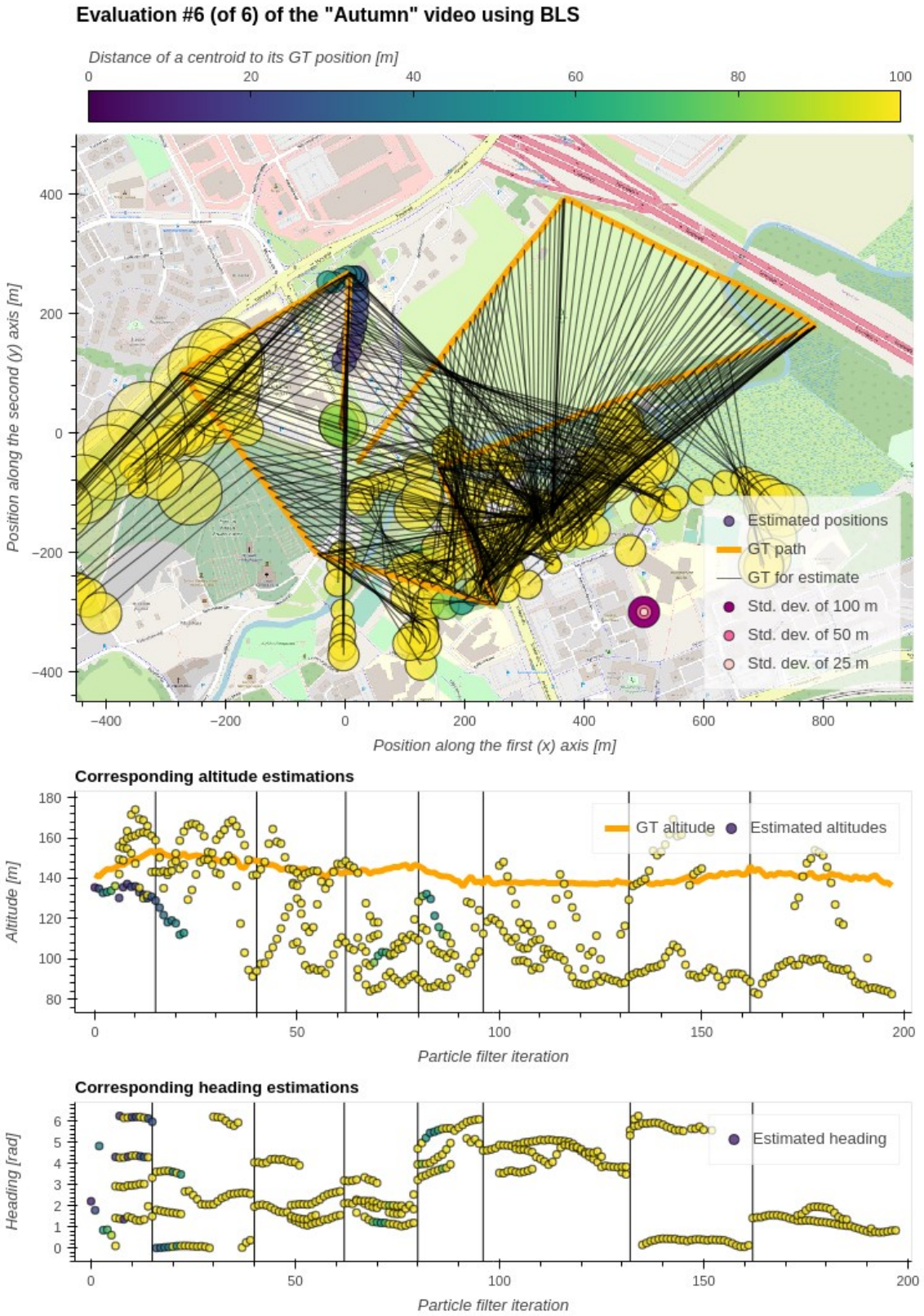

Evaluation #1 (of 6) of the "Autumn" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

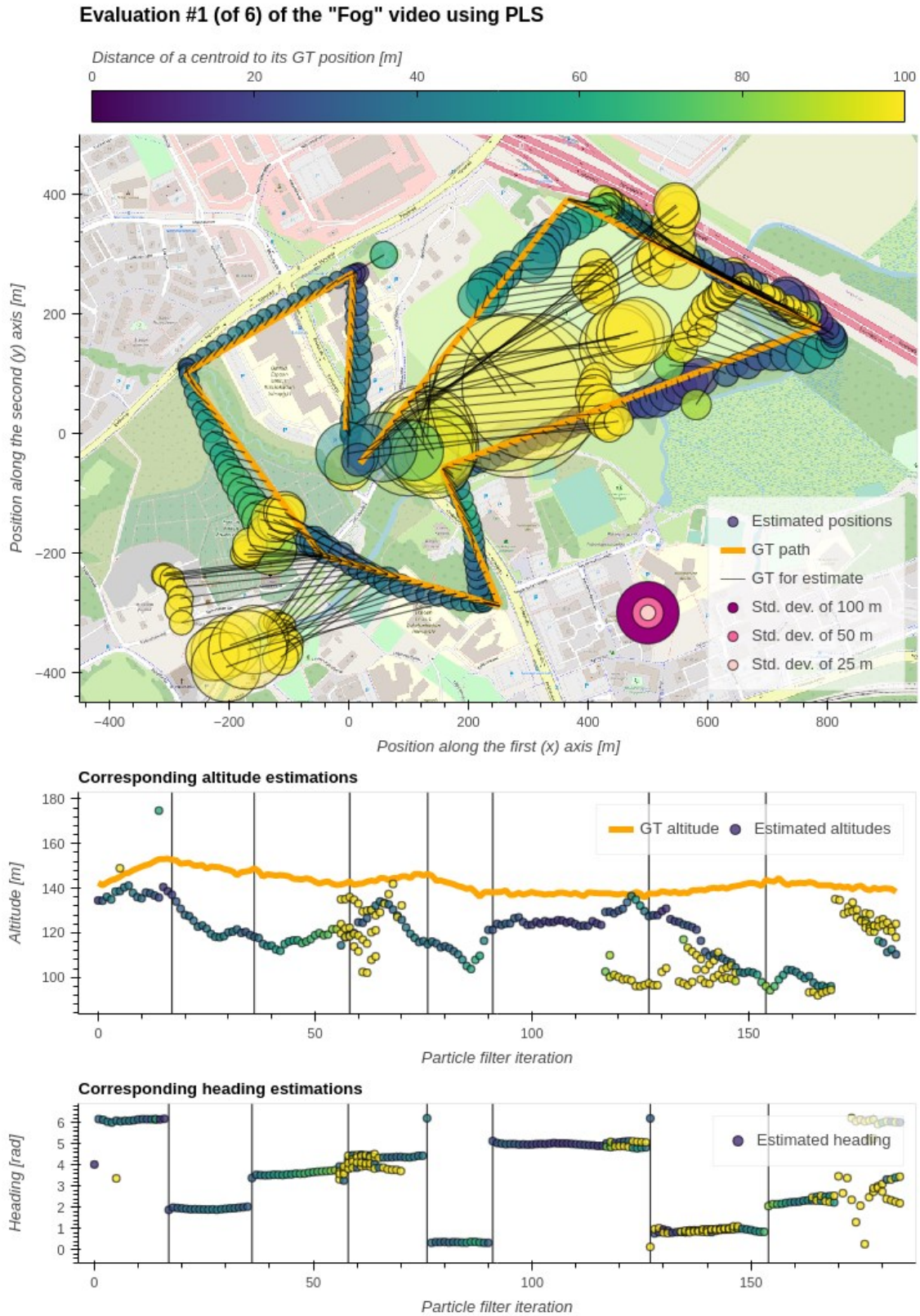

Evaluation #2 (of 6) of the "Autumn" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

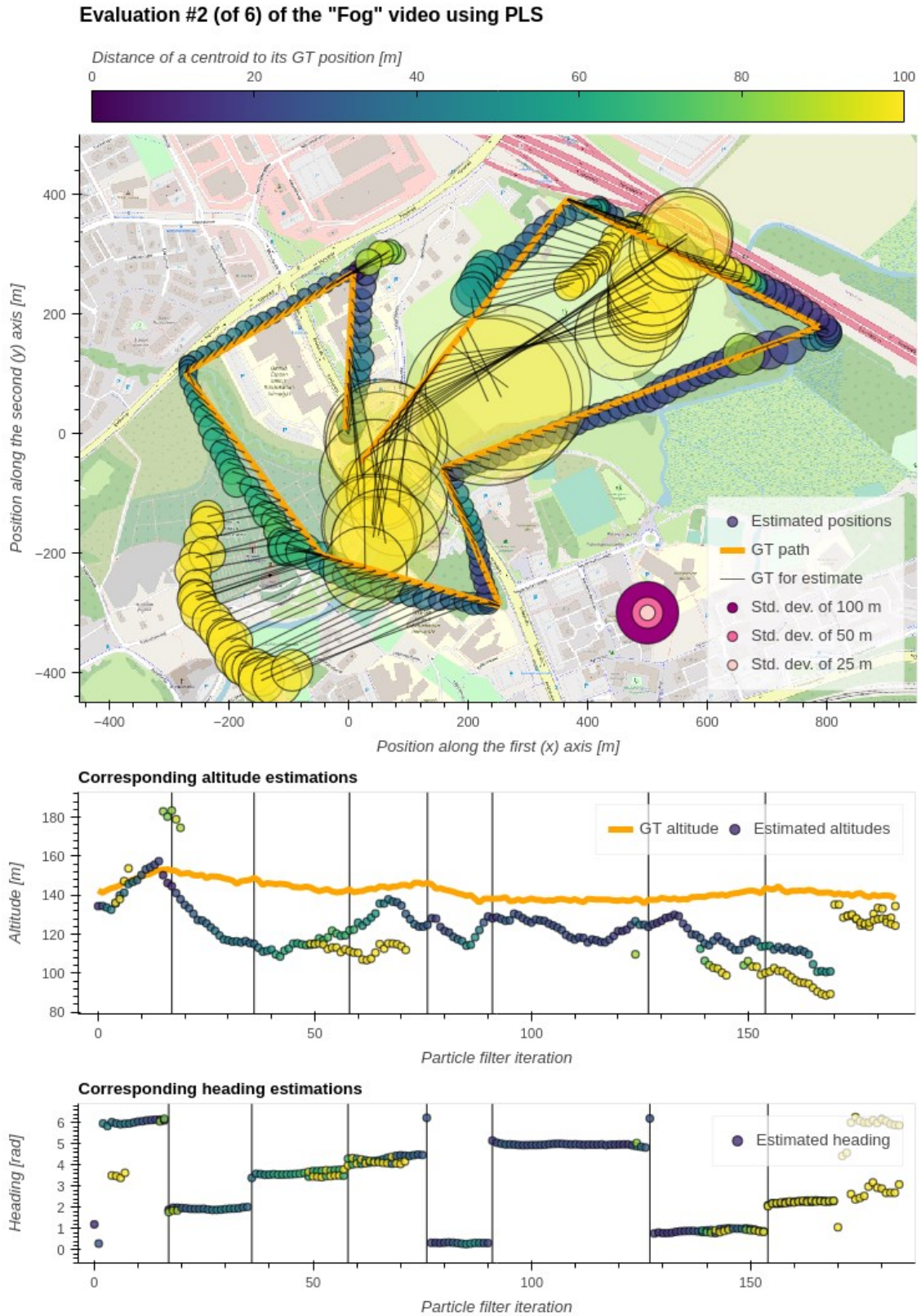

Evaluation #3 (of 6) of the "Autumn" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

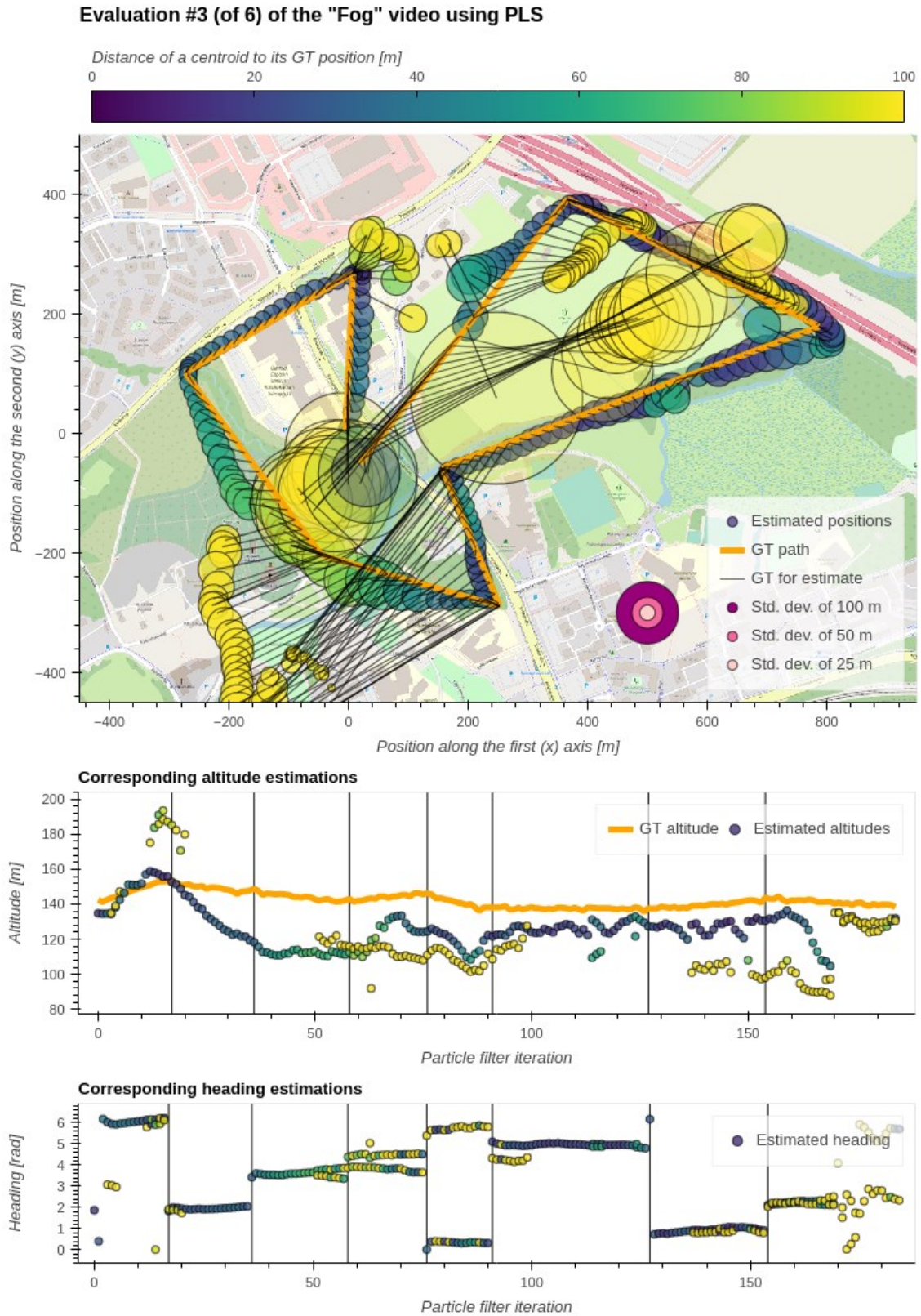

Evaluation #4 (of 6) of the "Autumn" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

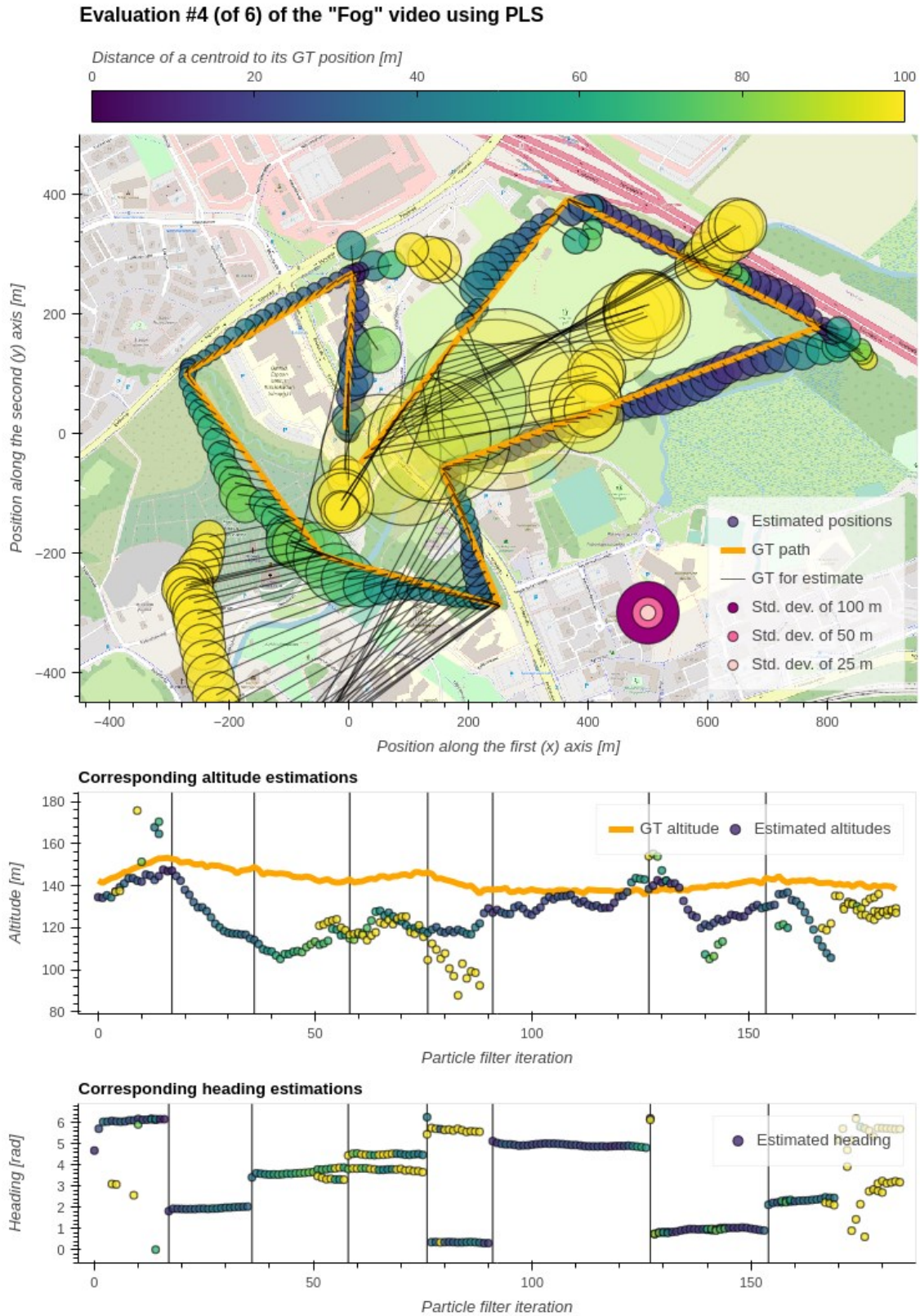

Evaluation #5 (of 6) of the "Autumn" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

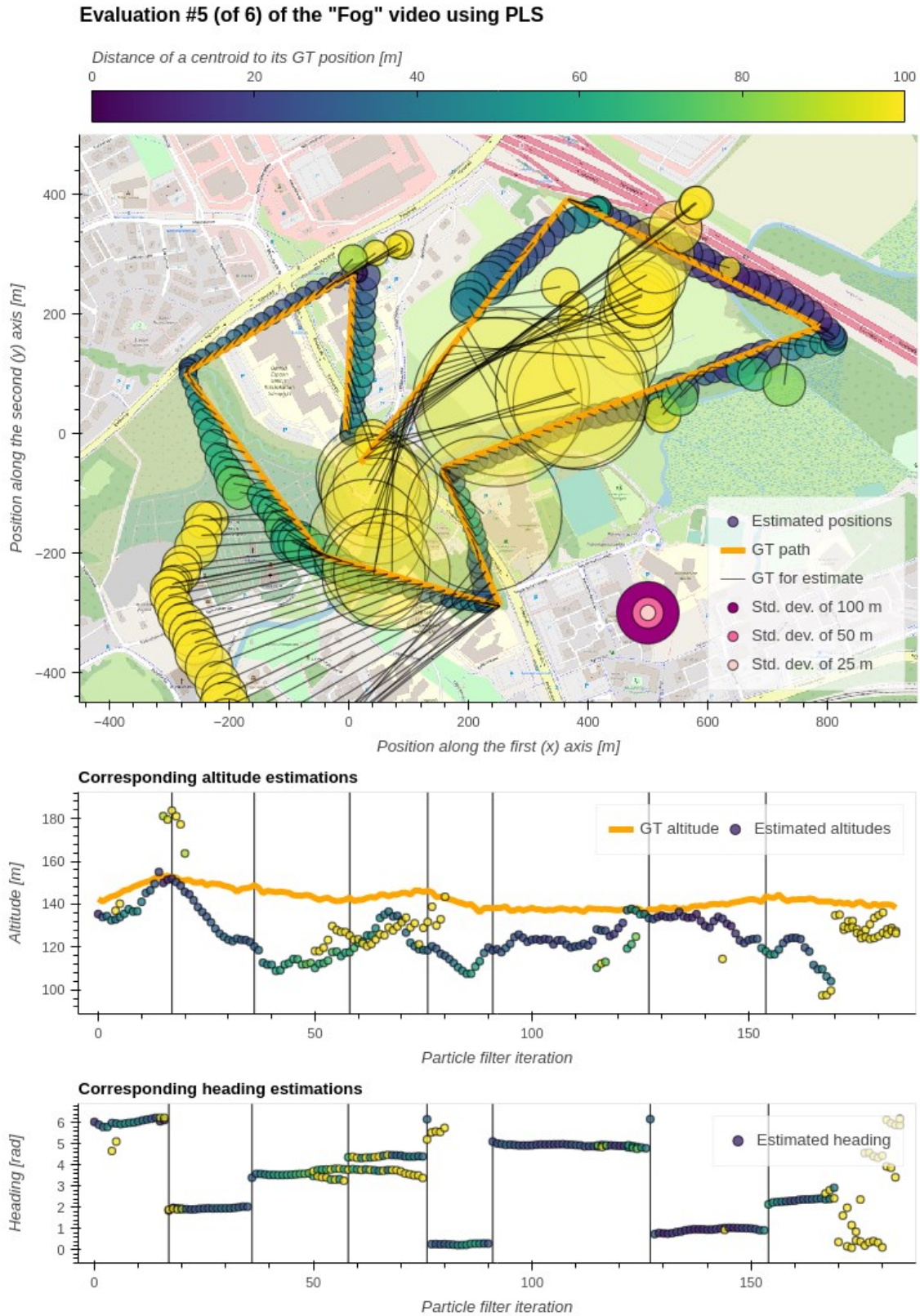

Evaluation #6 (of 6) of the "Autumn" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

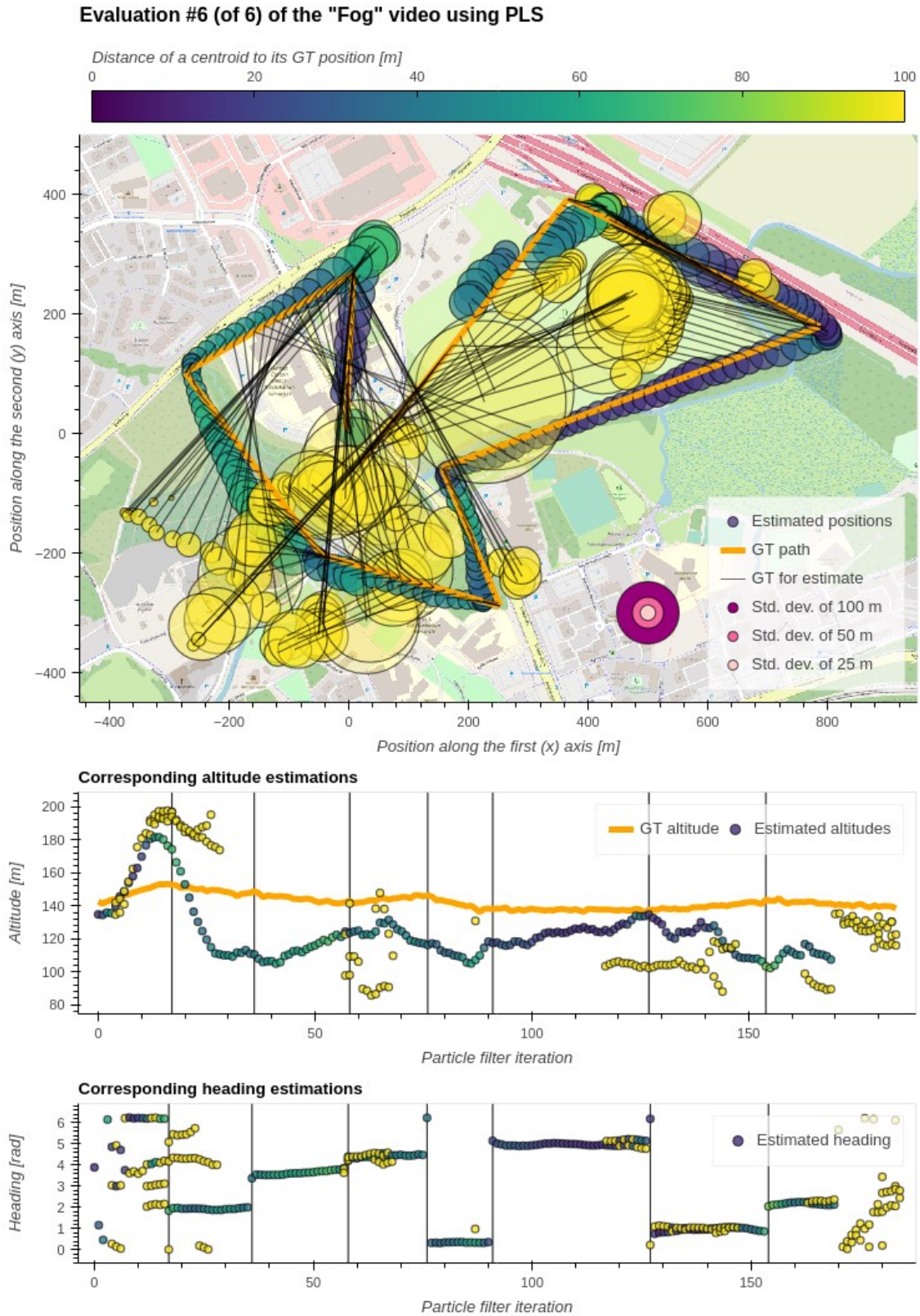

Evaluation #1 (of 6) of the "Fog" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

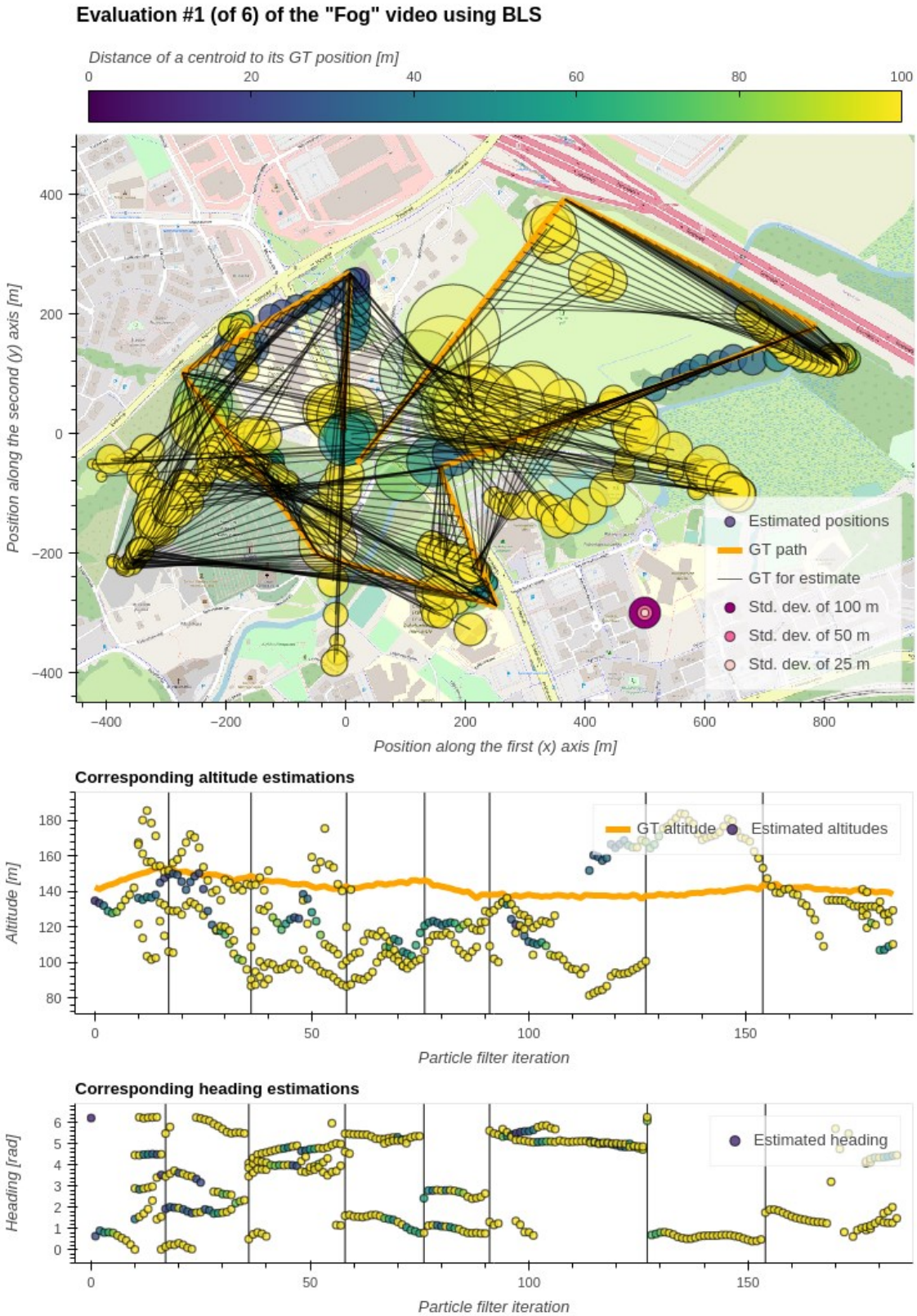

Evaluation #2 (of 6) of the "Fog" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

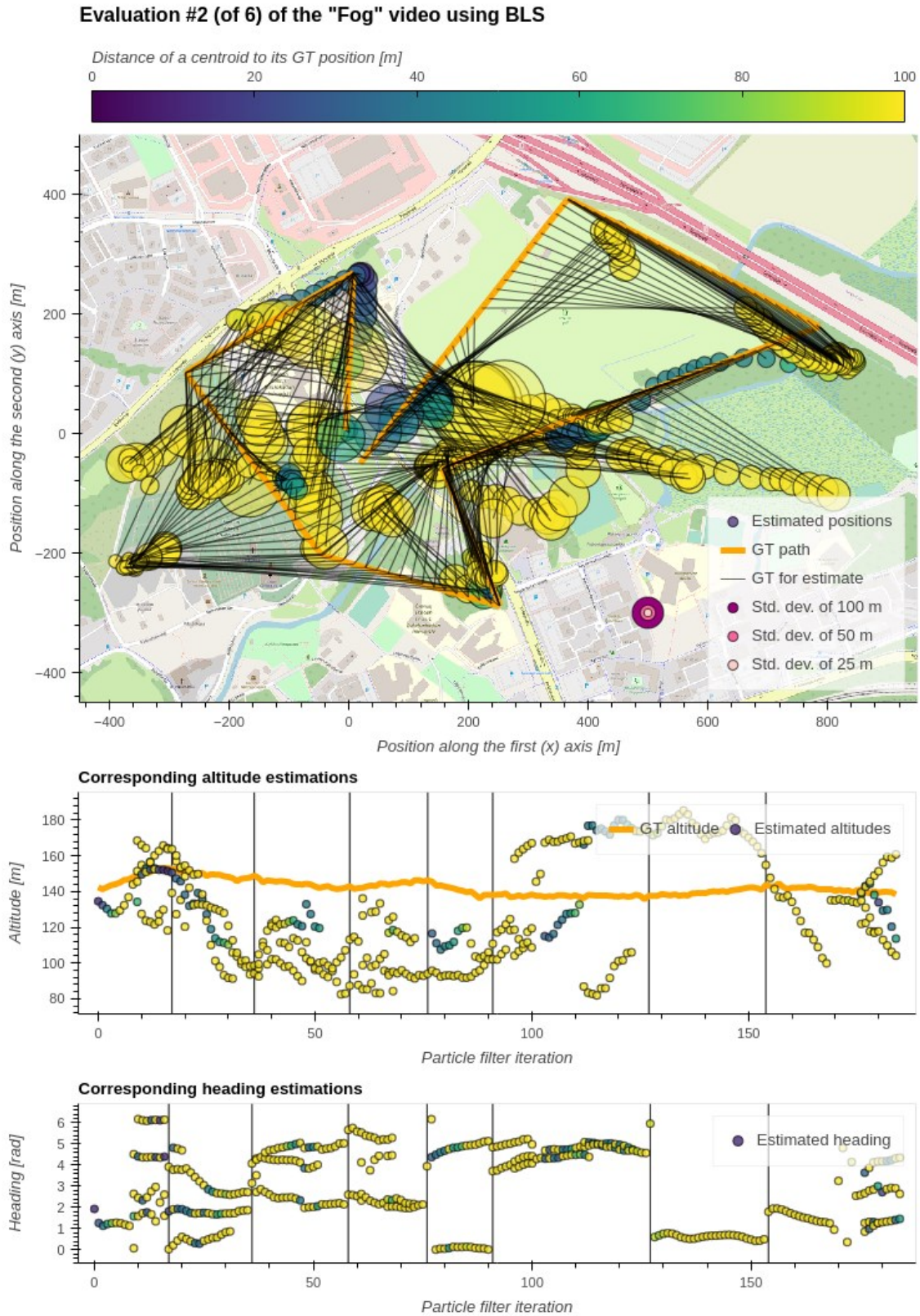

Evaluation #3 (of 6) of the "Fog" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
200
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

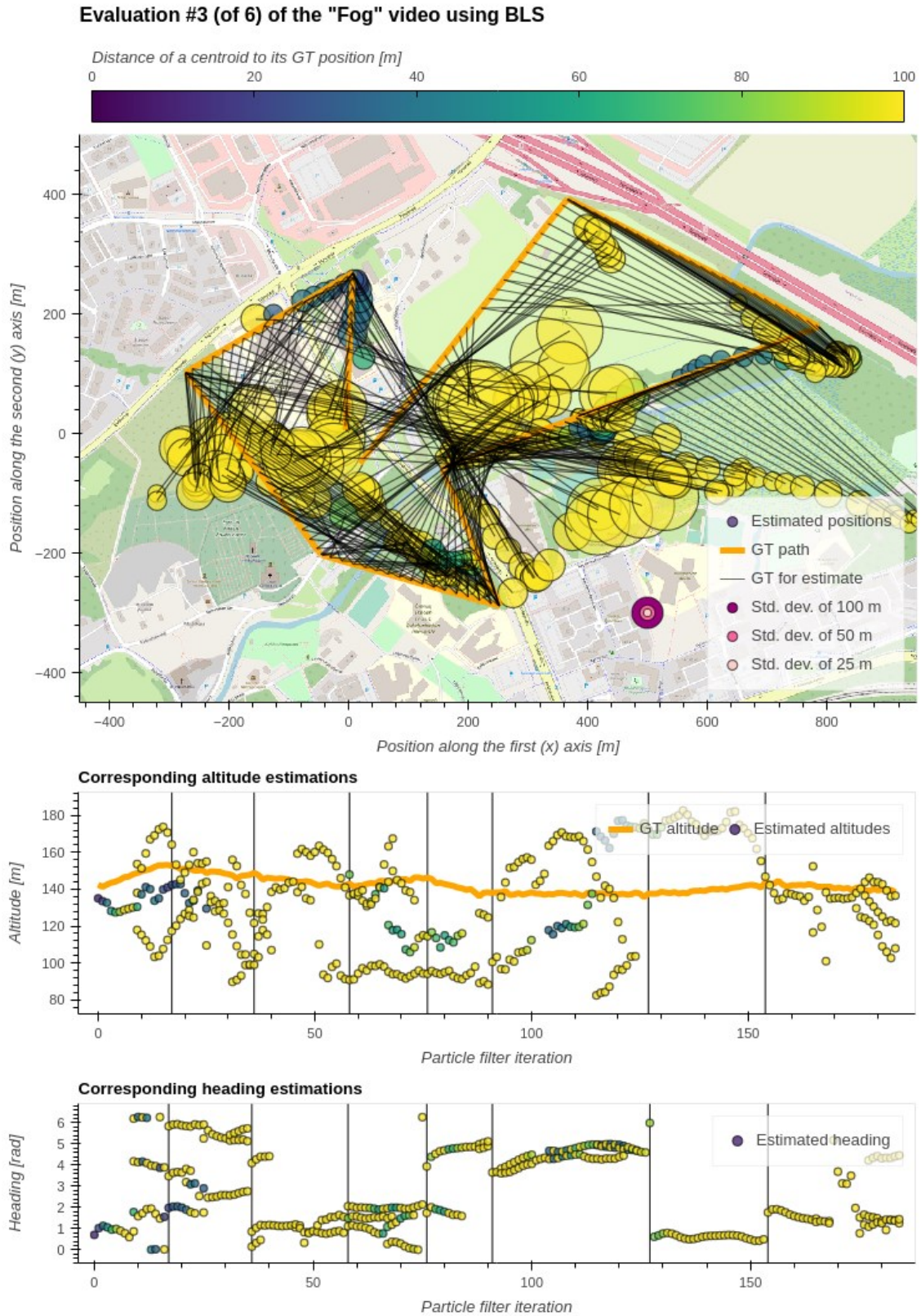

Evaluation #4 (of 6) of the "Fog" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

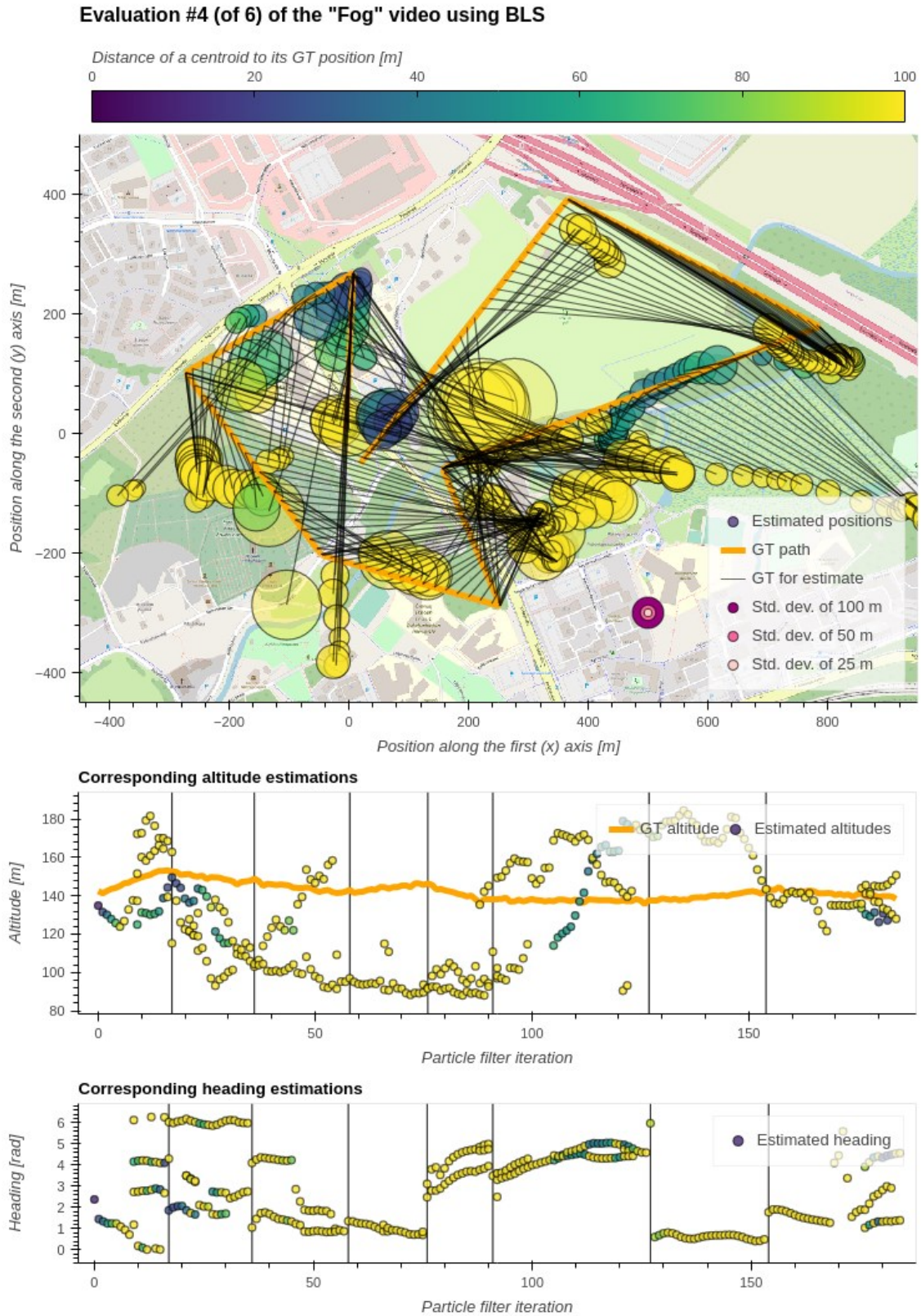

Evaluation #5 (of 6) of the "Fog" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

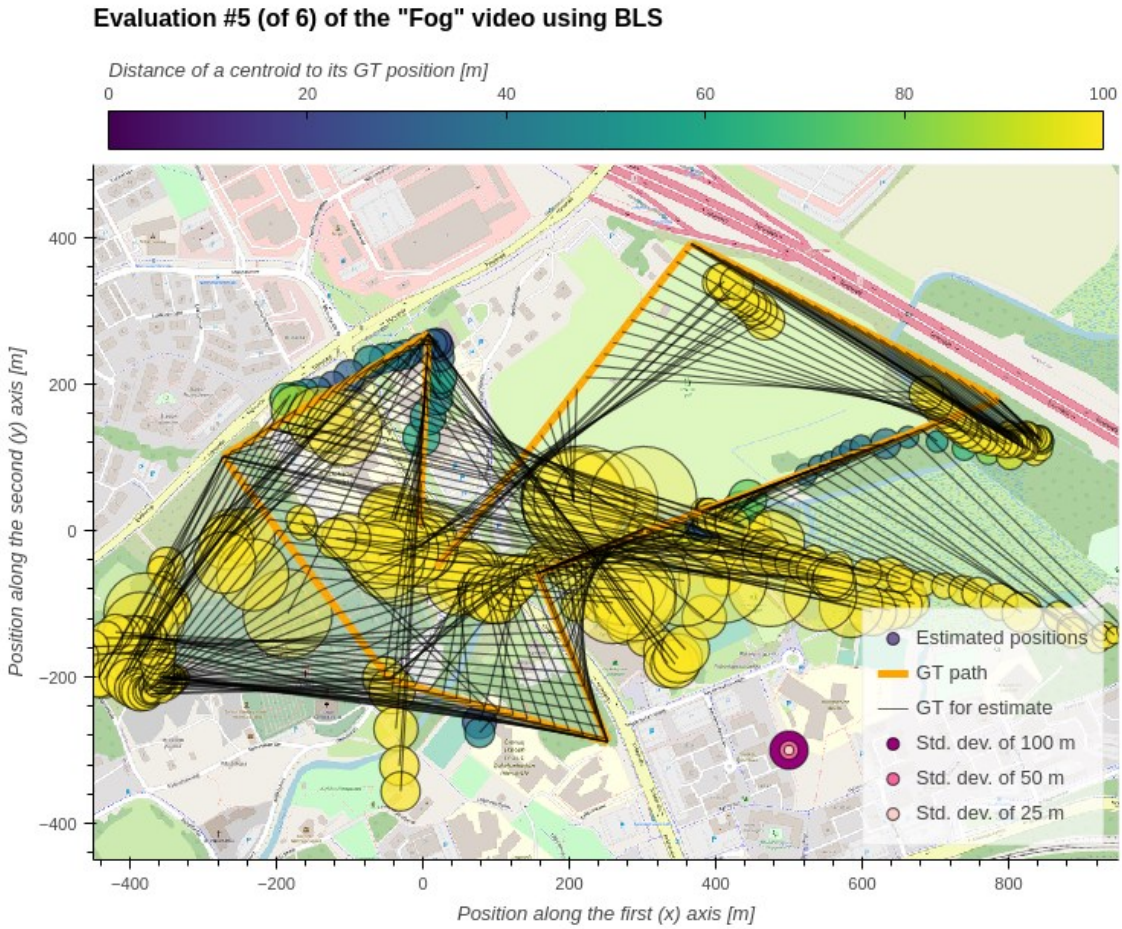

Evaluation #6 (of 6) of the "Fog" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

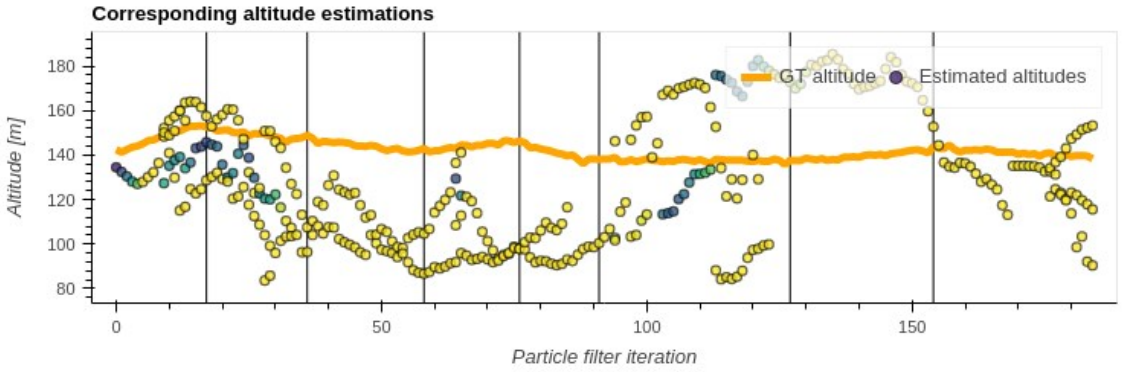

Evaluation #1 (of 6) of the "Fog" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

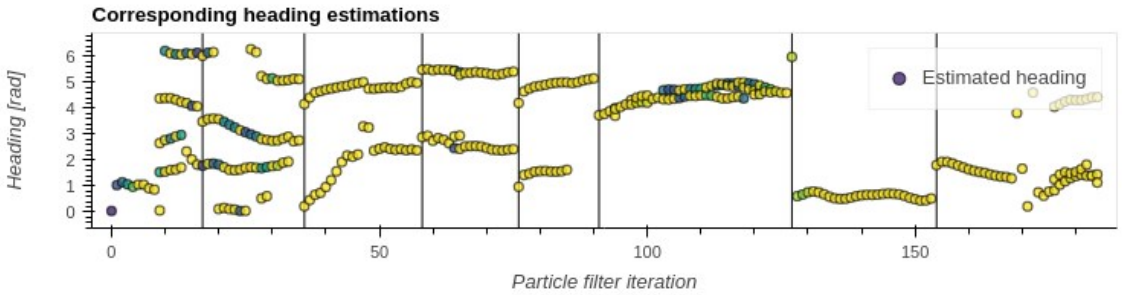

Evaluation #2 (of 6) of the "Fog" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

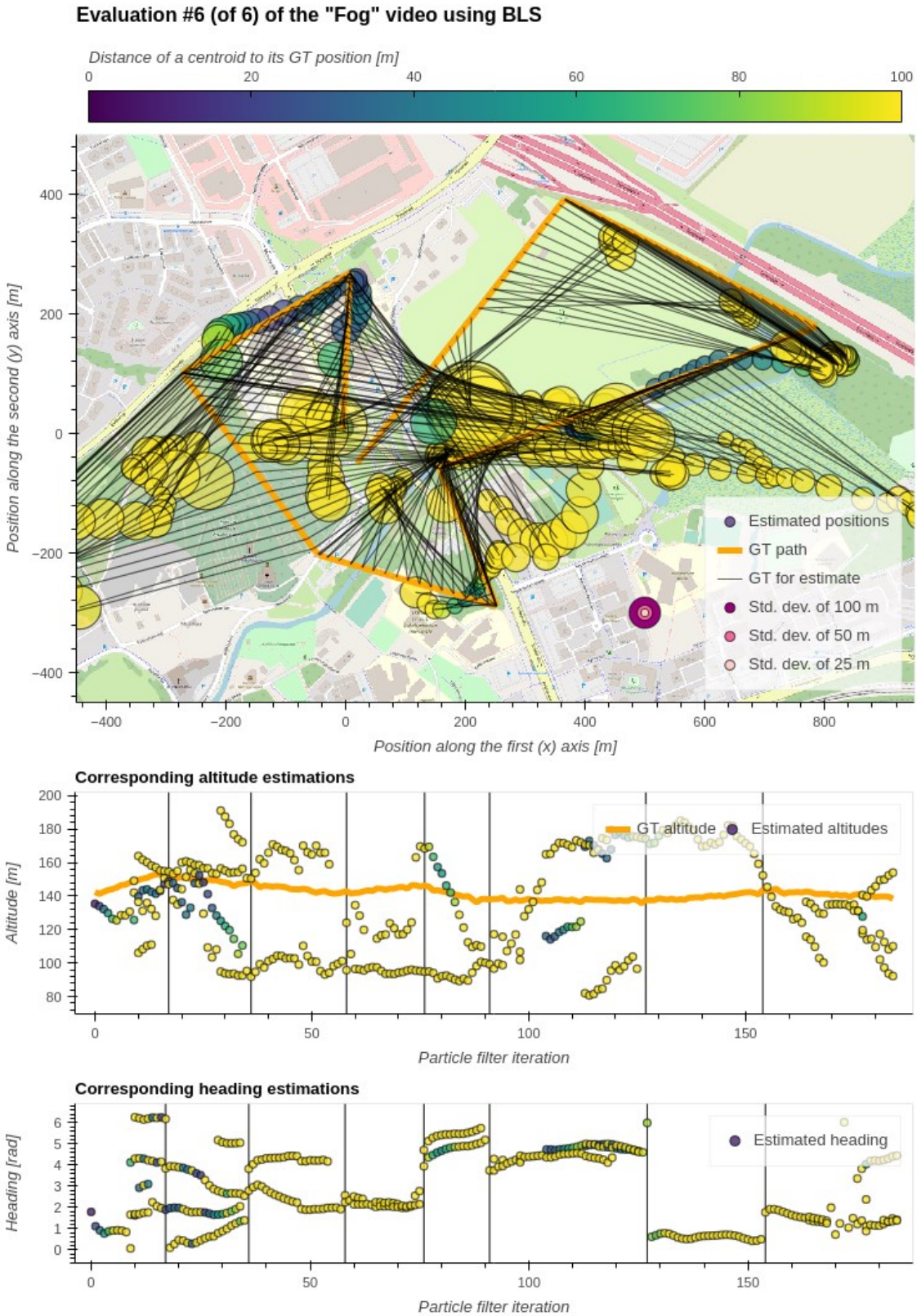

Evaluation #3 (of 6) of the "Fog" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

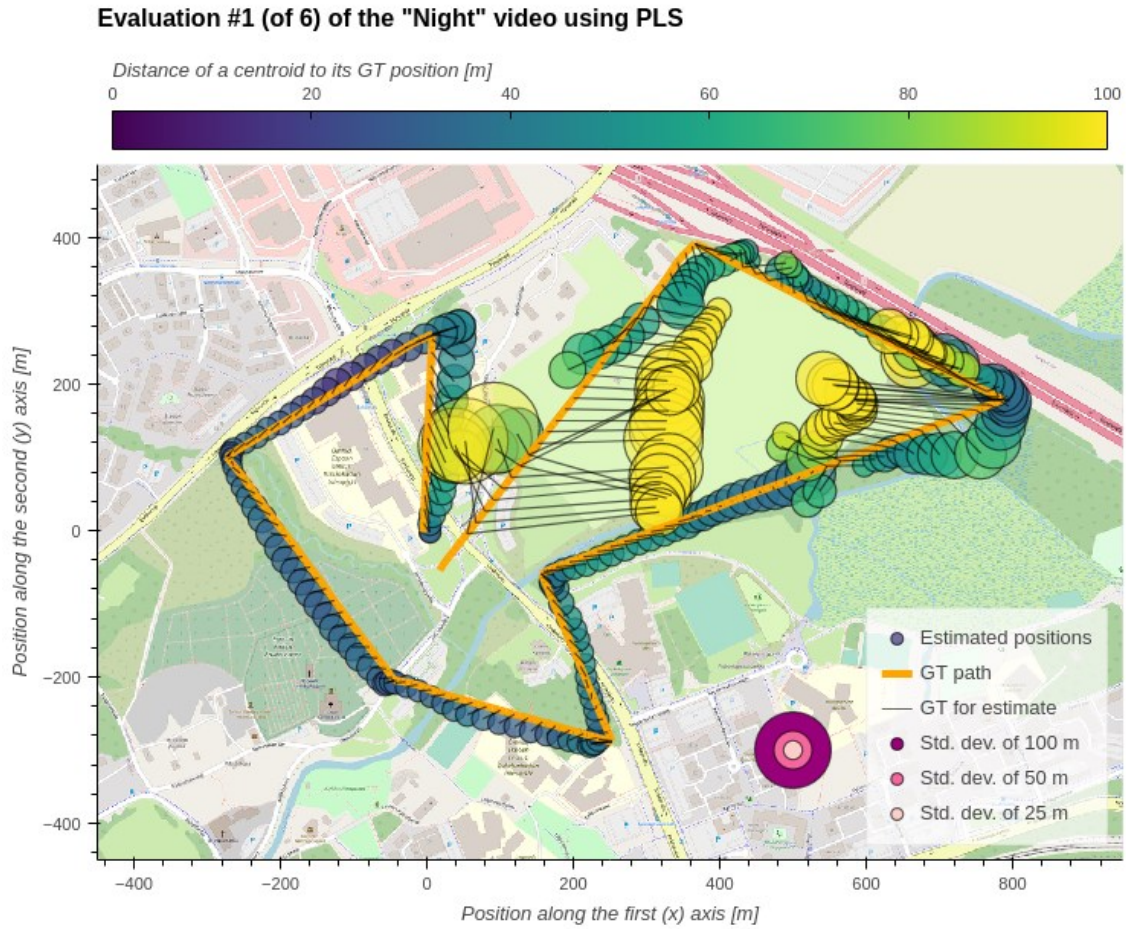

Evaluation #4 (of 6) of the "Fog" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

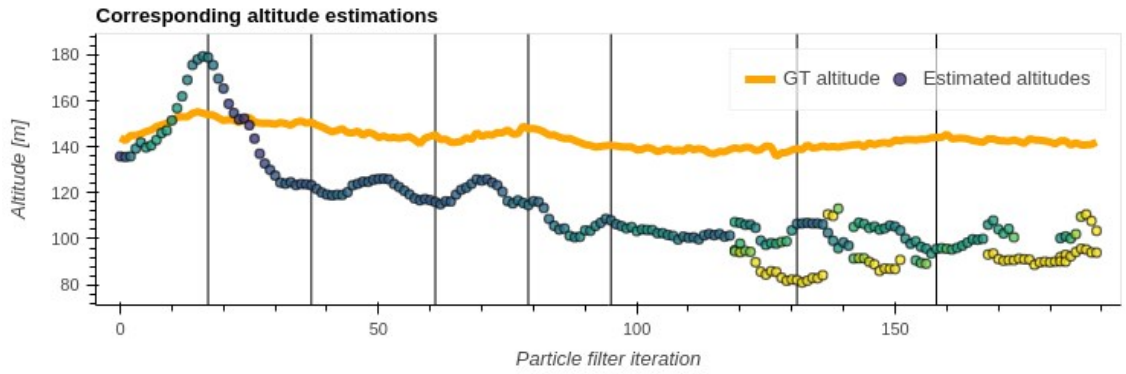

Evaluation #5 (of 6) of the "Fog" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m

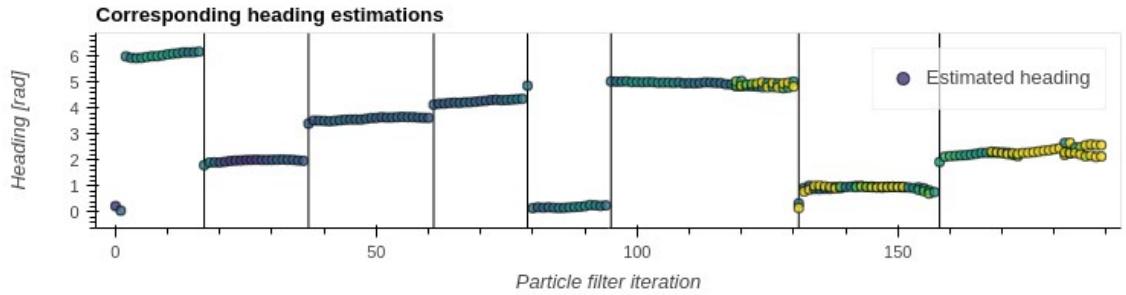

Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes

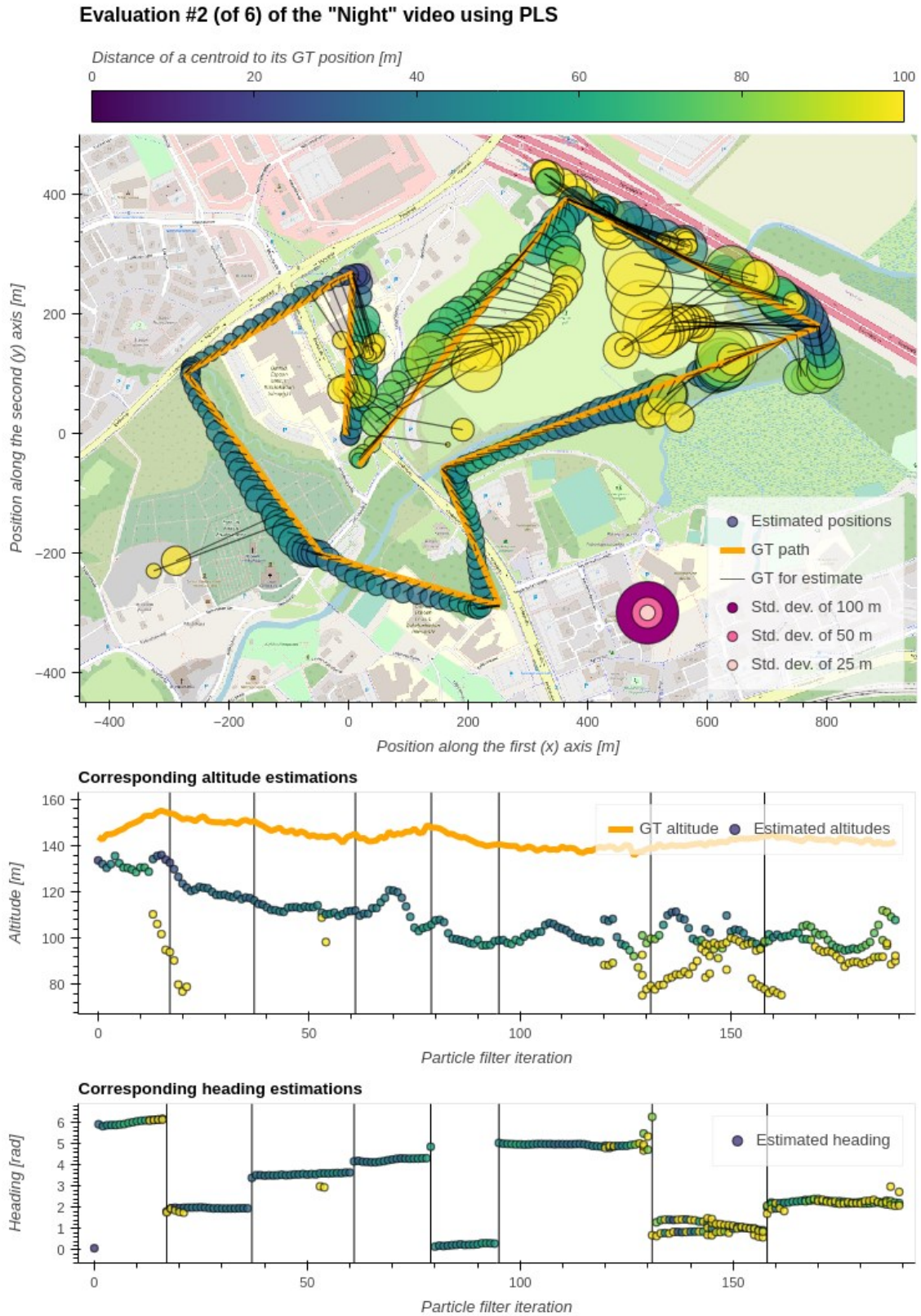

Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

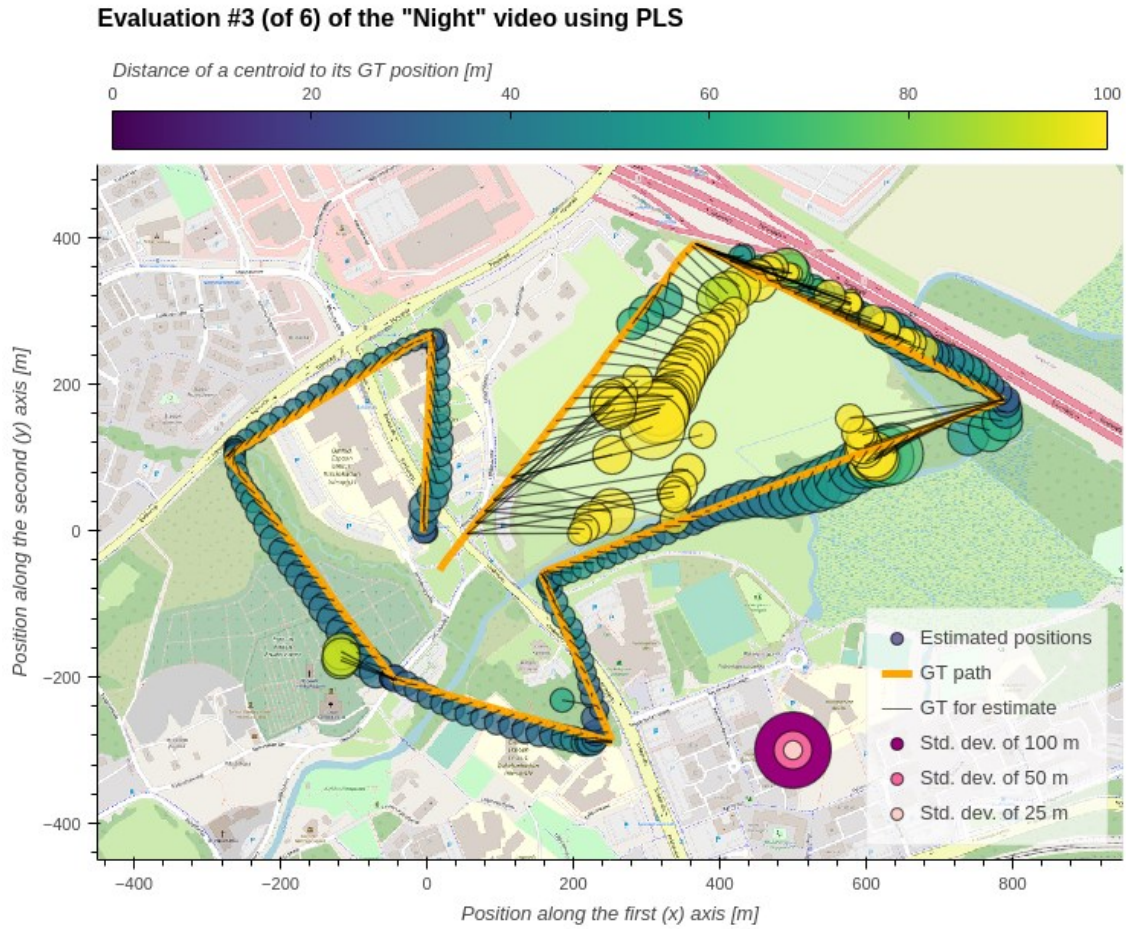

Evaluation #6 (of 6) of the "Fog" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

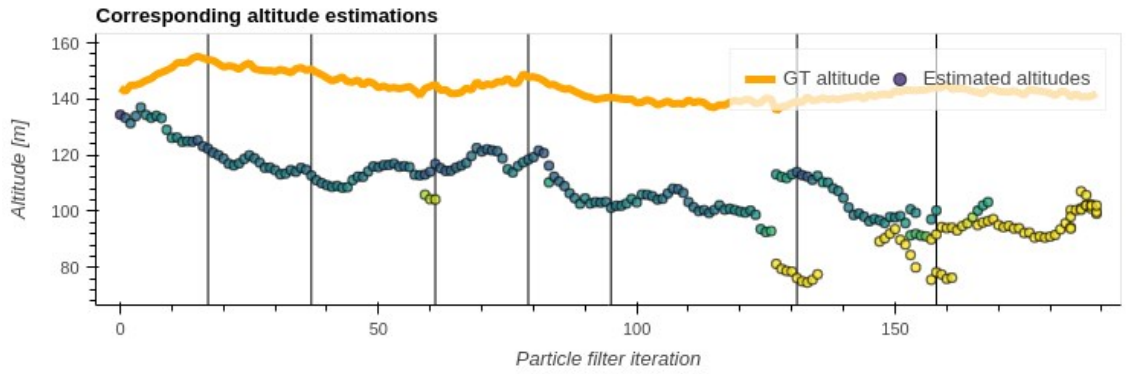

Evaluation #1 (of 6) of the "Night" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m

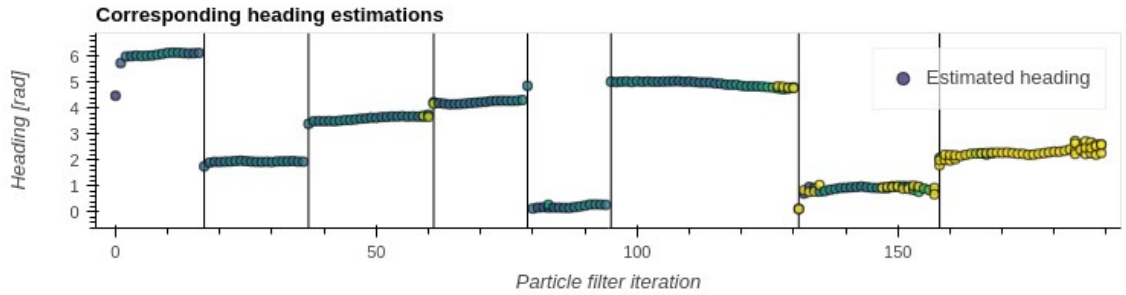

Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes

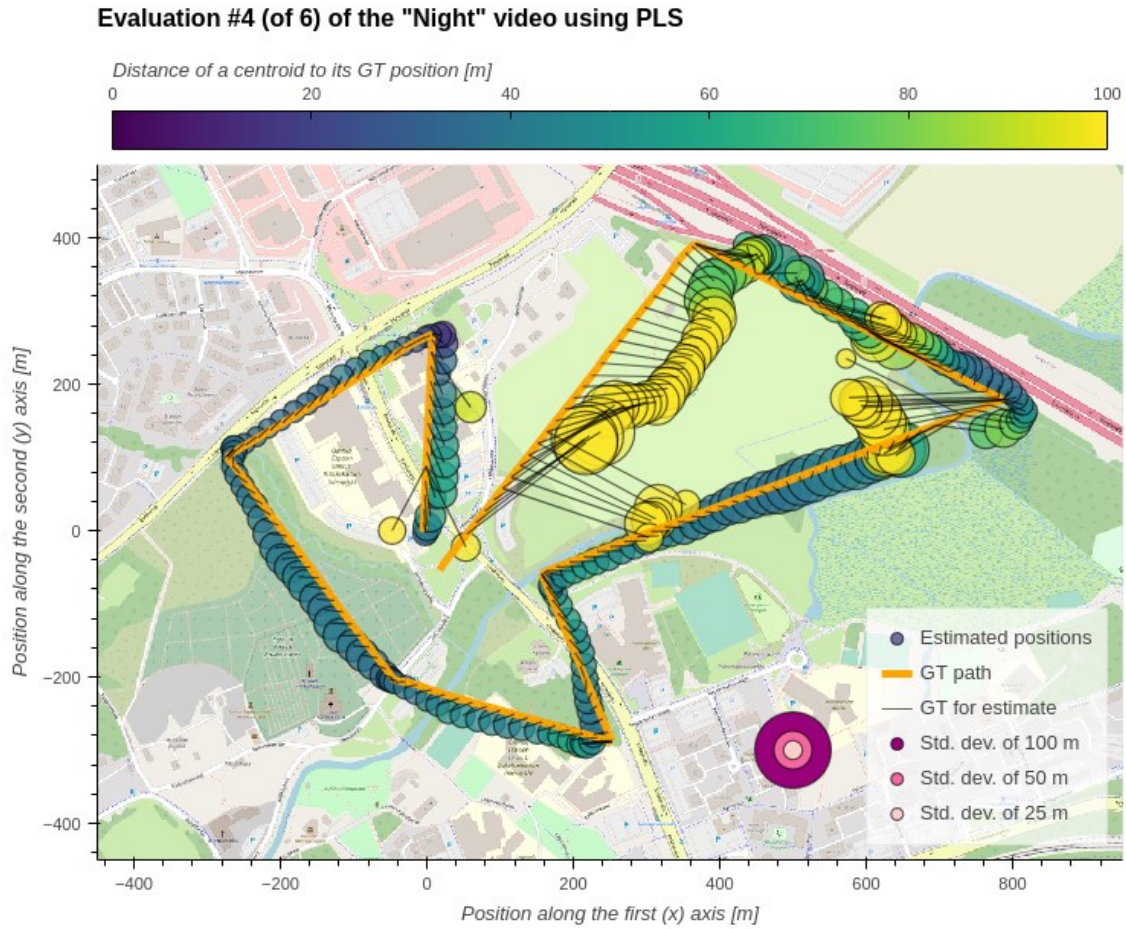

Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

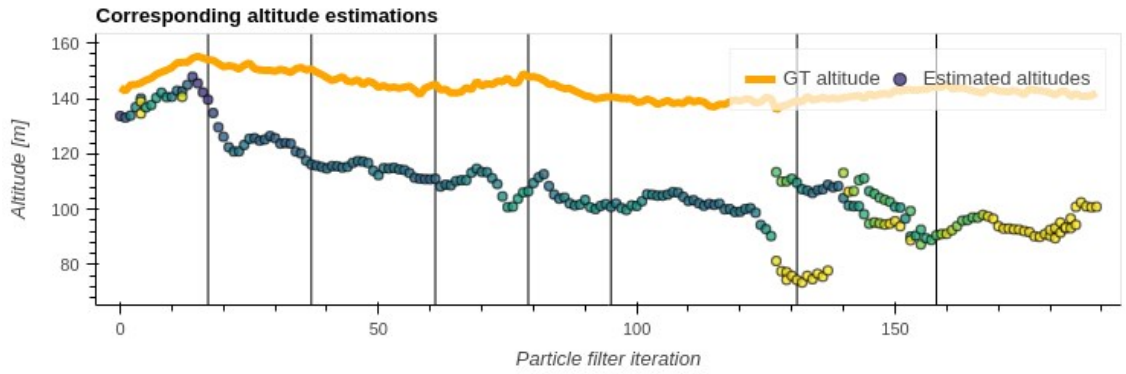

Evaluation #2 (of 6) of the "Night" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

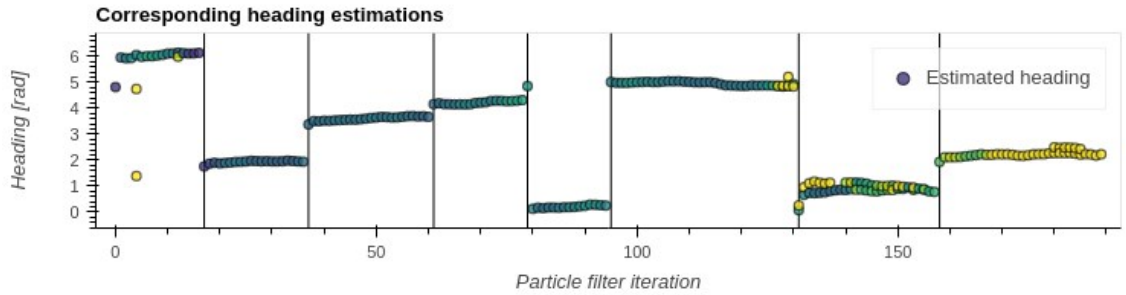

Evaluation #3 (of 6) of the "Night" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m

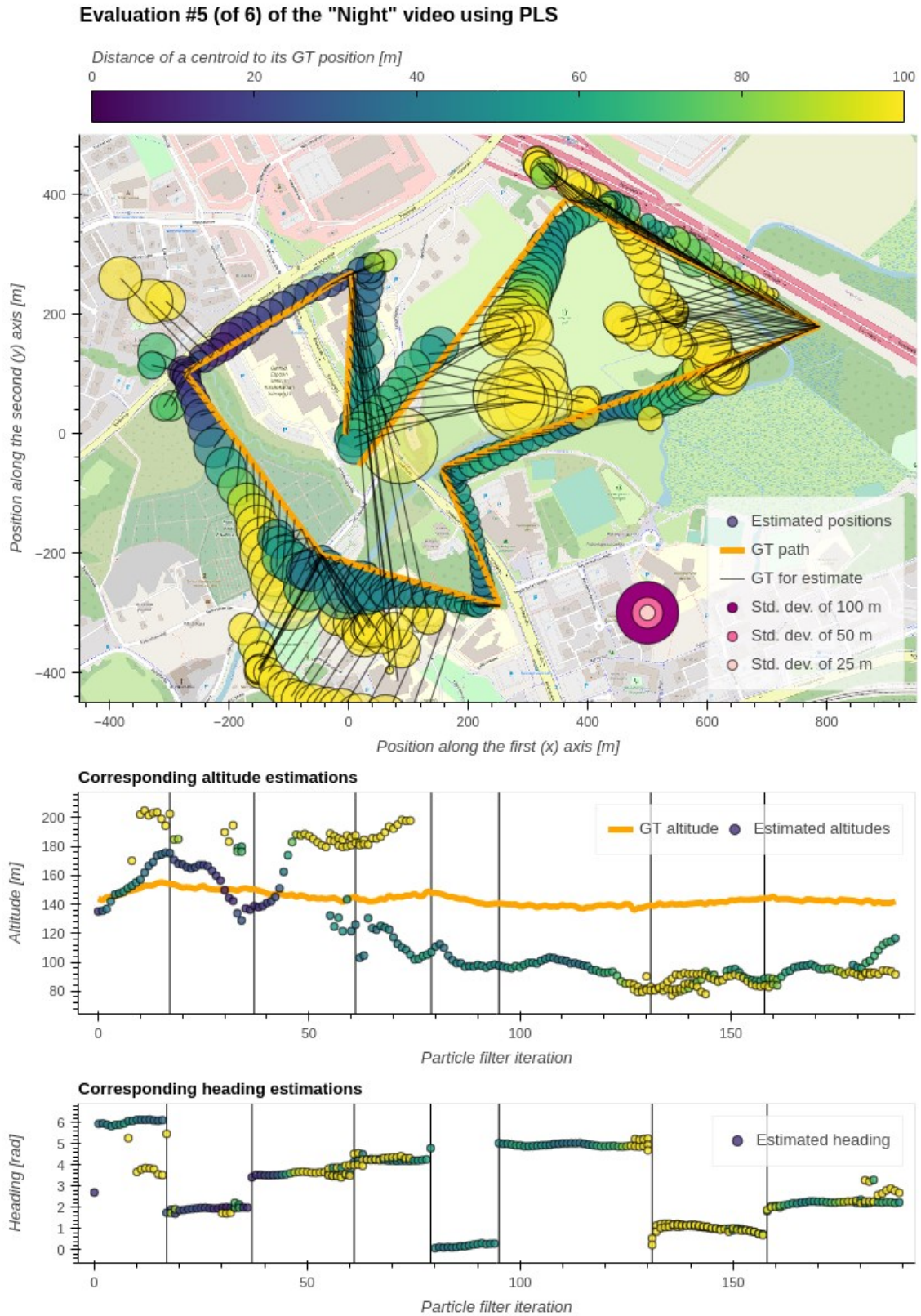

Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes

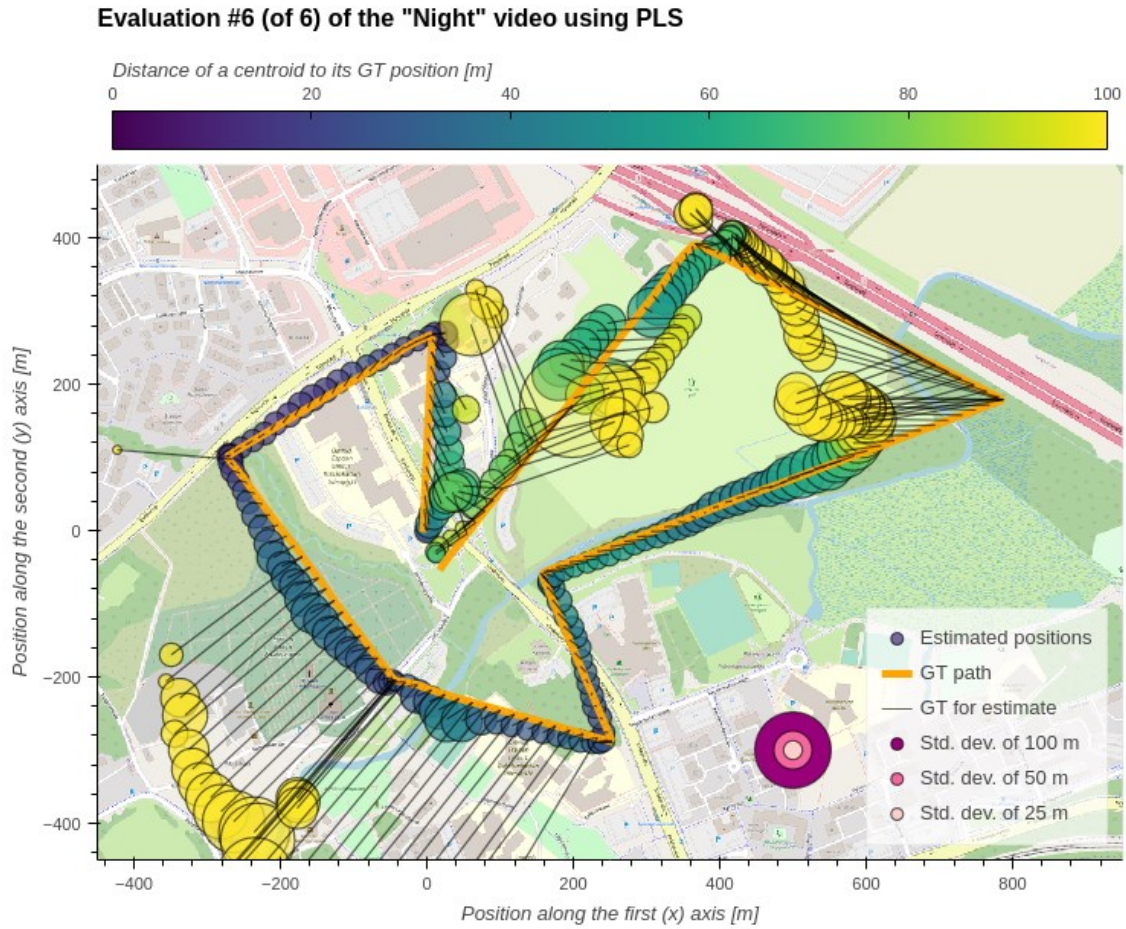

Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

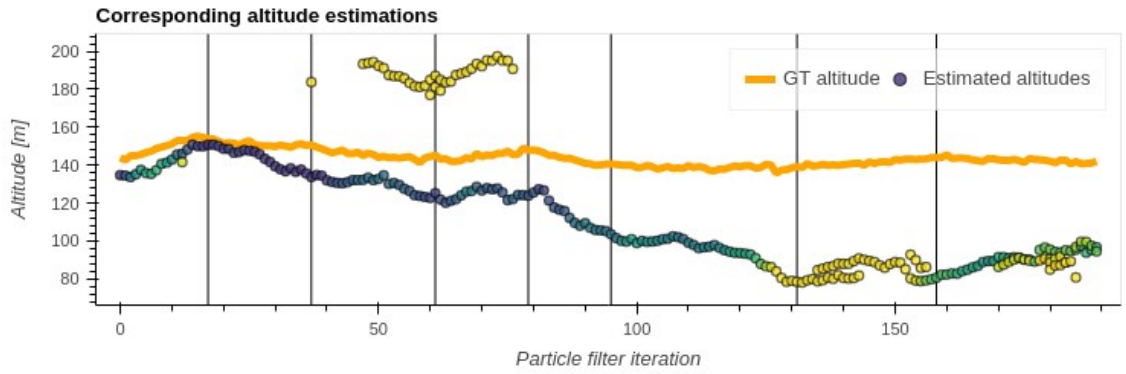

Evaluation #4 (of 6) of the "Night" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m

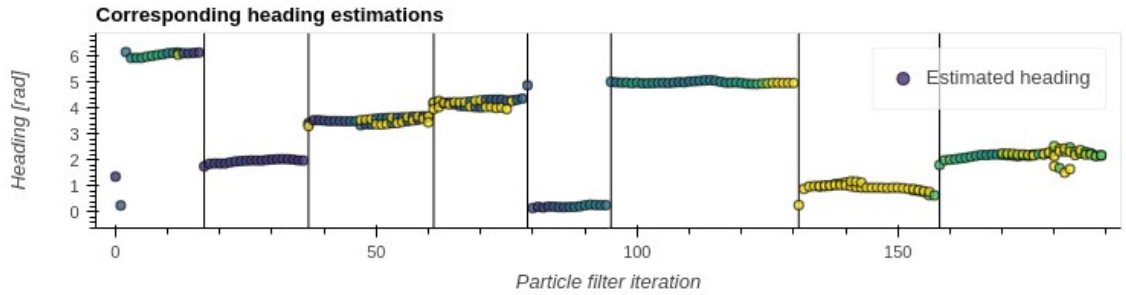

Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes

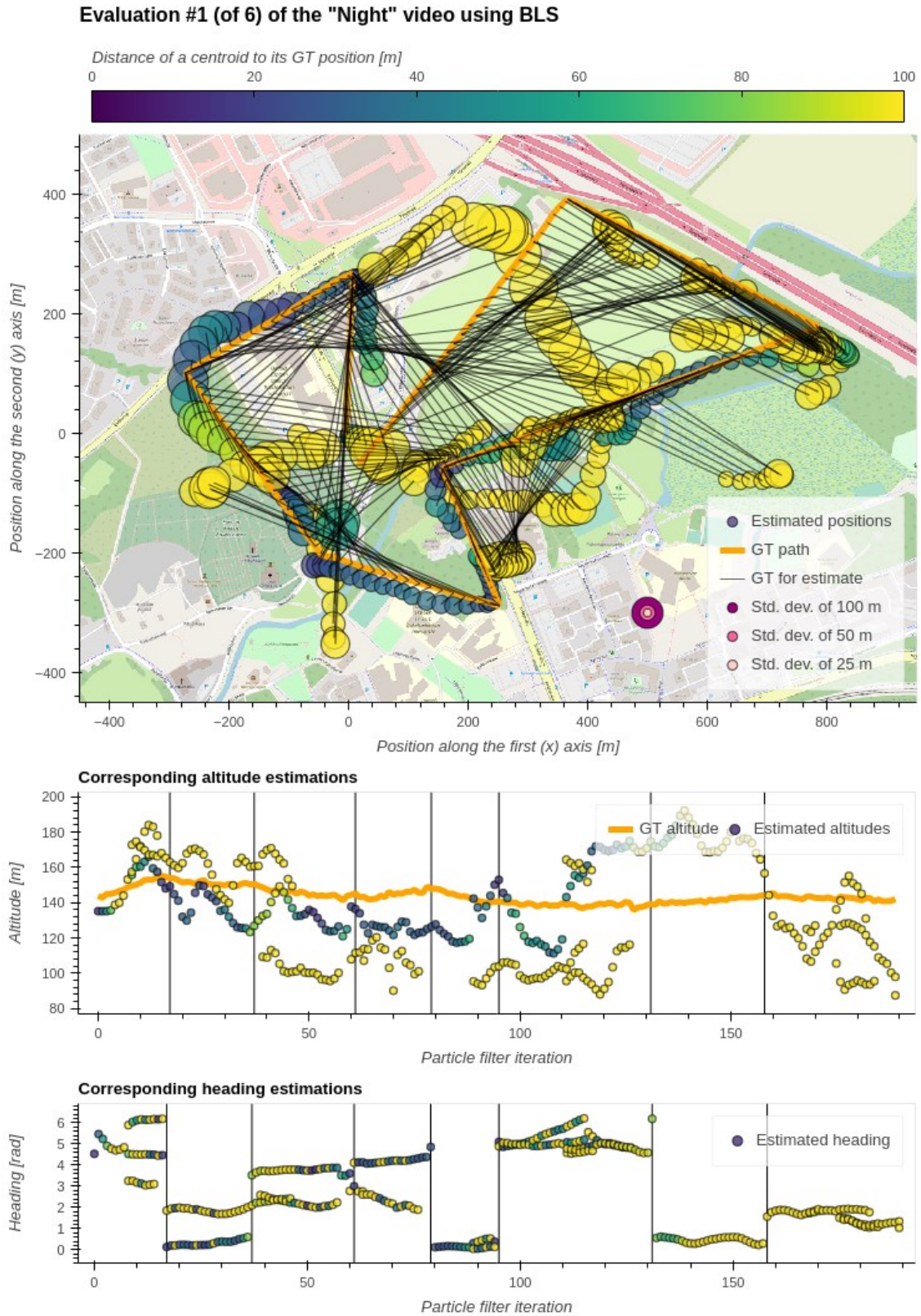

Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

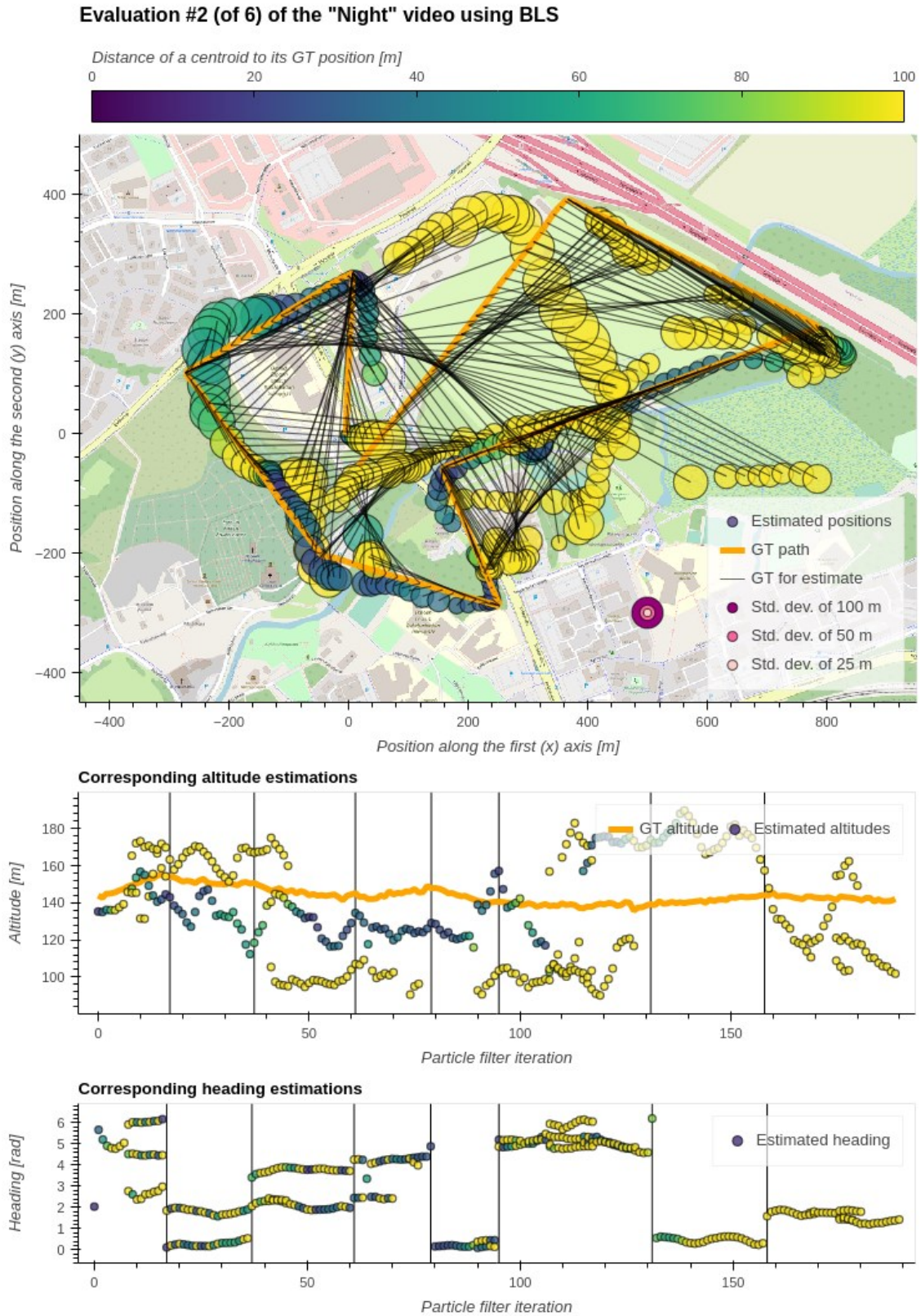

Evaluation #5 (of 6) of the "Night" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
200
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

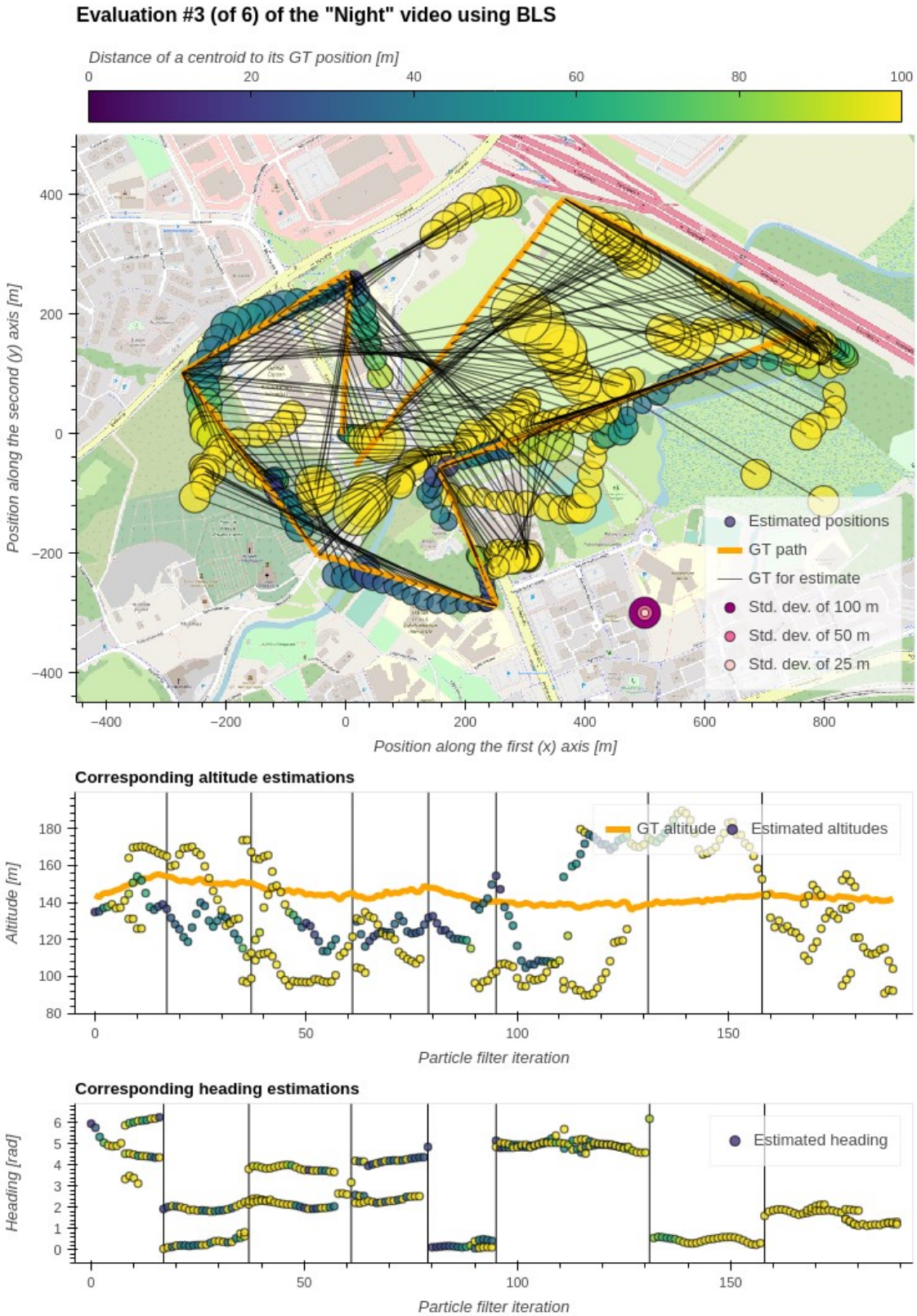

Evaluation #6 (of 6) of the "Night" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m

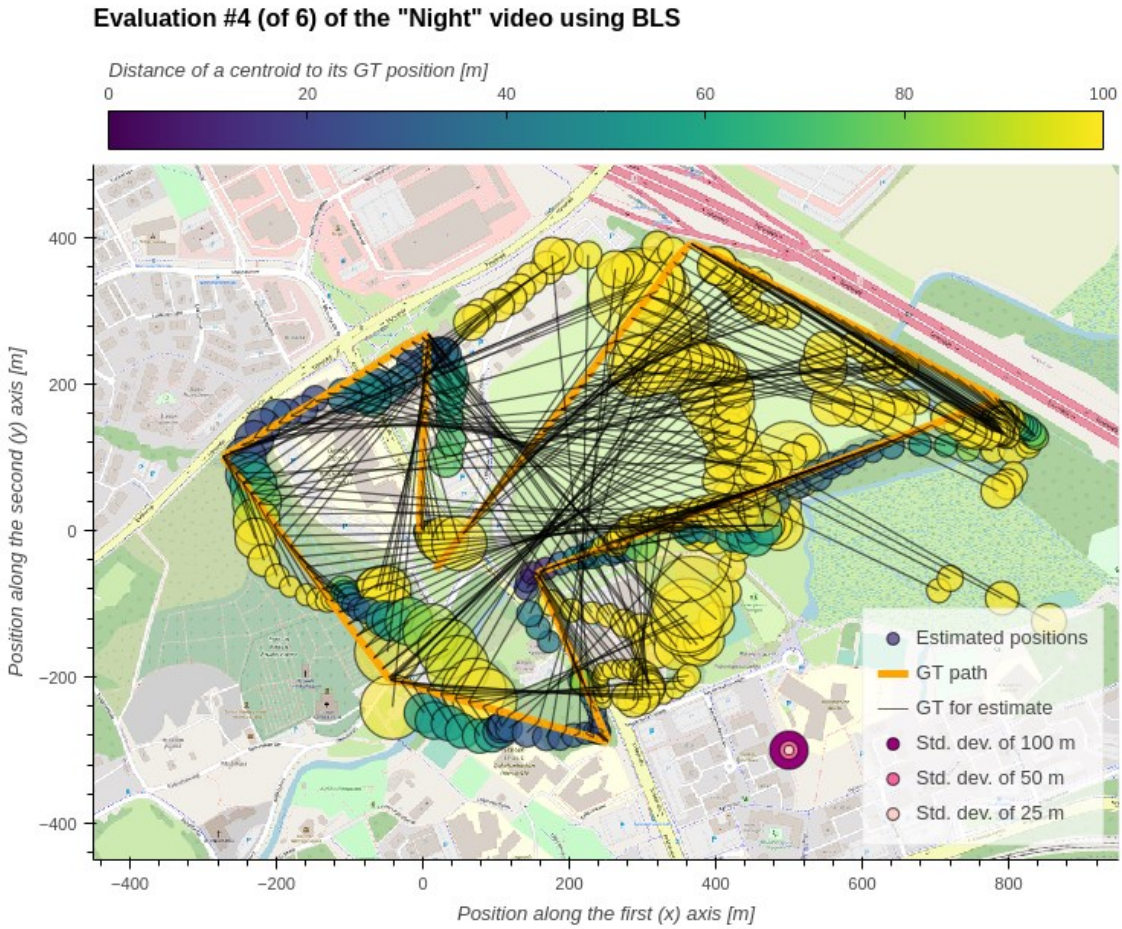

Corresponding altitude estimations
Altitude [m]
200
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes

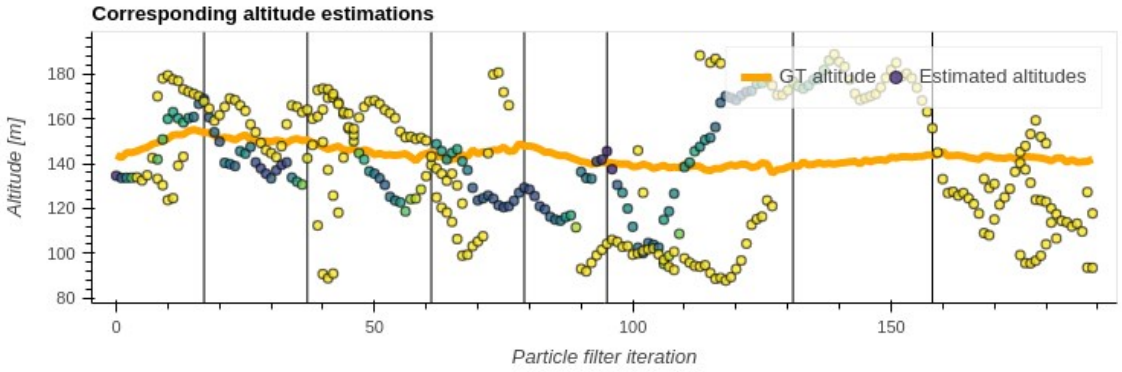

Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

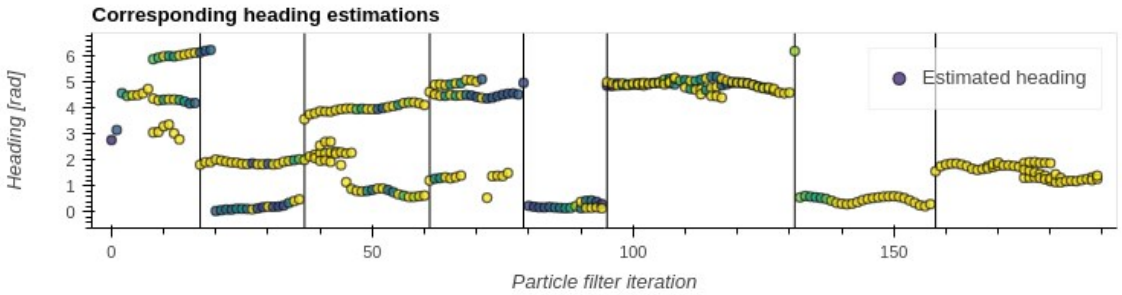

Evaluation #1 (of 6) of the "Night" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

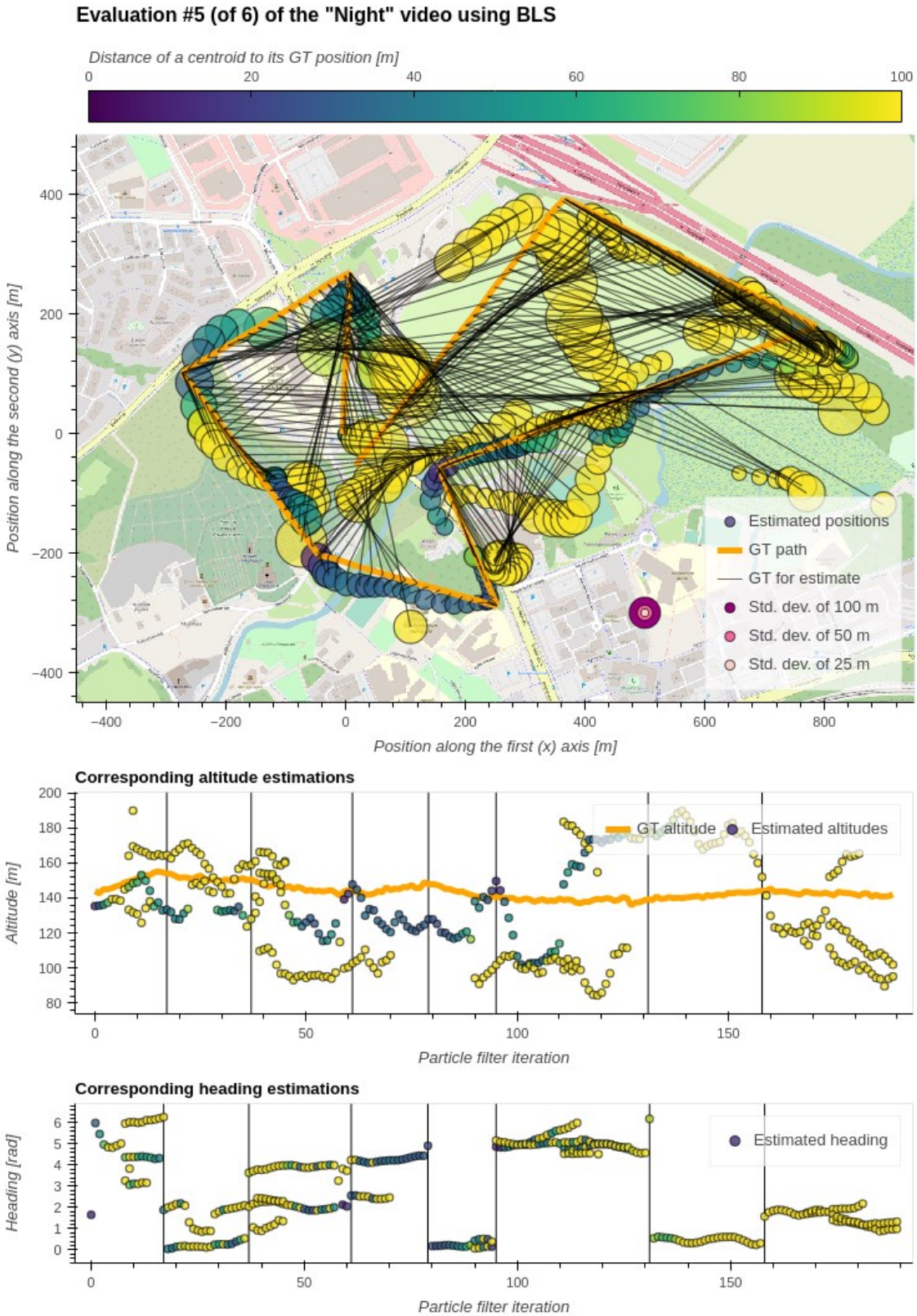

Evaluation #2 (of 6) of the "Night" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

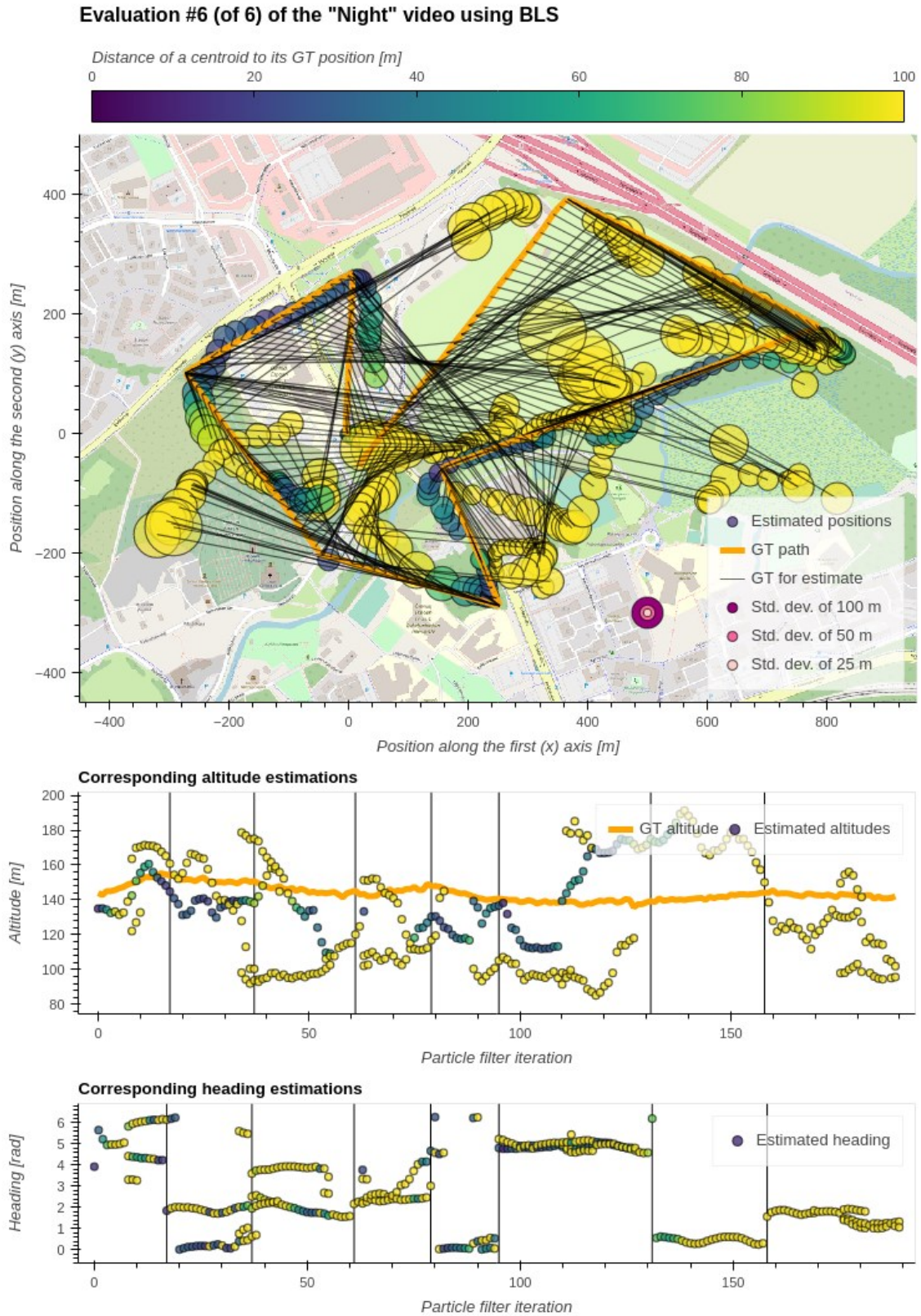

Evaluation #3 (of 6) of the "Night" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

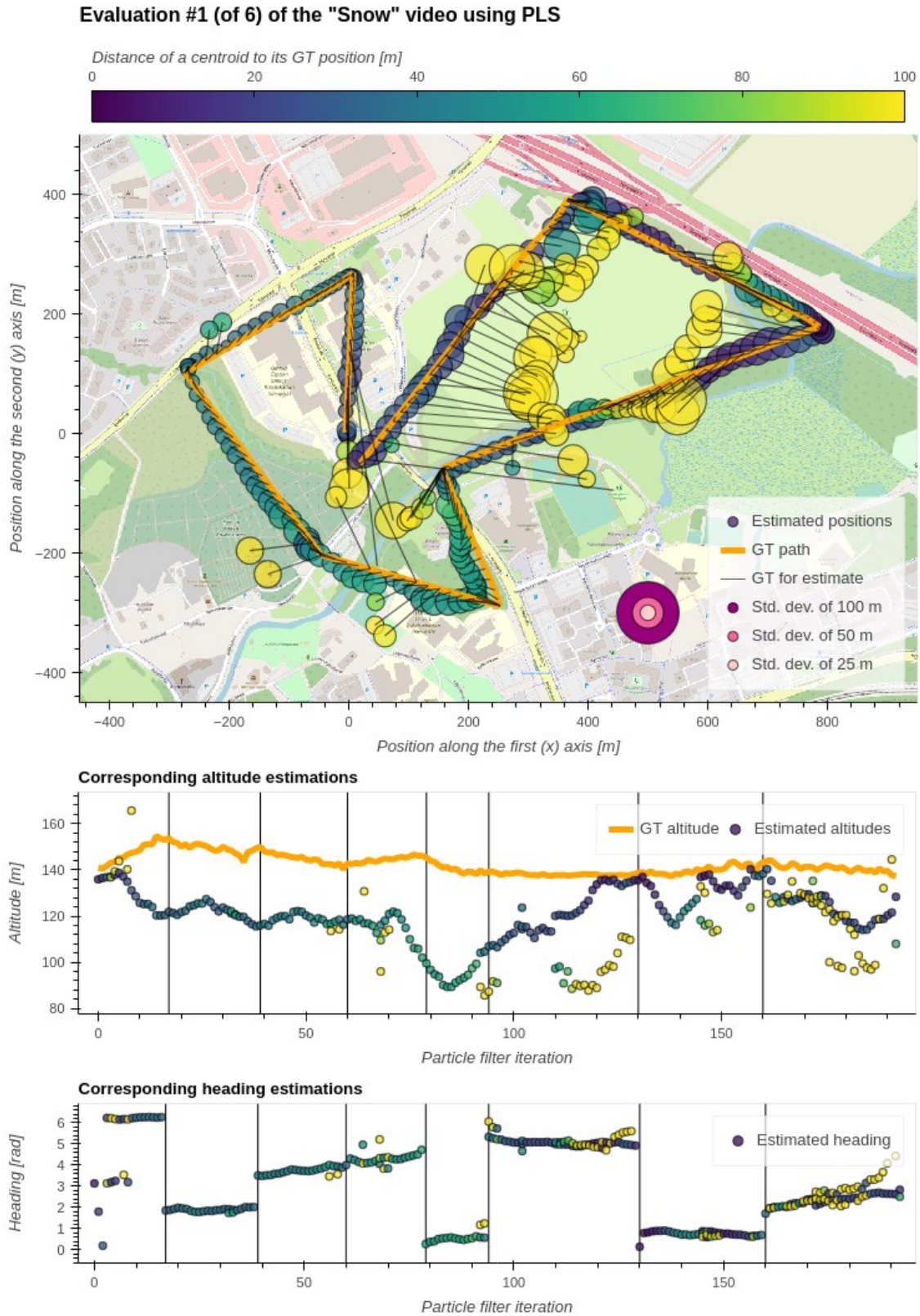

Evaluation #4 (of 6) of the "Night" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m

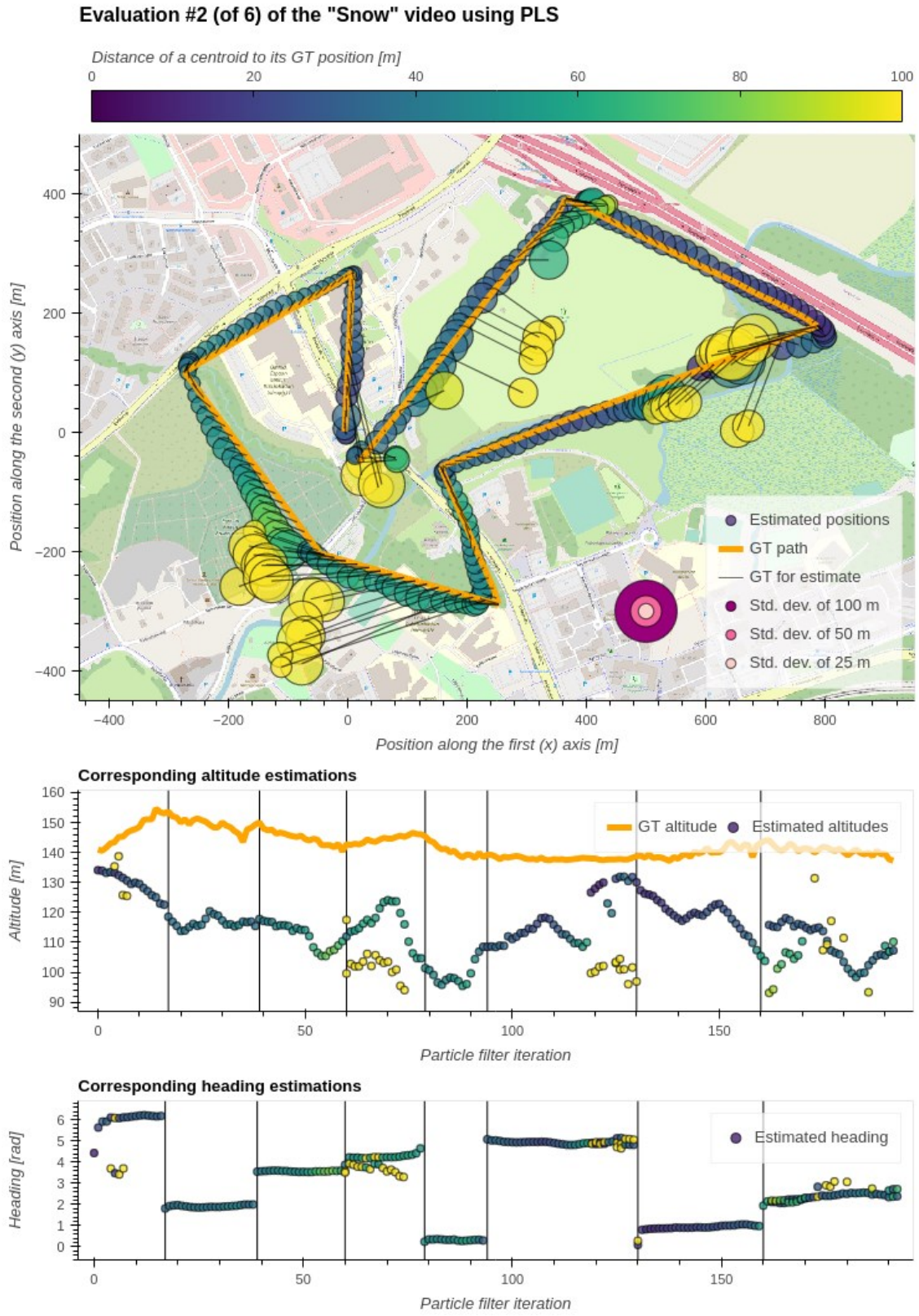

Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes

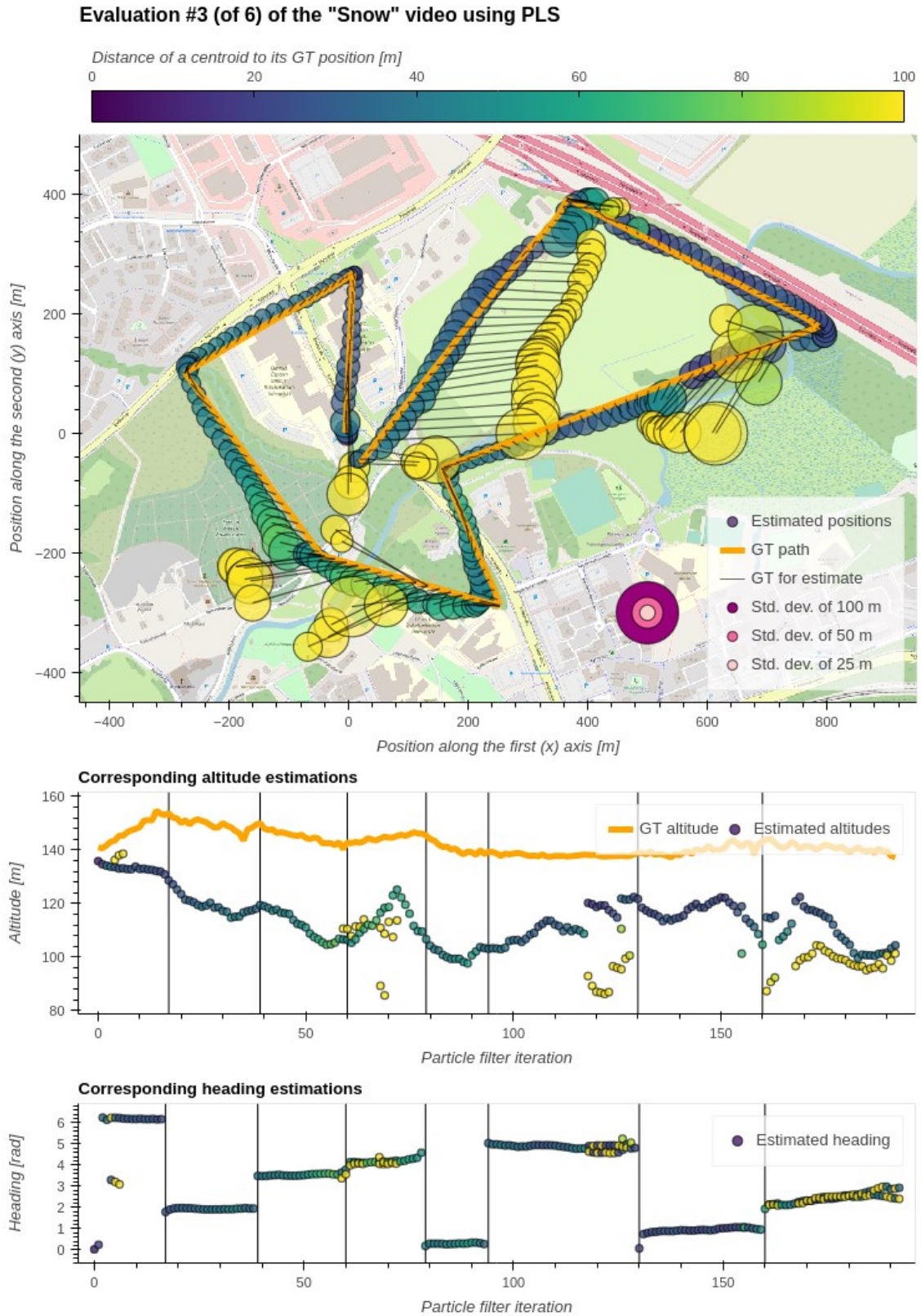

Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

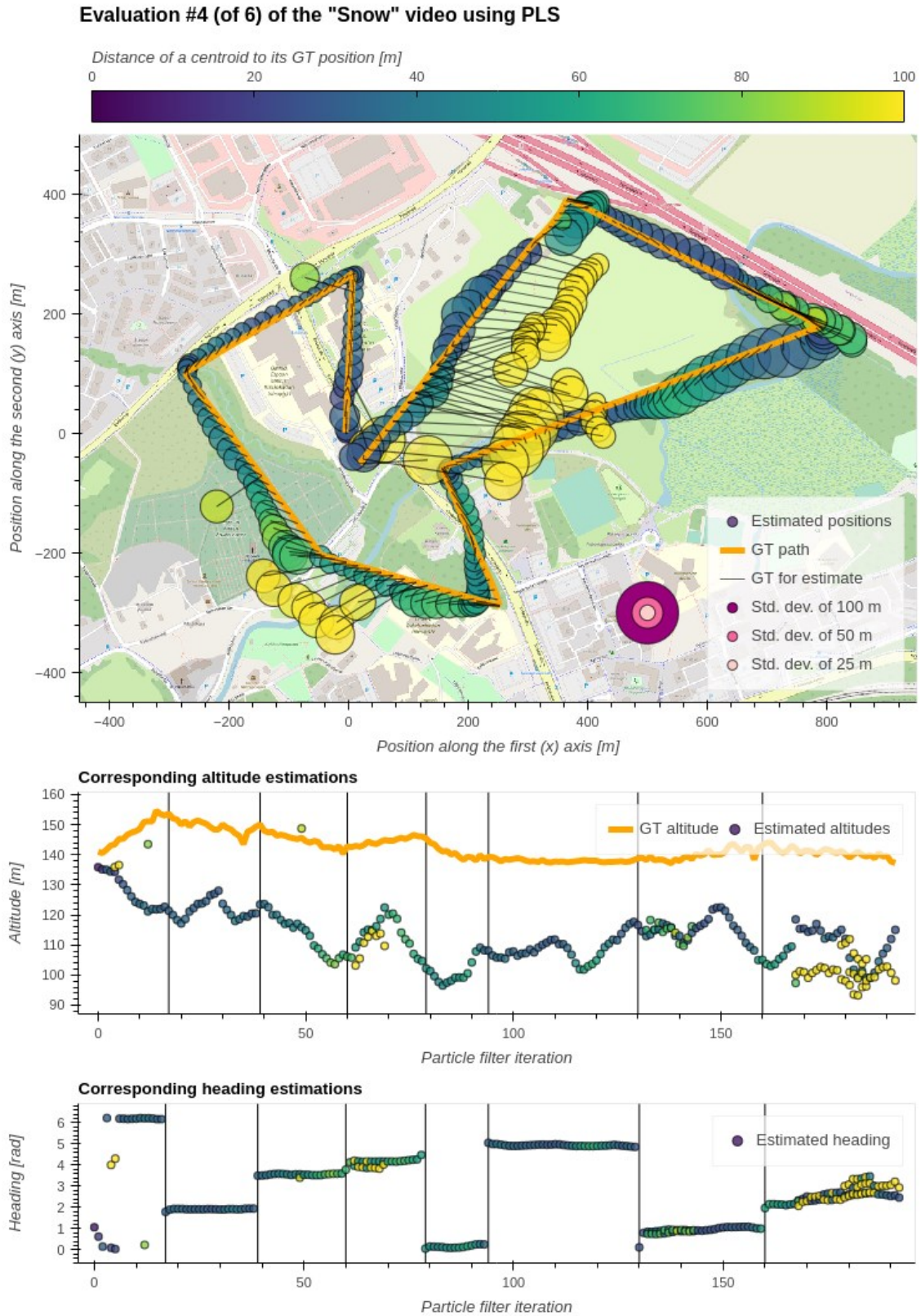

Evaluation #5 (of 6) of the "Night" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

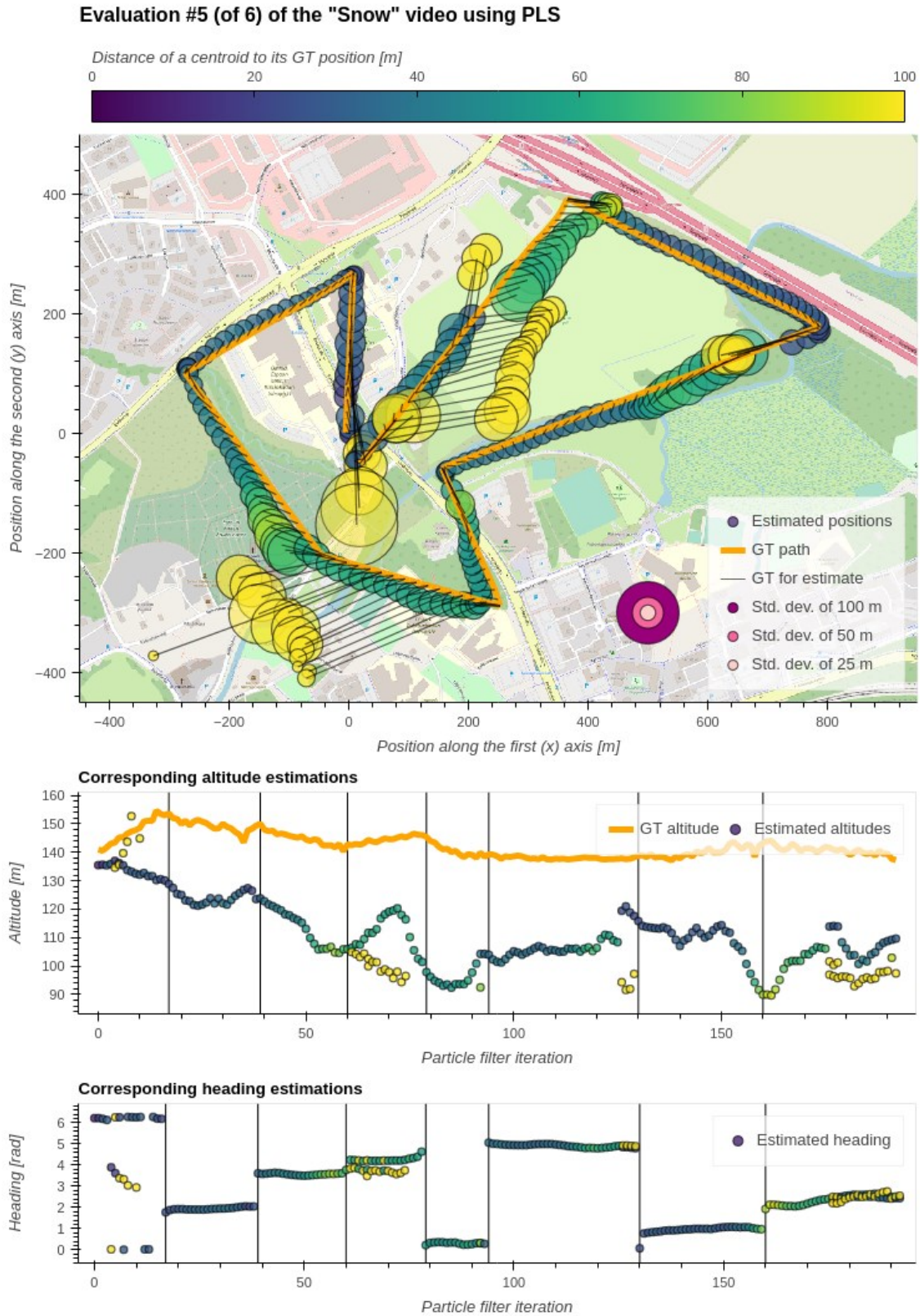

Evaluation #6 (of 6) of the "Night" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration

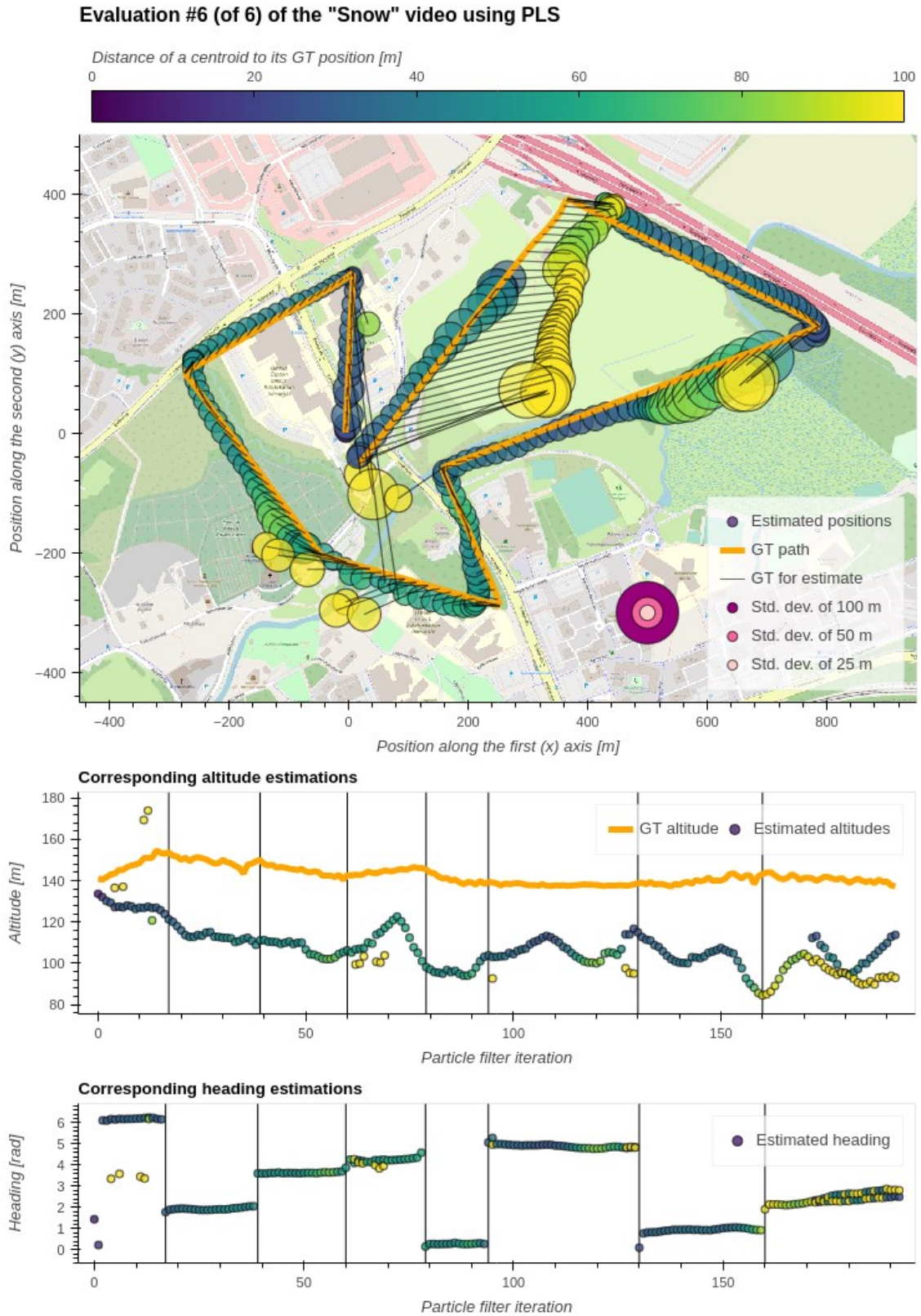

Evaluation #1 (of 6) of the "Snow" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
GT altitude
Estimated altitudes
Altitude [m]
Particle filter iteration
Corresponding heading estimations
Estimated heading
Heading [rad]
Particle filter iteration

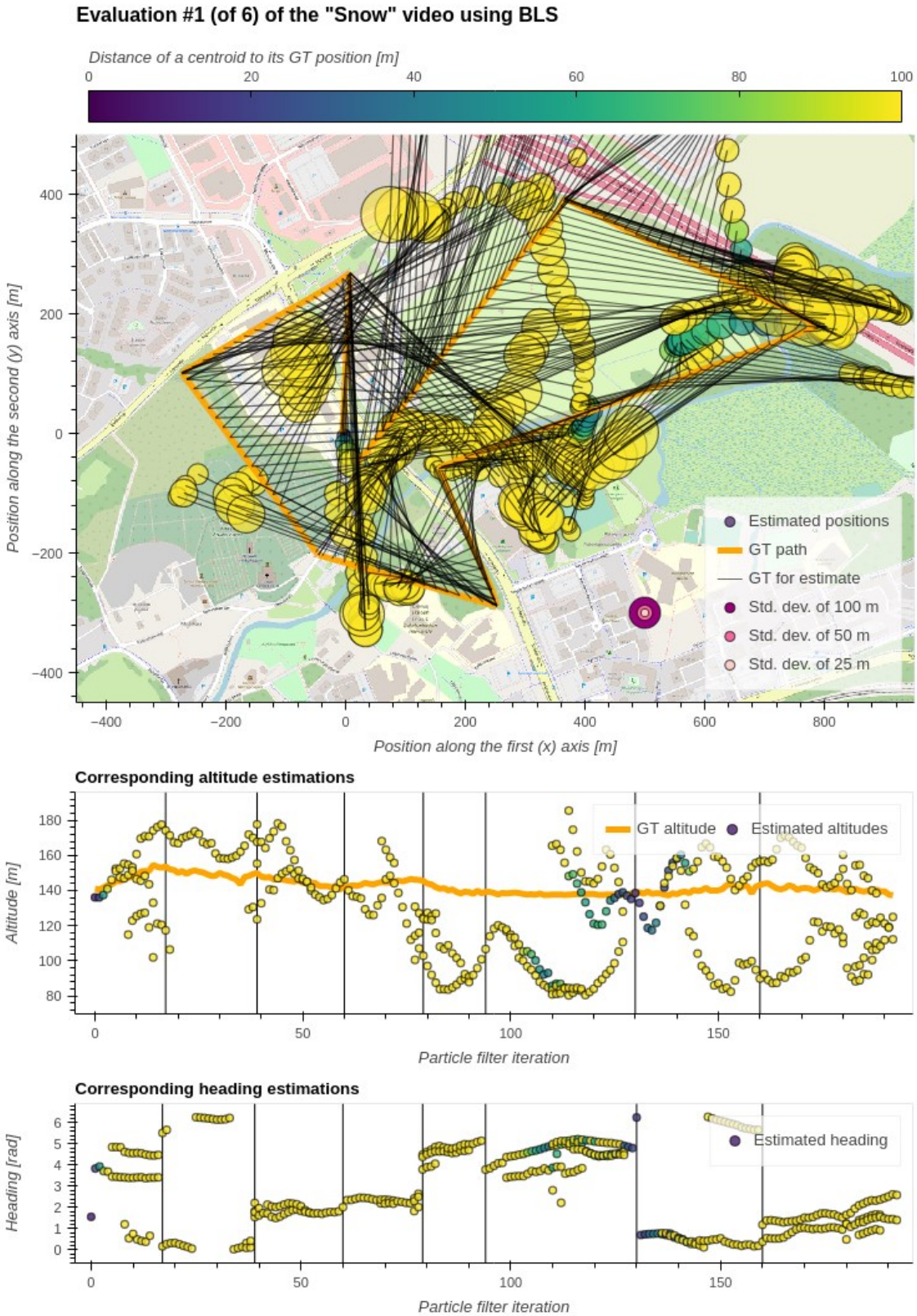

Evaluation #2 (of 6) of the "Snow" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
150
140
130
120
110
100
90
GT altitude
Estimated altitudes
0
50
100
150
Particle filter iteration
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
Estimated heading
0
50
100
150
Particle filter iteration

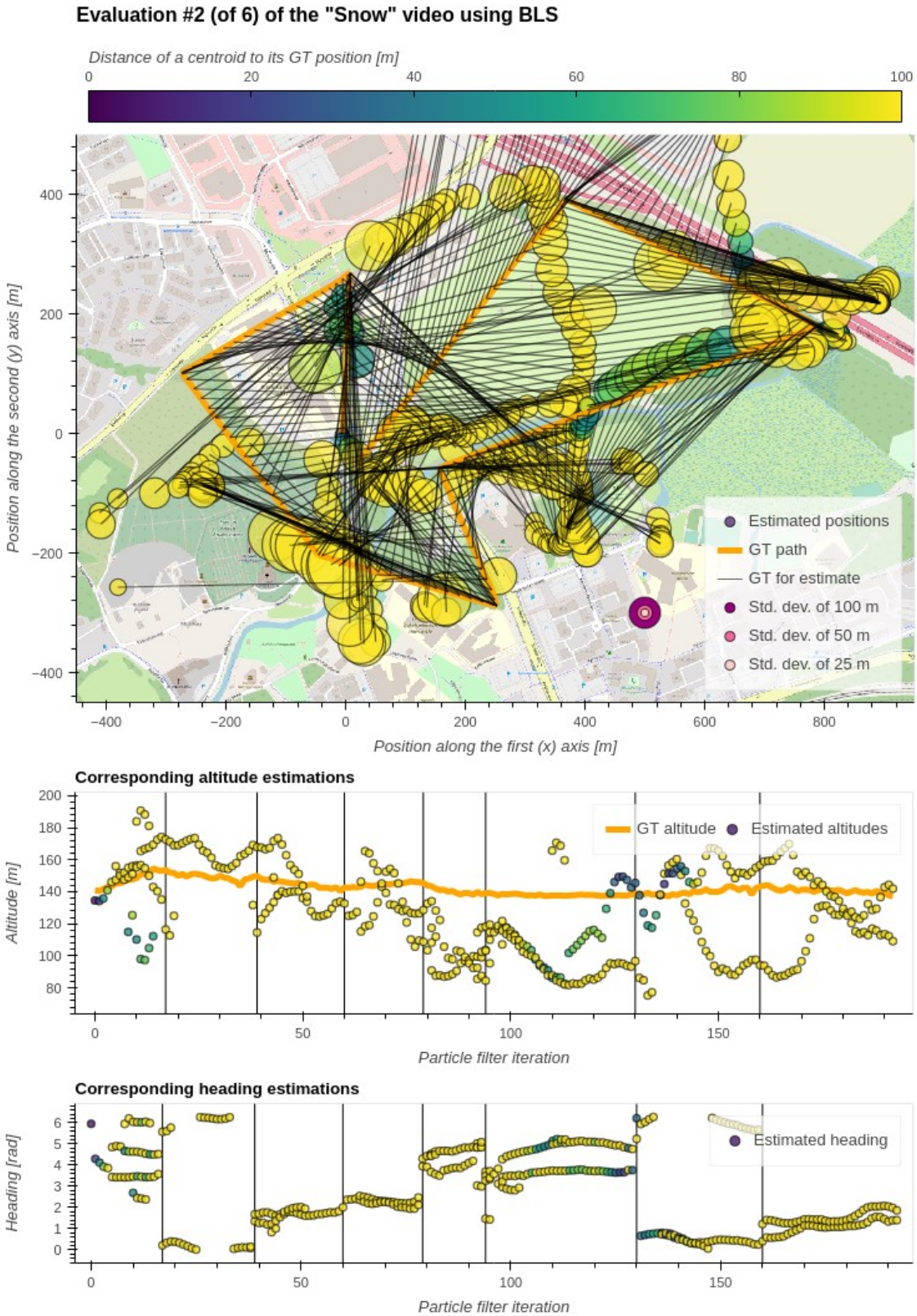

Evaluation #3 (of 6) of the "Snow" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

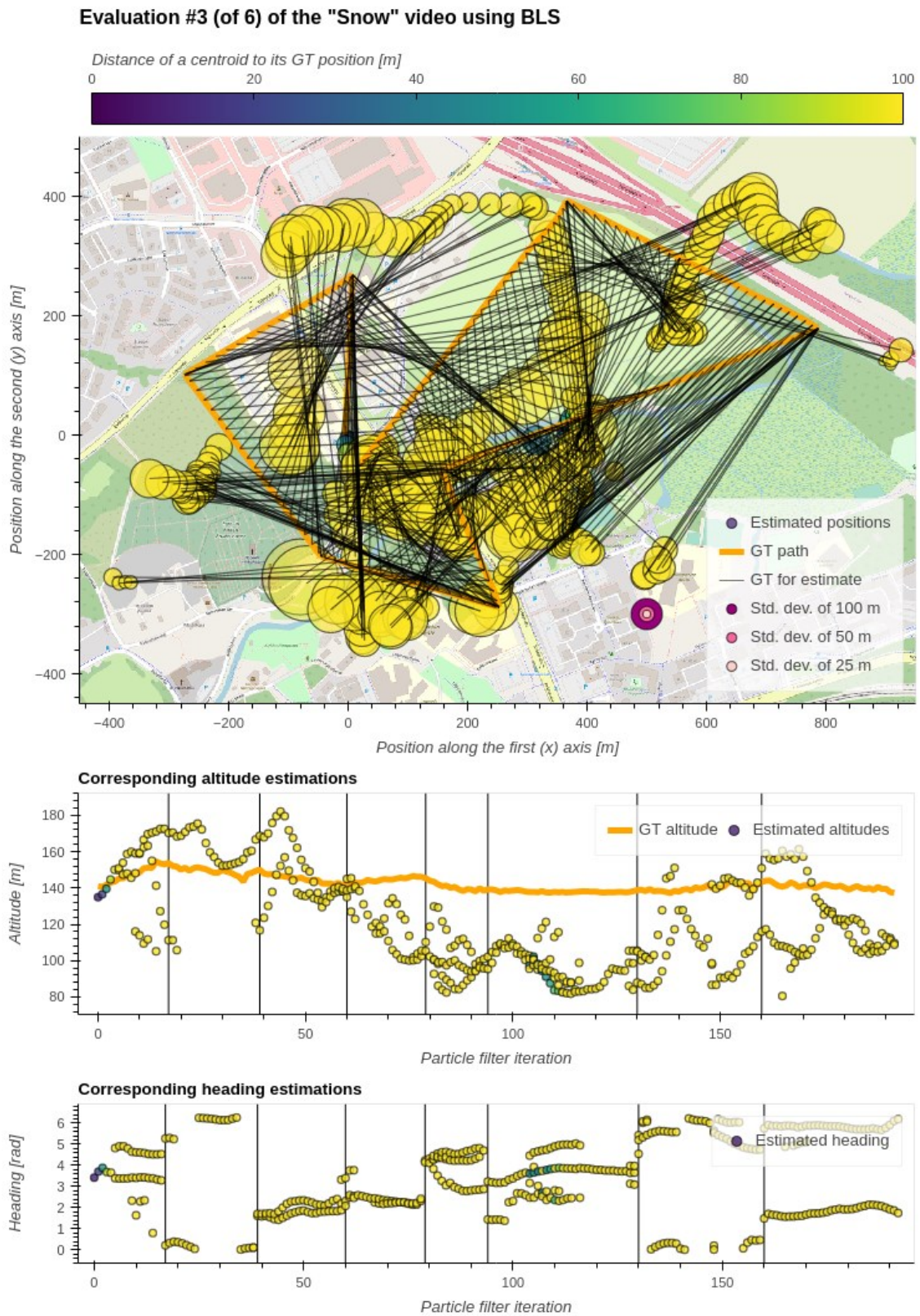

Evaluation #4 (of 6) of the "Snow" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
GT altitude
Estimated altitudes
Altitude [m]
Particle filter iteration
Corresponding heading estimations
Estimated heading
Heading [rad]
Particle filter iteration

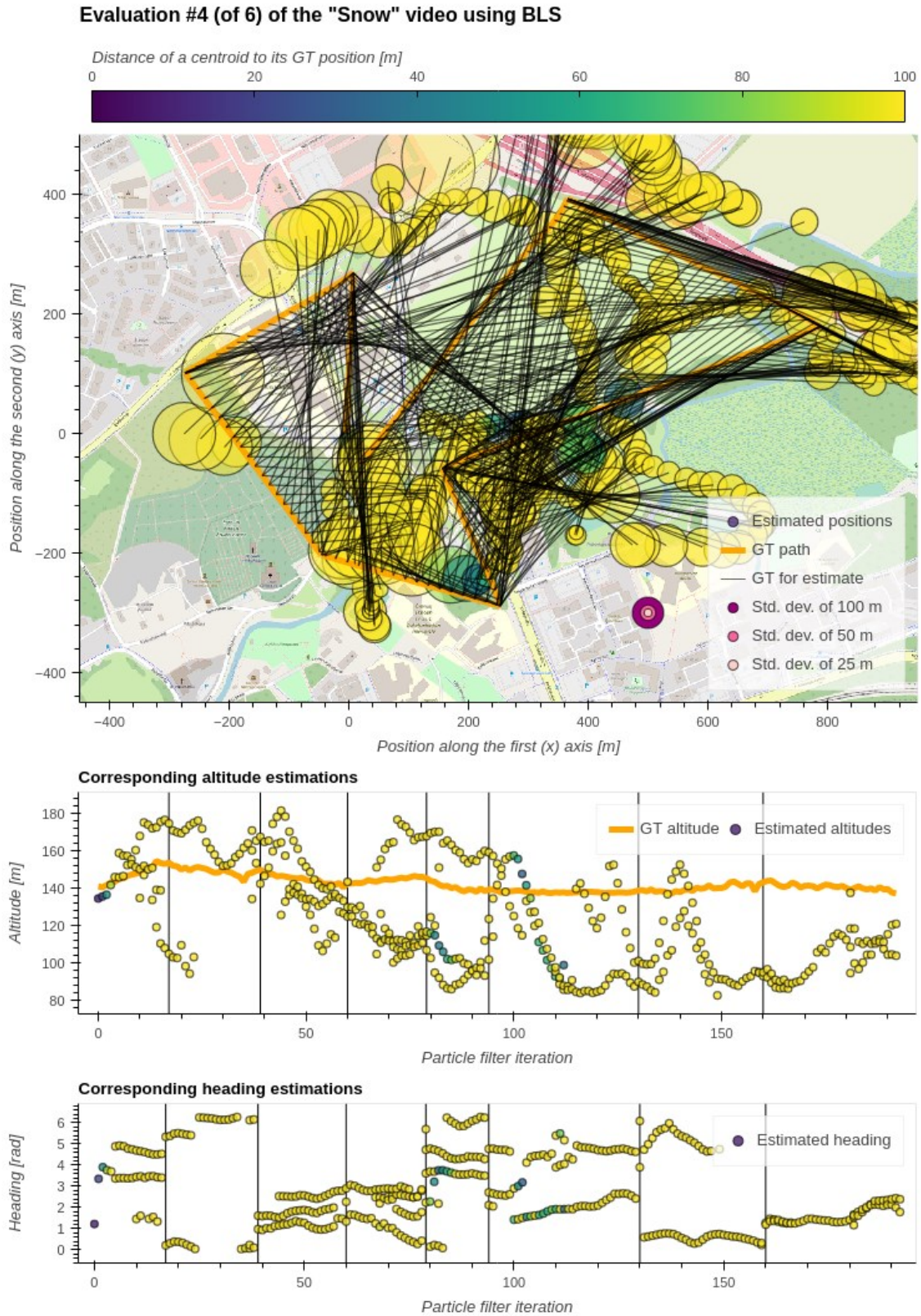

Evaluation #5 (of 6) of the "Snow" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
160
150
140
130
120
110
100
90
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

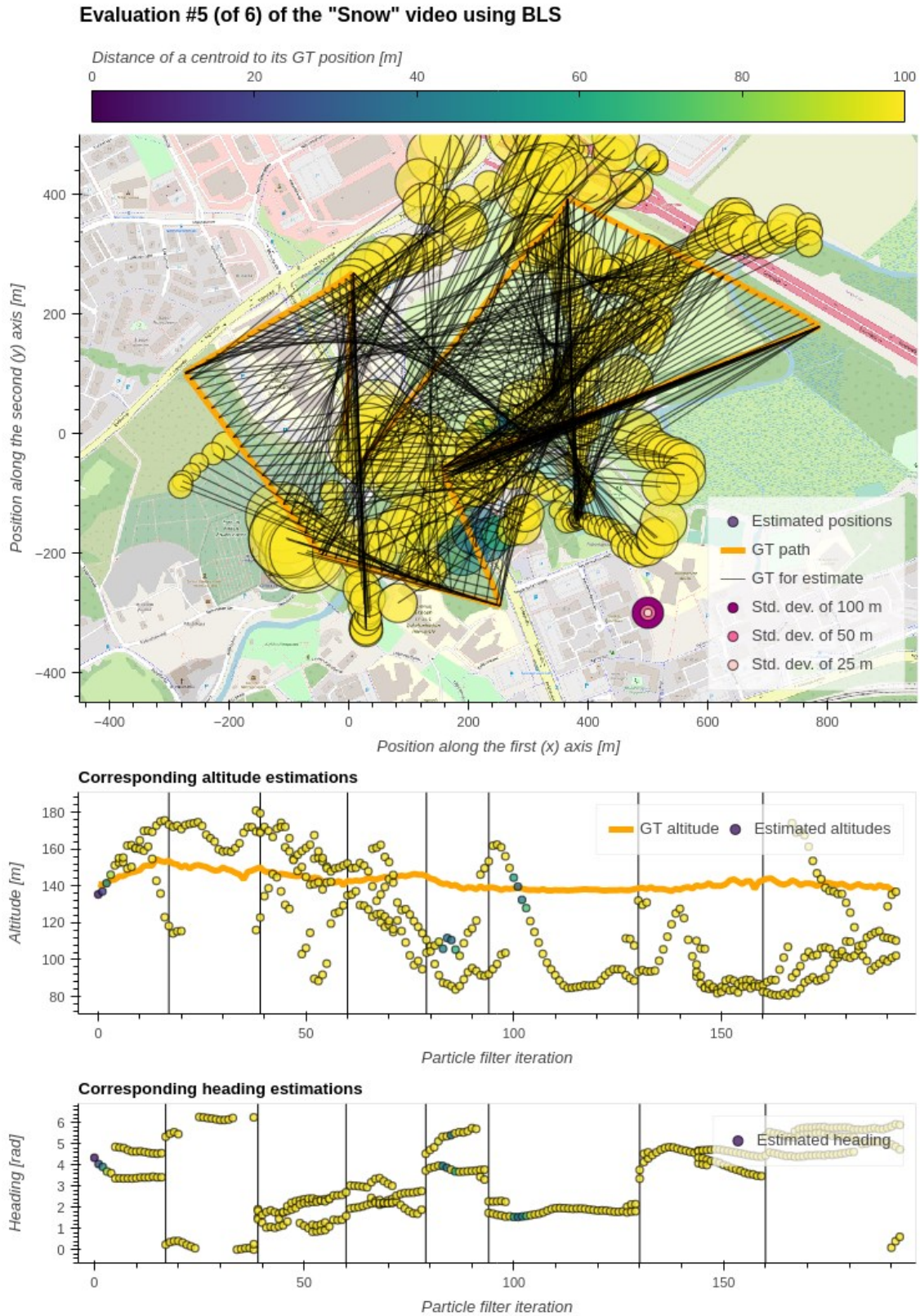

Evaluation #6 (of 6) of the "Snow" video using PLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading

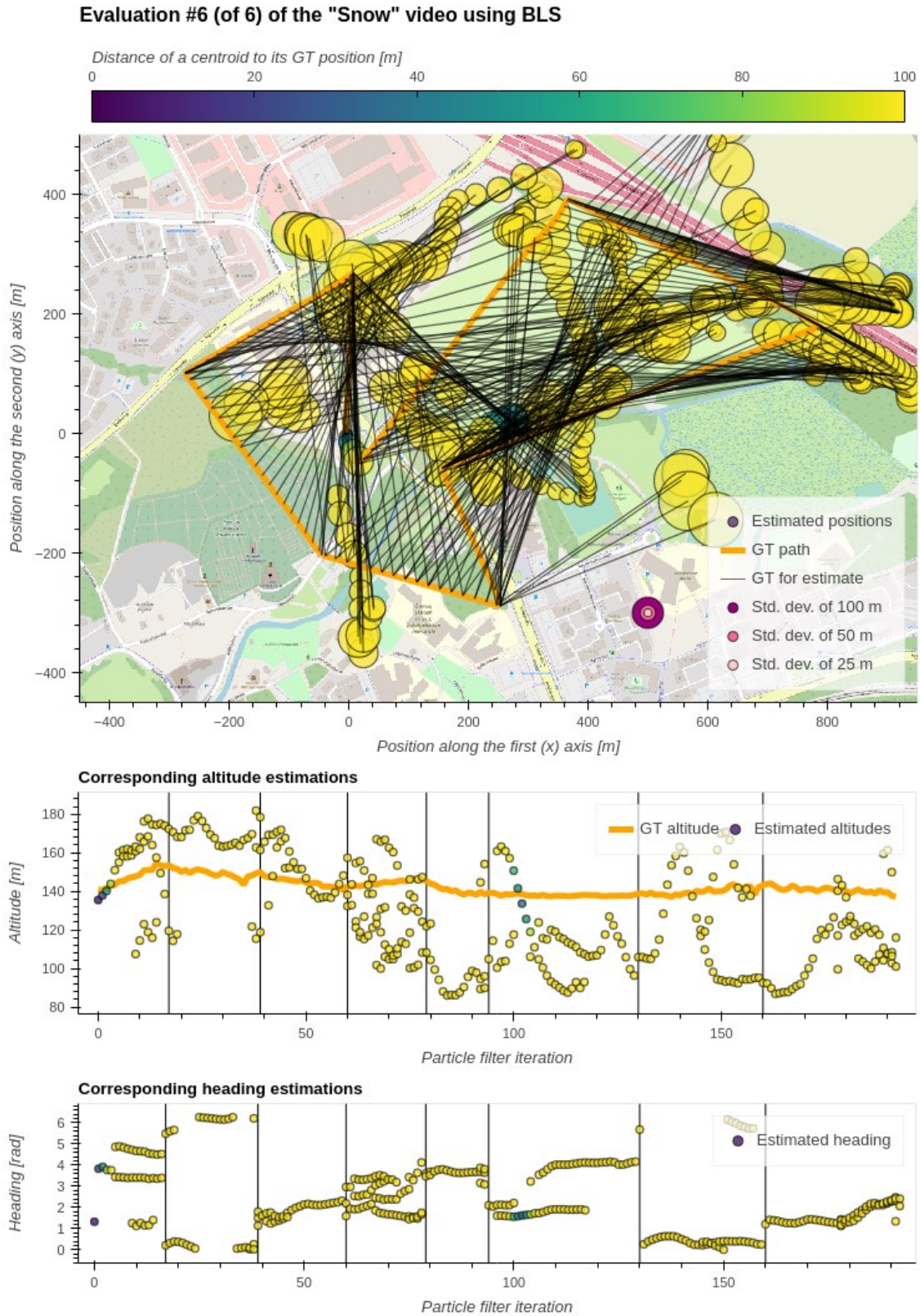

Evaluation #1 (of 6) of the "Snow" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration


Evaluation #2 (of 6) of the "Snow" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
200
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading


Evaluation #3 (of 6) of the "Snow" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration


Evaluation #4 (of 6) of the "Snow" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading


Evaluation #5 (of 6) of the "Snow" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
400
200
0
−200
−400
−400
−200
0
200
400
600
800
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
180
160
140
120
100
80
0
50
100
150
Particle filter iteration
GT altitude
Estimated altitudes
Corresponding heading estimations
Heading [rad]
6
5
4
3
2
1
0
0
50
100
150
Particle filter iteration
Estimated heading


Evaluation #6 (of 6) of the "Snow" video using BLS
Distance of a centroid to its GT position [m]
0
20
40
60
80
100
Position along the second (y) axis [m]
Position along the first (x) axis [m]
Estimated positions
GT path
GT for estimate
Std. dev. of 100 m
Std. dev. of 50 m
Std. dev. of 25 m
Corresponding altitude estimations
Altitude [m]
GT altitude
Estimated altitudes
Particle filter iteration
Corresponding heading estimations
Heading [rad]
Estimated heading
Particle filter iteration