A hardware-accelerated particle filter for the geolocation of demersal fishes

Chang Liu[1], Geoffrey W. Cowles[1], Douglas R. Zemeckis[2], Gavin Fay[1], Arnault Le Bris[3], Steven X. Cadrin[1]

[1]Department of Fisheries Oceanography, School for Marine Science and Technology, University of Massachusetts Dartmouth. 836 S Rodney French Blvd, New Bedford, MA 02744, USA

[2]Department of Agriculture and Natural Resources, Rutgers, The State University of New Jersey. 1623 Whitesville Road, Toms River, NJ 08755, USA

[3]Centre for Fisheries Ecosystems Research, Fisheries and Marine Institute of Memorial University of Newfoundland, St. John's, Canada

Corresponding author: Chang Liu, cliu3@umassd.edu

Keywords: geolocation, demersal fish, fish migration, particle filter, archival tagging, data storage tag, graphics processing unit

Funding: Funding for the research conducted as part of this manuscript was provided by NOAA Saltonstall-Kennedy Grant award NA15NMF4270267. Cod tagging research in the Spring Cod Conservation Zone was conducted in collaboration with the Massachusetts Division of Marine Fisheries and supported by the United States Fish and Wildlife Service through the Sportfish Restoration Act and the Massachusetts Marine Fisheries Institute.

Abstract

Geolocation is increasingly employed to reconstruct the movements of demersal fishes using data retrieved from electronic archival tags. However, geolocation methods commonly suffer from limitations such as low horizontal resolution of locations, flawed land boundary treatment, and extensive computation time. We addressed these issues using a state-space approach based on the particle filter (PF), and developed a geolocation package with graphics processing unit (GPU) acceleration. Our method focused on application to demersal fish and utilizes comparison of the tag-recorded depth and temperature to the same variables from an unstructured grid regional oceanographic model. A rigorous boundary treatment scheme was implemented to handle regions with complex coastline geometry. Validation exercises using stationary mooring tags and double-electronic-tagged (archival and acoustic tags) Atlantic cod in the Gulf of Maine resulted in <10 km median errors of the estimated tracks. Sensitivity analyses suggest that using 200,000 particles was adequate to stabilize the location track estimation. Acceleration of the particle filter using GPUs resulted in faster processing than the single threaded CPU (central processing unit) implementation, enabling rapid geolocations using consumer grade computer hardware. The geolocation output of each tagged fish includes the most probable track and the associated spatial probability distribution. The resulting PF geolocation package enables high resolution and accelerated geolocation analyses to be performed on affordable consumer-grade computer hardware, resolving the time intensiveness problem of the PF that may have prevented its adoptions in marine animal geolocation. Expanded application of geolocation will yield more reliable migration information to support management. Geolocation results from archival tagging will contribute to our understanding of the spatial ecology of marine species.

1 Introduction

Electronic tagging has offered improved fishery-independent insights into behavior and population structure of marine species (Galuardi and Lam, 2014; Hussey et al., 2015). Two commonly employed variants of electronic archival tags are data storage tags (DSTs) and pop-up satellite archival tags (PSATs). These are relatively compact devices that can be attached to a fish and are capable of recording key environmental data such as pressure (i.e., depth), light level, and temperature at precise time intervals, typically seconds to minutes. These data may be used to estimate locations and possible migration paths of the tagged individual through geolocation. The majority of geolocation methods for tracking individual aquatic animals use GPS and light level (Galuardi and Lam, 2014). However, due to attenuation in the water column, these signals are not suitable for geolocation of demersal species that reside at depth on or near the bottom of the water column. For demersal fish, geolocation using tag-recorded depth and temperature data is a more appropriate approach and has been incorporated into several methods, including Metcalfe and Arnold (1997); Hunter et al. (2003); Andersen et al. (2007); Righton and Mills (2008); Pedersen et al. (2008). Many of these prior approaches are based on state-space models that account for uncertainties related to the observations and the estimated quantities (Pedersen et al., 2008; Thygesen et al., 2009; Patterson et al., 2008; Jonsen et al., 2013).

The particle filter (PF), also known as sequential importance resampling or sequential Monte Carlo, is a statistical method that is commonly applied to tracking applications in fields such as robotics and image processing (Gustafsson et al., 2002). The PF has also been employed for fish geolocation using archival tagging data (Nielsen, 2004; Royer et al., 2005; Andersen et al., 2007; Brickman and Thorsteinsson, 2008; Coleman, 2015), where the possible geographic location of the fish is modeled by an ensemble of samples, or particles, filtered by the likelihood distributions in an iterative manner. An approach that has been more widely applied to the archival tagging geolocation problem is the hidden Markov model (HMM). HMMs typically require a known, finite number of states, thus the HMM-based geolocation methods operate on a horizontal regular rectangular grid. HMM-based geolocation software packages have been developed and made available by several research groups (e.g., Pedersen et al. 2008, 2011a; Liu et al. 2017; Braun et al. 2018). In comparing these two methods, the PF has two key advantages over the HMM-based methods for state-space modeling in the context of the geolocation problem. The first is that the PF is better suited for filtering both nonlinear and non-Gaussian probability density distributions for the horizontal locations. This is particularly advantageous for handling simulations when the fish is in coastal waters near land (Andersen et al., 2007) where Gaussian distributions are not suitable. In the PF, confinement to the domain can be implemented in a straightforward and robust manner. The second advantage of the PF over HMM-based geolocation is that the PF assumes a continuous state space for particle locations, i.e., modeled particle locations are not constrained to a finite set of discrete grid points of an underlying horizontal grid. This avoids the need for any interpolation or discretization of the 2-D spatial distributions onto fixed grids as required by the HMM approach, which may lead to information loss and render geolocation results dependent on the horizontal resolution.

Previous studies identified that a major drawback of the PF is that it can be computationally intensive due to the large number of particles needed for a given simulation (Pedersen et al., 2008; Thygesen et al., 2009; Woillez et al., 2016). This is likely the reason why the PF has been infrequently employed in geolocation studies despite the clear benefits of the approach. Fortunately, the nature of the PF algorithm enables the employment of modern computer hardware acceleration approaches to significantly reduce the computation time. The parallelization of the PF algorithm using multiple CPU cores or graphics processing units (GPU) to reduce runtime has been studied in the context of other applications (Hendeby et al., 2010; Goodrum et al., 2011). A GPU is a computer hardware device that was traditionally used to create images to be rendered on a display. Over the last two decades, software tools and algorithms have been developed to enable GPUs to be used to accelerate general purpose scientific computation (Vuduc and Choi, 2013). GPUs typically contain 100s to 1,000s of processing elements (cores) that can perform simple computations in parallel. Parallelization of the particle filtering problem can be implemented straightforwardly by taking advantage of the independence of the particles. The lack of interaction between particles allows processing elements to handle particles or groups of particles without incurring overhead costs related to exchanging information among particles. In contrast, the HMM geolocation approach is less amenable to straightforward parallelization and is thus less likely to benefit from modern hardware acceleration approaches.

The primary objective of this work was to develop an efficient geolocation method based on the PF for demersal fishes using archival tagging data. The approach builds from previous work on an HMM-based model (HMM Geolocation Toolbox, Liu et al. (2017)) and PF models (Royer et al., 2005; Andersen et al., 2007), and improves on some of the algorithmic deficiencies from these prior efforts. The computational approach is accelerated using GPUs, enabling significant speedup of the geolocation and rapid execution of the model with affordable desktop computing components. The PF geolocation package was developed in Python with CUDA for the accelerated sections and is available at https://github.com/cliu3/pf_geolocation. To the best of our knowledge, this work is the first to apply GPU-based parallelization to individual animal tracking applications and to introduce an open-source geolocation code for archival tagging based on the PF. In the following sections, we describe the specifics of the PF geolocation method and the implementation of hardware acceleration. We then present a skill assessment of the method using fixed location mooring tags and double-electronically-tagged Atlantic cod from the western Gulf of Maine. Finally we demonstrate an application of the approach by presenting geolocations of two cod.

2 Methods

2.1 The particle filter algorithm

Demersal fish geolocation can be described as a nonlinear filtering problem using the following state-space system (Royer et al., 2005):

$$\begin{aligned} \boldsymbol{x}^{(k)} &= f(\boldsymbol{x}^{(k-1)}), \\ \boldsymbol{y}^{(k)} &= g(\boldsymbol{x}^{(k)}) + \boldsymbol{e}_t. \end{aligned} \tag{1}$$

Here, $\boldsymbol{x}^{(k)}$ is the state variable (geographic horizontal location of the fish) at time $t = k\Delta t$ where Δt is the observation time step; $\boldsymbol{y}^{(k)}$ is the observation (temperature and depth recorded by the archival tag) at the concurrent time; f is a function describing the fish's horizontal movement; g is the observation function; and $\boldsymbol{e}_t$ is the observation error (tag sensor errors). The goal is to estimate the daily location distribution of the tagged fish, i.e., the unknown state series $\boldsymbol{x}$, which requires estimating a probability distribution series $p(\boldsymbol{x}^{(k)}|\boldsymbol{y}^{(0:k)})$, given the tag-recorded full observation series $\boldsymbol{y}^{(0:k)} = \{\boldsymbol{y}^{(0)}, \boldsymbol{y}^{(1)}, ..., \boldsymbol{y}^{(k)}\}$. This is achieved using Bayesian inference:

$$\begin{aligned} p(\boldsymbol{x}^{(k)}|\boldsymbol{y}^{(0:k-1)}) &= \int p(\boldsymbol{x}^{(k)}|\boldsymbol{x}^{(k-1)})p(\boldsymbol{x}^{(k-1)}|y^{(0:k-1)})\mathrm{d}\boldsymbol{x}^{(k-1)}, \\ p(\boldsymbol{x}^{(k)}|\boldsymbol{y}^{(0:k)}) &= \frac{p(\boldsymbol{y}^{(k)}|\boldsymbol{x}^{(k)})p(\boldsymbol{x}^{(k)}|\boldsymbol{y}^{(0:k-1)})}{p(\boldsymbol{y}^{(k)}|\boldsymbol{y}^{(0:k-1)})}, \end{aligned} \tag{2}$$

where the initial distribution $p(\boldsymbol{x}^{(0)}|\boldsymbol{y}^{(0)})$ is a Gaussian distribution centered at the release location of the tagged fish, with a small standard deviation of <50 m. A PF is an algorithm for estimating a state-space model in which a set of discrete samples in state space (referred to as particles) and weights indicating the relative importance of the particles are used to approximate the predicted distribution. With respect to the geolocation problem, each particle $\boldsymbol{x}_i^{(k)}$ where i is the particle index represents the possibility of the fish's horizontal location at discrete time k. Each particle has a corresponding weight ($w_i^{(k)}$) which quantifies that possibility. Given sufficiently large particle count, N, the particles collectively approximate the continuous probability distribution of the fish's location.

A likelihood function connects the observations and the corresponding hidden states at each discrete time k. Constructing the likelihood function requires a comparison between the environmental data from archival tagging and a regional environmental database. We used bottom water temperature, bathymetry, and tidal elevation output from the Northeast Coastal Ocean Forecasting System (NECOFS) (Beardsley et al., 2013; NECOFS, 2013), which was developed using the Finite-Volume Community Ocean Model (FVCOM) (Chen et al., 2006; Cowles et al., 2008). FVCOM utilizes unstructured triangular grids which enable variation in the horizontal resolution. In the NECOFS database, the horizontal resolution ranges from 5 km near the open boundary to 500 m along the coast and in the vicinity of persistent tidal mixing fronts. Values of bathymetry and bottom temperature are located at the vertices of the triangles. Previous skill assessment studies compared the NECOFS-estimated bottom temperature with *in situ* bottom temperature measurements and reported strong agreement (Li et al., 2017; Liu et al., 2017). The likelihood function $L(\boldsymbol{x}, t)$ is derived from a statistical comparison of environmental data from the tag and from the FVCOM database over a tolerance interval following Le Bris et al. (2013); Liu et al. (2017); Zemeckis et al. (2017):

$$L_{dt}(\boldsymbol{x}) = \int_{z-\Delta z}^{z+\Delta z} N\Big(z; \mu_z(\boldsymbol{x}), \sigma_z(\boldsymbol{x})\Big) \mathrm{d}z \times \int_{T-\Delta T}^{T+\Delta T} N\Big(T; \mu_T(\boldsymbol{x}), \sigma_T(\boldsymbol{x})\Big) \mathrm{d}T, \tag{3}$$

where Δz and ΔT are the tag measurement error for depth and temperature, respectively; z and T are daily bottom depth and the associated temperature determined from the tag data; $N(\mu, \sigma^2)$ is a normal distribution function of mean μ and standard deviation σ, and μ_z and μ_T are NECOFS depth and temperature. The standard deviations of bathymetry $\sigma_z(\boldsymbol{x})$ and temperature $\sigma_T(\boldsymbol{x})$ were determined using the NECOFS depth and temperature values from the neighboring vertices of $\boldsymbol{x}$ on the unstructured grid. Subsequently, likelihood values are assigned a value of zero at locations where the possible tidal range interval estimated from NECOFS does not include the range of tidal signal detected from the tag data (see Liu et al. (2017) for details). The known recapture location was also incorporated in the likelihood function to influence movement towards this location over the last several time steps, by confining the likelihood distribution within a circle of decreasing radius R_t around the reported recapture location, and the radius is informed by the remaining time until recapture, and the typical swimming speed of the species v_m, until the radius equals the reported uncertainty radius r_u associated with the recapture location:

$$R_t = \max(r_u, 0.5v_m(T - t)). \tag{4}$$

The likelihood approach is described in detail in Liu et al. (2017) and was implemented in MATLAB in the HMM Geolocation toolbox. For the present work, the routines that construct the daily likelihood function were converted to Python and are incorporated in the PF geolocation package.

There are four main steps in the PF geolocation scheme: release, prediction, update, and resampling (Fig. 1). In the first step, the particles are initiated at the release location of the fish (Fig. 1a). This occurs only at the beginning of the simulation. The remaining three steps are repeated each day of the geolocation and are implemented in this study following the basic PF approach of Royer et al. (2005) and Andersen et al. (2007) and are described in detail below.

The prediction step models the horizontal movement of the fish and represents behavior (see Fig. 1b). This movement is approximated here by a random walk and was modeled directly for each particle using:

$$\widetilde{\boldsymbol{x}}_i^{(k)} = \boldsymbol{x}_i^{(k-1)} + \frac{\Delta t}{\delta t}\boldsymbol{R}\sqrt{2D_m\delta t}, \tag{5}$$

where i is the particle index, $\Delta t = 24$ h is the time interval between observations, δt is the prediction sub-step, $\boldsymbol{R}$ is drawn from the standard normal distribution (mean $= 0$; s.d. $= 1$) representing the process error, and D_m is a diffusion coefficient corresponding to the behavior state m. Approximating fish movement behavior via random walk is common and estimated movement can encompass a range of possible mechanisms and behaviors, both in geolocation applications (e.g., Sibert et al., 2003; Andersen et al., 2007; Nielsen and Sibert, 2007; Pedersen et al., 2008, 2011a; Galuardi and Lam, 2014; Braun et al., 2018) and estimating fish movements in the context of spatial stock structure (e.g., Sibert et al., 1999; Goethel et al., 2011; Schwarz, 2014). We selected a prediction sub-step of $\delta t = 1$ h which prevented particle displacements from exceeding the FVCOM mesh resolution along the coast. The values used for the diffusivity coefficients D_m, are species-specific and are tied to discrete behavior states. The behavior state is established based on the detection and duration of a tidal signal in the tag data on a given day following the approach used in our HMM geolocation package (Liu et al., 2017; Zemeckis et al., 2017) based on the premise that a tidal signal is more discernible in low activity fish when they are sedentary on the bottom, and the diffusivity coefficient values were determined considering the typical swimming speed of the species (e.g., Fernö et al., 2011). For Atlantic cod, we allowed the behavior state m to be sedentary (low activity, 13 h tidal signal, $D_m = 1$ km^2 day^{-1}), intermediate (moderate activity, 5 h tidal signal, $D_m = 5$ km^2 day^{-1}), or migratory (high activity, no tidal signal, $D_m = 10$ km^2 day^{-1}).

A rigorous boundary treatment was implemented to conserve the number of particles in the simulation by preventing particles from crossing onto land. To determine if a particle moved onto land during a prediction sub-step (δt), a nearest-neighbor search was performed to find the two FVCOM mesh vertices nearest the new particle location. A particle that is not contained within any of the triangular cells that are connected to these two vertices was considered to have exited the domain and is subsequently reset to its prior position within the domain at the previous prediction sub-step (Fig. 2a). Conversely, a particle that is inside any of the triangular cells that are connected to the two vertices nearest the particle was considered to be in the domain and the new particle location is retained (Fig. 2b). This boundary treatment approximates a reflecting boundary condition, which is appropriate for modeling fish movements (Sibert et al., 1999). In the serial CPU version of the code, the nearest neighbor search is performed using a k-d tree algorithm (Maneewongvatana and Mount, 1999) from the SciPy Python package (Jones et al., 2001). The k-d tree is an efficient search algorithm that is optimized for the CPU. For the present work it is considerably faster than a brute-force nearest neighbor search, providing a factor of 35 speedup in benchmark testing.

In the update step, particle weights are first drawn from the likelihood function $L(\boldsymbol{x}, t)$ evaluated at particle locations $\boldsymbol{x}_i^{(k)}$ and time $t = k\Delta t$:

$$\widetilde{w}_i^{(k)} = L(\boldsymbol{x}_i^{(k)}, t), \tag{6}$$

The particle likelihood values are computed using data that is stored discretely on the horizontal unstructured grid of the NECOFS database. Execution of eq. (6) requires interpolating $L(\boldsymbol{x}, t)$ onto each particle location. For this work we use a routine for bilinear interpolation on triangular grids provided in the Python package Matplotlib (Hunter, 2007). The particle weights are then normalized into the range $0 \leq w_i^{(k)} \leq 1$

$$w_i^{(k)} = \widetilde{w}_i^{(k)} / \sum_i \widetilde{w}_{(k)}^{(k)}, \tag{7}$$

to give the resulting posterior probability distribution at time t (Fig. 1c).

In the last step of the daily iteration, particles are resampled according to the particle weights ($w_i^{(k)}$), such that particles with low weights are removed and replaced by those with higher weights and particle numbers are reordered in the new set of particles so that they are proportional to their weights (Fig. 1d). The resampling is implemented following the approach of Labbe (2016) and is demonstrated in Fig. 3 for a simple simulation with $N = 10$ particles. In the first step, a cumulative density function (cdf; blue line) is constructed using the normalized weights ($w_j^{(k)}$). The cdf is then divided into N equal divisions where N is the number of particles and a random offset is used to displace these divisions (Fig. 3, green arrows). The N particles identified by the green arrows in the cdf curve are then selected for resampling. Note that the particle multiplicity may be greater than one. For the cdf and divisions shown in Fig. 3, the selected set of particles is $I = \{0, 0, 1, 3, 4, 4, 6, 8, 8, 9\}$. The particles with multiplicity greater than unity $\{0, 4, 8\}$ are particles with greater weight. The particles with lower weights $\{2, 5, 7\}$ will be re-initialized at the locations of particles $\{0, 4, 8\}$, respectively. The particle position histories are transferred using the indexing array I so that particles initialized to a new location carry the location time series of the particle in that location at time t:

$$\boldsymbol{x}_i^{(j)} = \widetilde{\boldsymbol{x}}_{\boldsymbol{I}_i}^{(j)} \qquad j = 0...k. \tag{8}$$

To conduct this step we used the systematic resampling function from the package FilterPy (Labbe, 2016) to generate an index array $\boldsymbol{I}$ to the particles that have been chosen for resampling such that the numbers of the indices to the particles before resampling equals these particles' weights:

$$P(\boldsymbol{I}_i = j) = w_j. \tag{9}$$

After the model has been integrated from release to recapture, the estimated most probable track (MPT) is determined. The MPT represents the track of the particle with the highest overall importance score, defined as the product of the weight at the last time step and the sum of the weights from the first to the second to last time steps. The index of the particle associated with the MPT is given by

$$I_{MPT} = \arg\max_i \,(w_i^{(T)} \sum_{k=0}^{T-1} w_i^{(k)}) \tag{10}$$

where T is the last time step of the filter. In addition to the MPT, daily posterior probability distributions of the fish are reconstructed from the horizontal distribution of particles using non-parametric kernel density estimation. These may also be interpreted as the uncertainty distribution around the most probable track and may be useful in interpreting the results.

A summary of the work flow in the present PF geolocation algorithm is provided in the table below.

1. Initialize the particles by placing them at the release location of the tagged fish (Fig. 1a).
2. FOR each day the fish is at large, do steps (a)–(c):
    (a) Predict: move the particles horizontally using a random walk and ensure that particles do not exit the domain (Fig. 1b).
    (b) Update: weight the particles by interpolating the observation likelihood function to the particles, and normalize the weights (Fig. 1c).
    (c) Resample: remove particles with lower weight and replace them by those with higher weight (Fig. 1d).
3. Construct the overall probability distribution.
4. Determine the most probable track (MPT)

2.2 GPU parallelization

The PF geolocation algorithm was first implemented as serial CPU code. To achieve acceleration of the geolocation computation, the serial CPU code was parallelized by taking advantage of the significant computing capabilities of modern GPUs, specifically those manufactured by NVIDIA. For this we used the PyCUDA package (Klöckner et al., 2012), a Python library that provides access to the NVIDIA CUDA parallel computation platform (Nickolls et al., 2008). In the CUDA platform, memory spaces on the host (CPU) and the device (GPU) are handled separately, and data must be available in the device memory for the GPU to perform computations. Transfers of data between the host and the device are explicitly programmed and must be carefully planned because they can incur a significant overhead. Functions that are submitted to the GPU for parallel execution are referred to as kernels and are written in CUDA C, a variant of the C programming language.

Here we describe the details of the GPU-accelerated version of the PF geolocation, hereby referred to as the GPU code. We refer explicitly to the three primary steps of the PF algorithm: prediction, update, and resample (Fig. 4). During the initial benchmarking and profiling of the serial CPU code, the nearest neighbor search was identified to be the most computationally intensive component. Therefore, the GPU implementation focuses primarily on the parallelization of the prediction step. In the GPU code, the x and y arrays representing the horizontal coordinates of the particles are first initiated on the host at the reported release location of the fish. These arrays are then transferred to the global memory on the device. In the prediction step, random numbers required for the random walk are generated on the device using an intrinsic function provided by PyCUDA and the particle positions are updated on the device. To apply the boundary conditions, a brute force nearest neighbor search is used. This required two separate kernels, one to determine the mesh elements surrounding the two mesh vertices nearest to each particle and a second to determine if the particle resides within any of these elements. In the update step, the x and y arrays are transferred from the host to the device. The likelihood distribution L is then interpolated onto the particle positions (x, y) on the host to compute particle weights. These interpolated weights are subsequently transferred to the device. In the resampling step, the index array I is generated on the host. A kernel was written to re-arrange the x and y arrays according to I on the GPU. Arrays x and y are subsequently transferred from the device to the host and stored in an array that is archived to an external data file.

3 Validation and Performance Results

3.1 Tag Data and Skill Metrics

Following Liu et al. (2017), two types of tag data were used for assessing the skill of the PF geolocation method. First, bottom-mooring tags which challenge the model to maintain a fixed position over time. A total of 14 Star-ODDI DSTs were moored on the bottom of different known fixed locations in Massachusetts Bay, Ipswich Bay, and Jeffreys Ledge between 2010 and 2015 (Fig. 5). The second set of tag data is derived from double-tagged Atlantic cod. During a study conducted from 2010 to 2012, individual Atlantic cod were tagged with both Star-ODDI milli-L DSTs and Vemco V16P-6H acoustic transmitters in the Spring Cod Conservation Zone (SCCZ; Fig. 5), located in northern Massachusetts Bay in the western Gulf of Maine, USA (See Dean et al. (2014); Zemeckis et al. (2014, 2017) for full tagging methods). Importantly, ten double-tagged fish were recaptured and the acoustic transmitters carried by these fish provide an independent set of location estimates accurate to <10 m in the SCCZ and <1 km otherwise, when the tagged fish were detected within the receiving range of acoustic receiver arrays. Since the position of the animal is accurately known at these discrete locations while at-liberty, these data can be directly incorporated in the skill assessment of the geolocation method. Three error metrics were used to evaluate the model skill. The first metric (E1) is the distance between known locations and the location of the nearest modeled particle on the day of detection. The second metric (E2) is the distance between known locations and the position of the fish along the MPT on the day of detection. The third metric (E3) is whether the known location falls within the 95% credible area of the same-day probability distribution, reconstructed from all particles. The 95% credible area is defined such that the sum of the probability within the area is 95% of the total probability.

3.2 Sensitivity to the number of particles

A study was carried out to examine the influence of the particle count N on the geolocation. Seven sets of geolocation tests were conducted using particle numbers N ranging from 2×10^3 to 400×10^3. For each N, an ensemble of 30 runs were made with identical parameters. Both the E1 and E2 metrics were computed for each model run and used to evaluate convergence of the solution with N. The mean value of E1 over the ensemble decreases rapidly with increasing particle count to around 200×10^3 particles and then begins to asymptote towards a fixed value with further increases in particle count (Fig. 6a). Statistics for the root mean square (RMS) of the E2 metric were also evaluated. The median value of the RMS of the E2 does not depend on particle count (Fig. 6b). However, the variation of the RMS of the E2 decreases with particle count up to around $N = 200 \times 10^3$ particles and remains fairly static for $N \geq 200 \times 10^3$. This indicates that the geolocation reaches a particle-converged solution at $N \sim 200 \times 10^3$ particles. Results from both the E1 and E2 metrics indicate that using $N = 200 \times 10^3$ particles is an optimal choice for both accuracy in particle filtering and computational load. This particle number was thus used in all experiments.

3.3 Skill Assessment

A skill assessment based on the E2 metric (error in the MPT) and E3 (whether known locations fall within 95% credible areas) was conducted using tags from 14 mooring and 10 double-tagged cod representing 984 d of data. The MPT estimations of the PF geolocation method for the mooring DST locations had an RMS error of 14.95 km, and the error range was 0.01–27.53 km. The median MPT error for all mooring tags was 9.71 km (Table 1), and 61.9% of the known locations fell within the 95% credible areas of the same-day posterior probability distributions. For the 10 double-tagged cod with high-resolution positions determined by acoustic telemetry detections, the RMS error of the same-day MPT estimation was 18.19 km and the median error was 6.0 km. The error range was 0.29–46.77 km (Table 1). All known locations fell within the 95% credible areas of the same-day posterior probability distributions. These results indicate that the MPT determined using PF geolocation method was able to provide accurate location estimates typically on the horizontal scale of <18 km.

3.4 Benchmarking and profiling

Wall clock time for the PF geolocation code executed on serial CPU and GPU was evaluated on a high-performance computing cluster. Each node in the cluster was equipped with an Intel Core i7-950 CPU and an NVIDIA GeForce GTX 560 Ti GPU. A range of problem sizes from 12,500 to 200,000 particles was tested on data from a tagged cod with 56 d at liberty. Using the Python profiling module "cProfile", the total runtime was decomposed into fractions spent in the prediction, update, and resampling steps. Profiling demonstrated that the majority of the computational time (>97% for all serial CPU cases, >52% for all GPU cases) was spent in the prediction step (Fig. 7a). The relationship between runtime and particle number is approximately linear for both the serial CPU and GPU implementations, and the speedup that the GPU implementation provides over the serial CPU approach increases with increasing particle count N, ranging from a factor of 19.0 to 48.9 (Fig. 7b). Furthermore, in both CPU and GPU implementations, time spent in prediction and resampling steps increases as N increases, whereas the time spent in other parts is nearly constant regardless of particle counts, resulting in decreasing portion of total time (Fig. 7a). Thus, accelerating the prediction step through GPU parallelization effectively reduced overall runtime of PF geolocation.

A performance study of the PF geolocation method was also conducted on a wide range of NVIDIA GPUs. The set included products from four generations of hardware microarchitectures (Fermi, Kepler, Maxwell, and Pascal) and both consumer (GeForce) and high-performance computing (Tesla) lines (Table 2) and represents a factor of 10 in the range of theoretical single-precision performance. These tests were conducted using 200,000 particles and tag data from the same Atlantic cod used in the CPU-GPU comparison study. CUDA Toolkit version 9.1 was used to compile kernels for all tests with the exception of those run on the legacy Fermi generation GPUs which are not supported beyond CUDA 8.0. CUDA 9.1 contains optimizations in routines used by the PF package which enable a 10% increase in performance over CUDA 8.0. Throughput, measured as the number of days at liberty that can be geolocated in an hour of compute time using 200,000 particles, was used as the performance metric. The throughput on the serial CPU code was 6.4 d/h. The results suggest that performance of the model generally correlates with the theoretical performance of the hardware (GFLOPS) and that throughput is enhanced on the newer architectures with greater memory bandwidth. The greatest performance was achieved on the NVIDIA Volta V100, a powerful GPU aimed at deep learning applications with an approximate price of $10,000 USD. Such high end hardware, however, is not necessary. The GeForce GTX 1050 is capable of geolocating 483 d of fish movement in under an hour of wall clock time. The 1050 is commonly specified in laptops and entry-level desktops and sells for ∼$100 USD, considerably less than the cost of an archival storage tag. In summary, this study indicates that the GPU enables routine PF geolocations to be performed on affordable consumer-grade computers.

3.5 Geolocation of Atlantic cod in western Gulf of Maine

To demonstrate the capabilities of the PF geolocation package, we applied it to the geolocation of two Atlantic cod. For each fish, the estimated MPT and daily posterior probability distributions for each day the fish was at liberty are shown in Figs. 8 and 9. The reconstructed depth and temperature time series from the MPT are generally in good agreement with the raw tag data (Fig. 10).

Cod #13 was released on 11 May 2010 and recaptured on 21 Nov. 2010. During its 194 d at large, the tidal fitting algorithm identified 151 d as low activity, 32 d as moderate activity, and 11 d as high activity. The cod migrated southward from the tagging location and remained in the region between Stellwagen Bank and Cape Cod Bay for a prolonged period of time (approximately from day 20 to 120) before heading north to the location of its recapture (Fig. 8). The prolonged period of sedentary behavior was also evident in the depth time series data recorded by the DST (Fig. 10a). The considerable time spent on Stellwagen Bank, away from the release and recapture locations represents information that would not be possible to determine from conventional tagging which can only inform release and recapture locations..

Cod #17 was released on 18 June 2010 and recaptured on 29 Aug. 2010 (72 d at large); 26, 33, and 13 days were classified as low, moderate, and high activity days, respectively. As the fish migrated northward after day 30 towards the recapture location, the posterior probability distribution exhibited a bimodal pattern, suggesting two plausible trajectories: one that extended directly northward and the other that took a more circuitous route to the east around the southern portion of Jeffreys Ledge (Fig. 9). In an ensemble of model runs, MPTs along both of these trajectories were observed, although the circuitous route was the dominant solution. The MPT from the particular model run shown in Fig. 9 follows this circuitous second trajectory.

4 Discussion

The open source PF geolocation package presented in this work was developed with the goal of making geolocation analyses more accessible to fisheries researchers who conduct archival tagging studies on demersal fishes. The kernel of the solver represents an implementation of the basic filter outlined in Andersen et al. (2007) combined with the likelihood function approach developed in our prior geolocation work (Liu et al., 2017). The implementation of a rigorous boundary treatment scheme and GPU parallelization enables this software package to estimate movement in regions with complex coastline geometry and provides rapid solutions using consumer grade computer hardware readily available to researchers.

Results of MPT errors from PF geolocation of both mooring and double-tagging validation tests were similar to those obtained using the HMM geolocation toolbox (Liu et al., 2017): RMS error for mooring tags were 14.95 km with PF and 11.07 km with HMM, while for double-tagged cod errors were 18.19 km with PF and 21.87 km with HMM. The PF geolocation exhibited slightly better overall skill in the geolocations of double-tagged fish, but with shorter runtime. For six out of ten double-tagged fish, the PF geolocation code outperformed the HMM geolocation toolbox in median geolocation error by 0.45–34.8 km. These errors were not found to decrease substantially with further increases in the number of particles in the PF or refinement of the mesh in HMM. This indicates that for this specific combination of species, tag type, and the given environmental database, we may be at the limit of estimation accuracy that can be provided by state-space methods. The PF geolocation performances are similar to or better than other geolocation efforts. For example, Hunter et al. (2003) and Thorsteinsson et al. (2012) used mooring tags fixed at known locations to validate their tidal-based method and reported average error of 15.7 ± 3.5 km and 18.91 km, respectively. Double-tagging studies of sharks (Teo et al., 2004; Winship et al., 2012) found errors >0.5°(approximately 55 km). A recent HMM-based geolocation study of shark species reported median errors of 66–150 km compared to known locations with accuracy of <10 km (Braun et al., 2018). Precision obtained with this methods is the among highest documented in tracking marine animals.

The GPU implementation of the PF geolocation package achieves up to 75× the speed of the serial CPU implementation on the affordable, consumer-grade NVIDIA GTX 1050, and up to 266× on a high-end Tesla V100 GPU. The runtime of PF geolocation of a 210 d track is well under an hour running on a typical consumer grade NVIDIA GPU with minimal specifications. This acceleration factor may be even higher if the suboptimal brute force nearest neighbor search were used in the CPU implementation rather than the optimized k-d tree algorithm. The significant acceleration achieved through GPU parallelization eliminates the requirements for costly specialized hardware. For comparison with the computation time of other geolocation applications on consumer-grade hardware, Pedersen et al. (2011b) reported that the finite-element geolocation method they developed takes on the order of days to estimate a 294 d track on a 1.4 GHz laptop and the recently published HMM-based geolocation package HMMoce (Braun et al., 2018) written in R takes nearly a full day to run a 134 d track on a quad-core personal computer. Parts of the current PF geolocation package may be further parallelized, but doing so is not likely to result in any significant improvement in performance. For example, in the current method, the prediction step is the most computationally intensive. Brute-force nearest neighbor search represents an embarrassingly parallel algorithm and the GPU implementation is much faster than the optimized k-d tree on serial CPU. The k-d tree is an optimized algorithm for serial execution that may provide 35× speedup over serial brute-force, but it is not suitable for parallel execution (Hering, 2013). Implementing this k-d tree on GPU would require a considerable undertaking with no guarantee of performance gain over the brute-force algorithm. As another example, the PF resampling in the current method is not parallelized. While parallelized PF resampling algorithms have been proposed (McAlinn et al., 2016), the benefit to the overall performance would be nominal, because resampling accounts for only <1.2% of the total runtime, and Amdahl's Law (Amdahl, 1967) predicts a maximum speedup of only $S = 1/\left(1 - \frac{1.2}{100}\right) \approx 1.2\%$.

Most approaches to marine animal geolocation do not place emphasis on the boundary treatment. Simple boundary schemes, such as masking out values on grid points representing land, may be sufficient for estimating large-scale movements of pelagic animals in which case the influence of land boundaries may be negligible, but these simple schemes cannot adequately handle the estimations of movements of coastal species in regions with complex land boundaries. In the present work, we use the unstructured triangular mesh of NECOFS database which provides significantly better resolution of the coastline compared with structured grid approaches (Chen et al., 2006). This enables us to implement a robust reflection boundary scheme in the PF geolocation package that effectively prevents particles from moving onto or crossing over land and models the fish movements more realistically. As an alternative boundary treatment, particles that move out of the domain can be eliminated. This is equivalent to an absorbing boundary condition which is not appropriate for the land-ocean boundary in modeling marine animal movements (Sibert et al., 1999). The PF geolocation package can potentially be adapted to work with other oceanographic databases that provide bathymetry and bottom temperature data for other regions. Since the boundary treatment in the PF geolocation package is dependent on the grid of the FVCOM bottom temperature data, using data from other databases requires re-implementation of the boundary treatment scheme. Bottom temperature data from many of the oceanographic databases are based on popular ocean models such as ROMS or HYCOM that use curvilinear grids, which would make the particle-based boundary treatment scheme easier to implement than the triangular grid of FVCOM (e.g., Sumner et al., 2009).

The PF geolocation results include the daily posterior probability distributions and the MPT. Due to the stochastic nature of the simulation, two runs with identical parameters will not produce identical results. The particle number sensitivity experiments indicate that using larger particle numbers will decrease the variability of the outcome (MPT), but there is a limit beyond which further increase in the particle count will not provide further convergence of the MPT over an ensemble of runs. As an alternative point estimate metric, maximum *a posteriori* (MAP) (Saha et al., 2009) may provide less stochastic location track estimates, but the high computational complexity is likely prohibitive, especially when the particle number is large. In addition, being a single sample of all the particles, the MPT ensures that the movement model is strictly followed, making MPT a more plausible track than one estimated by the MAP. It should be noted that, as the daily posterior distributions are largely consistent across an ensemble of runs using a fixed model setup, any point estimate metric including the MPT should not be the sole information to be considered when interpreting and understanding the movements of the tagged individual.

The PF geolocation method uses the random walk to model individual movements, because the random walk and the equivalent Fokker-Planck diffusion model are widely accepted as appropriate for the spatial and temporal scales corresponding to tagging studies of fishes (e.g., Sibert et al., 1999; Andersen et al., 2007; Pedersen et al., 2008; Goethel et al., 2011), and for animal movement modeling using the particle filter (e.g., Andersen et al., 2007; Tremblay et al., 2009; Dowd and Joy, 2011; Rakhimberdiev et al., 2015). The random walk was also the choice of the movement model in many popular geolocation software packages for marine animals (e.g., hmmgeolocation: Pedersen et al. 2008; Wildlife Computers GPE3: based on Pedersen et al. 2011a; TrackIt: Lam et al. 2010; HMMoce: Braun et al. 2018). Alternative movement models, such as Lévy flight, have been shown to have a negligible effect in geolocation applications compared to the random walk (Thygesen and Nielsen, 2009). Furthermore, the random walk model is effectively being used as a prior on possible moves and the estimated movement is being updated by the data very frequently, therefore the performance may be less sensitive to the choice of the movement model. Given the reasonably good performance indicated by the validation results, implementing a different movement model may unnecessarily increase the complexity of the geolocation method. In geolocating the double-tagged cod, the diffusion coefficient was estimated from the measured modal swimming speed of Atlantic cod (0.1–0.4 body lengths per second, Fernö et al. 2011). Given that the lengths of the double-tagged cod are in the range of 70–110 cm, the appropriate diffusion coefficient was estimated to be 1 km^2 day^{-1} for the low activity level, considering a small, slow fish (70 cm, 0.05 body lengths per second) and 10 km^2 day^{-1} for the high activity level, considering a larger, faster fish (110 cm, 0.1 body lengths per second), using the equation $D = \rho v^2/2$, where v is the swimming speed and $\rho = 6$ h is an assumed decorrelation time (Pedersen, 2007). The PF geolocation results were found to be sensitive to values selected for the diffusion coefficients. We performed the double-tagged cod validation with the increased diffusion coefficient values for low, moderate, and high activity levels of 5, 25, and 50 $km^2 \, day^{-1}$, and the overall median error increased 23 km. The PF geolocation method may be further improved to be capable of estimating unknown parameters, such as the diffusion coefficient, based on a maximum likelihood approach. For example, Andersen et al. (2007) proposed using Random Walk Metropolis-Hastings combined with the PF to approximate the probability distributions of the unknown parameters. Parameter estimation for the PF is an active field of research (see e.g., Kantas et al., 2015), and developing the GPU implementation of the optimal approach is a promising future direction.

The main contribution of this work is the successful development of a GPU-accelerated open-source geolocation package using archival tagging data that can be executed on affordable computers. The source code is available on a GitHub repository at `https://github.com/cliu3/pf_geolocation`, where instructions for users and an example case that can be executed are also available. Researchers can further adapt the source code for applications to other species and regions.

Acknowledgements

The authors thank the Center for Scientific Computing and Visualization Research (CSCVR) at UMass Dartmouth for providing an array of hardware for benchmarking and testing. We are grateful for two anonymous reviewers who provided helpful feedback to the manuscript. Cod tagging research in the Spring Cod Conservation Zone was conducted in collaboration with the Massachusetts Division of Marine Fisheries and supported by the United States Fish and Wildlife Service through the Sportfish Restoration Act and the Massachusetts Marine Fisheries Institute. Funding for the research conducted as part of this manuscript was provided by NOAA Saltonstall-Kennedy Grant award NA15NMF4270267.

Author Contributions

CL designed methodology, developed the PF geolocation package, and performed analyses; DZ collected the cod DST and acoustic telemetry data; CL and GC led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

References

Amdahl, G., 1967. Validity of the single processor approach to achieving large-scale computing capabilities, in: Proceedings of the American Federation of Information Processing Societies, volume 30, 483–485, URL http://doi.acm.org/10.1145/1465482.1465560.

Andersen, K., Nielsen, A., Thygesen, U., Hinrichsen, H.-H., Neuenfeldt, S., 2007. Using the particle filter to geolocate Atlantic cod (*Gadus morhua*) in the Baltic Sea, with special emphasis on determining uncertainty, Can. J. Fish. Aquat. Sci. 64, 618–627, doi:10.1139/f07-037.

Beardsley, R. C., Chen, C., Xu, Q., 2013. Coastal flooding in Scituate (MA): A FVCOM study of the 27 December 2010 nor'easter, J. Geophys. Res.-Oceans 118 (11), 6030–6045, doi:10.1002/2013JC008862, URL http://onlinelibrary.wiley.com/doi/10.1002/2013JC008862/abstract.

Braun, C. D., Galuardi, B., Thorrold, S. R., 2018. HMMoce: An R package for improved geolocation of archival-tagged fishes using a hidden Markov method, Methods in Ecology and Evolution doi:10.1111/2041-210X.12959.

Brickman, D., Thorsteinsson, V., 2008. Geolocation of Icelandic cod from DST data using a modified particle filter method, ICES CM 2008/P.09 .

Chen, C., Beardsley, R. C., Cowles, G., 2006. An Unstructured Grid, Finite-Volume Coastal Ocean Model (FVCOM) System, Oceanography 19 (1), 78, URL http://www.tos.org/oceanography/archive/19-1_chen.html.

Coleman, K. E., 2015. Understanding the winter flounder (*Pseudopleuronectes americanus*) southern New England/Mid-Atlantic stock through historical trawl surveys and monitoring cross continental shelf movement, MS thesis, Rutgers University, New Brunswick, NJ.

Cowles, G. W., Lentz, S. J., Chen, C., Xu, Q., Beardsley, R. C., 2008. Comparison of observed and model-computed low frequency circulation and hydrography on the New England Shelf, J. Geophys. Res.-

Oceans 113 (C9), C09015, doi:10.1029/2007JC004394, URL http://www.agu.org/pubs/crossref/2008/2007JC004394.shtml.

Dean, M. J., Hoffman, W. S., Zemeckis, D. R., Armstrong, M. P., 2014. Fine-scale diel and gender-based patterns in behaviour of Atlantic cod (*Gadus morhua*) on a spawning ground in the Western Gulf of Maine, ICES J. Mar. Sci. 71 (6), 1474–1489, doi:10.1093/icesjms/fsu040, URL http://icesjms.oxfordjournals.org/cgi/doi/10.1093/icesjms/fsu040.

Dowd, M., Joy, R., 2011. Estimating behavioral parameters in animal movement models using a state-augmented particle filter, Ecology 92 (3), 568–575, doi:10.1890/10-0611.1.

Fernö, A., Jørgensen, T., Løkkeborg, S., Winger, P. D., 2011. Variable swimming speeds in individual Atlantic cod (*Gadus morhua* L.) determined by high-resolution acoustic tracking, Marine Biology Research 7 (3), 310–313, doi:10.1080/17451000.2010.492223, URL http://www.tandfonline.com/doi/abs/10.1080/17451000.2010.492223.

Galuardi, B., Lam, C. H. T., 2014. Telemetry Analysis of Highly Migratory Species, in: Cadrin, S. X., Kerr, L. A., Mariani, S. (Eds.), Stock Identification Methods (Second Edition), Academic Press, San Diego, 447–476.

Goethel, D. R., Quinn, T. J., Cadrin, S. X., 2011. Incorporating spatial structure in stock assessment: movement modeling in marine fish population dynamics, Rev. Fish. Sci. 19 (2), 119–136, doi:10.1080/10641262.2011.557451.

Goodrum, M. A., Trotter, M. J., Aksel, A., Acton, S. T., Skadron, K., 2011. Parallelization of Particle Filter Algorithms, in: Varbanescu, A. L., Molnos, A., Nieuwpoort, R. v. (Eds.), Computer Architecture, Springer Berlin Heidelberg, no. 6161 in Lecture Notes in Computer Science, 139–149, URL http://link.springer.com.silk.library.umass.edu/chapter/10.1007/978-3-642-24322-6_12.

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., Nordlund, P. J., 2002. Particle filters for positioning, navigation, and tracking, IEEE Transactions on Signal Processing 50 (2), 425–437, doi:10.1109/78.978396.

Hendeby, G., Karlsson, R., Gustafsson, F., 2010. Particle Filtering: The Need for Speed, EURASIP Journal on Advances in Signal Processing 2010, 22:1–22:9, doi:10.1155/2010/181403, URL http://dx.doi.org/10.1155/2010/181403.

Hering, T., 2013. Parallel Execution of kNN-Queries on in-memory KD Trees, in: Proc. of 15th GI Symposium on Business, Technology & Web (BTW'13), Magdeburg, Germany, 257–266, URL http://cs.emis.de/LNI/Proceedings/Proceedings216/257.pdf.

Hunter, E., Aldridge, J. N., Metcalfe, J. D., Arnold, G. P., 2003. Geolocation of free-ranging fish on the European continental shelf as determined from environmental variables - I. Tidal location method, Marine Biology 142 (3), 601–609, doi:10.1007/s00227-0984-5. WOS:000182387900021.

Hunter, J. D., 2007. Matplotlib: A 2d graphics environment, Computing In Science & Engineering 9 (3), 90–95, doi:10.1109/MCSE.2007.55.

Hussey, N. E., Kessel, S. T., Aarestrup, K., Cooke, S. J., Cowley, P. D., Fisk, A. T., Harcourt, R. G., Holland, K. N., Iverson, S. J., Kocik, J. F., Flemming, J. E. M., Whoriskey, F. G., 2015. Aquatic animal telemetry: A panoramic window into the underwater world, Science 348 (6240), 1255642, doi:10.1126/science.1255642, URL http://science.sciencemag.org/content/348/6240/1255642.

Jones, E., Oliphant, T., Peterson, P., others, 2001. SciPy: Open source scientific tools for Python, URL http://www.scipy.org. http://www.scipy.org. [Online; accessed 2018-01-11].

Jonsen, I., Basson, M., Bestley, S., Bravington, M., Patterson, T., Pedersen, M., Thomson, R., Thygesen, U., Wotherspoon, S., 2013. State-space models for bio-loggers: A methodological road map, Deep Sea Res. Part II 88-89, 34–46, doi:10.1016/j.dsr2.2012.07.008, URL http://linkinghub.elsevier.com/retrieve/pii/S096706451200094X.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., 2015. On particle methods for parameter estimation in state-space models, Statistical Science 30 (3), 328–351, doi:10.1214/14-STS511, URL https://projecteuclid.org/euclid.ss/1439220716.

Klöckner, A., Pinto, N., Lee, Y., Catanzaro, B., Ivanov, P., Fasih, A., 2012. PyCUDA and PyOpenCL: A Scripting-Based Approach to GPU Run-Time Code Generation, Parallel Computing 38 (3), 157–174, doi:10.1016/j.parco.2011.09.001.

Labbe, R. R., 2016. Kalman and bayesian filters in python, URL https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python/. Available at https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python/. Accessed 2018-01-07.

Lam, C. H., Nielsen, A., Sibert, J. R., 2010. Incorporating sea-surface temperature to the light-based geolocation model TrackIt, Marine Ecology Progress Series 419, 71–84, doi:10.3354/meps08862.

Le Bris, A., Fréchet, A., Wroblewski, J. S., 2013. Supplementing electronic tagging with conventional tagging to redesign fishery closed areas, Fish. Res. 148, 106–116, doi:10.1016/j.fishres.2013.08.013, URL http://linkinghub.elsevier.com/retrieve/pii/S0165783613002063.

Li, B., Tanaka, K. R., Chen, Y., Brady, D. C., Thomas, A. C., 2017. Assessing the quality of bottom water temperatures from the Finite-Volume Community Ocean Model (FVCOM) in the Northwest Atlantic Shelf region, Journal of Marine Systems doi:10.1016/j.jmarsys.2017.04.001, URL http://www.sciencedirect.com/science/article/pii/S092479631630358X.

Liu, C., Cowles, G. W., Zemeckis, D. R., Cadrin, S. X., Dean, M. J., 2017. Validation of a hidden Markov model for the geolocation of Atlantic cod, Canadian Journal of Fisheries and Aquatic Sciences 74 (11), 1862–1877, doi:10.1139/cjfas-2016-0376, URL http://www.nrcresearchpress.com/doi/10.1139/cjfas-2016-0376.

Maneewongvatana, S., Mount, D. M., 1999. It's okay to be skinny, if your friends are fat, in: Center for Geometric Computing 4th Annual Workshop on Computational Geometry, URL https://cs.umd.edu/~mount/Papers/cgc99-smpack.pdf.

McAlinn, K., Katsura, H., Nakatsuma, T., 2016. Fully Parallel Particle Learning for GPGPUs and Other Parallel Devices, arXiv preprint arXiv:1212.1639 URL http://arxiv.org/abs/1212.1639.

Metcalfe, J. D., Arnold, G. P., 1997. Tracking fish with electronic tags, Nature 387 (6634), 665–666, doi: 10.1038/42622.

NECOFS, 2013. Northeast Coastal Ocean Forecasting System (NECOFS) Main Portal http://fvcom.smast.umassd.edu/necofs/. Accessed: 2018-01-08.

Nickolls, J., Buck, I., Garland, M., Skadron, K., 2008. Scalable Parallel Programming with CUDA, Queue 6 (2), 40–53, doi:10.1145/1365490.1365500, URL http://doi.acm.org/10.1145/1365490.1365500.

Nielsen, A., 2004. Estimating fish movement, Ph.D. thesis, Royal Vetenary and Argricultural University, URL http://geolocation.umassd.wikispaces.net/file/view/Nielson_2004.pdf.

Nielsen, A., Sibert, J. R., 2007. State–space model for light-based tracking of marine animals, Canadian Journal of Fisheries and Aquatic Sciences 64 (8), 1055–1068, doi:10.1139/f07-064.

Patterson, T., Thomas, L., Wilcox, C., Ovaskainen, O., Matthiopoulos, J., 2008. State–space models of individual animal movement, Trends in Ecology & Evolution 23 (2), 87–94, doi:10.1016/j.tree.2007.10.009.

Pedersen, M., Berg, C., Thygesen, U., Nielsen, A., Madsen, H., 2011a. Estimation methods for nonlinear state-space models in ecology, Ecol. Model. 222 (8), 1394–1400, doi:10.1016/j.ecolmodel.2011.01.007, URL http://linkinghub.elsevier.com/retrieve/pii/S0304380011000299.

Pedersen, M., Thygesen, U., Madsen, H., 2011b. Nonlinear tracking in a diffusion process with a Bayesian filter and the finite element method, Comput Stat Data Anal 55 (1), 280–290, doi:10.1016/j.csda.2010.04.018, URL http://linkinghub.elsevier.com/retrieve/pii/S0167947310001635.

Pedersen, M. W., 2007. Hidden Markov models for geolocation of fish, Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, URL http://etd.dtu.dk/thesis/200722/.

Pedersen, M. W., Righton, D., Thygesen, U. H., Andersen, K. H., Madsen, H., 2008. Geolocation of North Sea cod (*Gadus morhua*) using hidden Markov models and behavioural switching, Can. J. Fish. Aquat. Sci. 65 (11), 2367–2377.

Rakhimberdiev, E., Winkler, D. W., Bridge, E., Seavy, N. E., Sheldon, D., Piersma, T., Saveliev, A., 2015. A hidden Markov model for reconstructing animal paths from solar geolocation loggers using templates for light intensity, Movement Ecology 3 (1), 25, doi:10.1186/s40462-015-0062-5.

Righton, D., Mills, C., 2008. Reconstructing the movements of free-ranging demersal fish in the North Sea: a data-matching and simulation method, Mar. Biol. 153 (4), 507–521, doi:10.1007/s00227-007-0818-6, URL http://dx.doi.org/10.1007/s00227-007-0818-6.

Royer, F., Fromentin, J.-M., Gaspar, P., 2005. A state-space model to derive bluefin tuna movement and habitat from archival tags, Oikos 109 (3), 473–484, doi:10.1111/j.0030-1299.2005.13777.x, URL http://onlinelibrary.wiley.com/doi/10.1111/j.0030-1299.2005.13777.x/abstract.

Saha, S., Boers, Y., Driessen, H., Mandal, P. K., Bagchi, A., 2009. Particle based MAP state estimation: A comparison, in: 2009 12th International Conference on Information Fusion, 278–283.

Schwarz, C. J., 2014. Estimation of movement from tagging data, in: Cadrin, S. X., Kerr, L. A., Mariani, S. (Eds.), Stock Identification Methods (Second Edition), Academic Press, San Diego, 429–446.

Sibert, J. R., Hampton, J., Fournier, D. A., Bills, P. J., 1999. An advection–diffusion–reaction model for the estimation of fish movement parameters from tagging data, with application to skipjack tuna (*Katsuwonus pelamis*), Canadian Journal of Fisheries and Aquatic Sciences 56 (6), 925–938, doi:10.1139/f99-017, URL http://www.nrcresearchpress.com/doi/abs/10.1139/f99-017.

Sibert, J. R., Musyl, M. K., Brill, R. W., 2003. Horizontal movements of bigeye tuna (*Thunnus Obesus*) near Hawaii determined by Kalman filter analysis of archival tagging data, Fisheries Oceanography 12 (3), 141–151, doi:10.1046/j.1365-2419.2003.00228.x.

Sumner, M. D., Wotherspoon, S. J., Hindell, M. A., 2009. Bayesian Estimation of Animal Movement from Archival and Satellite Tags, PLoS One 4 (10), e7324, doi:10.1371/journal.pone.0007324, URL http://dx.doi.org/10.1371/journal.pone.0007324.

Teo, S. L., Boustany, A., Blackwell, S., Walli, A., Weng, K. C., Block, B. A., 2004. Validation of geolocation estimates based on light level and sea surface temperature from electronic tags, Mar. Ecol.

Prog. Ser. 283, 81–98, URL http://www.soest.hawaii.edu/oceanography/faculty/kweng/Teo2004_geolocation.pdf.

Thorsteinsson, V., Pálsson, O. K., Tómasson, G. G., Jónsdóttir, I. G., Pampoulie, C., 2012. Consistency in the behaviour types of the Atlantic cod: repeatability, timing of migration and geo-location, Mar. Ecol. Prog. Ser. 462, 251–260, doi:10.3354/meps09852, URL http://www.int-res.com/abstracts/meps/v462/p251-260/.

Thygesen, U. H., Nielsen, A., 2009. Lessons from a Prototype Geolocation Problem, in: Nielsen, J. L., Arrizabalaga, H., Fragoso, N., Hobday, A., Lutcavage, M., Sibert, J. (Eds.), Tagging and Tracking of Marine Animals with Electronic Devices, Springer Netherlands, no. 9 in Reviews: Methods and Technologies in Fish Biology and Fisheries, 257–276.

Thygesen, U. H., Pedersen, M. W., Madsen, H., 2009. Geolocating Fish Using Hidden Markov Models and Data Storage Tags, in: Nielsen, J. L., Arrizabalaga, H., Fragoso, N., Hobday, A., Lutcavage, M., Sibert, J. (Eds.), Tagging and Tracking of Marine Animals with Electronic Devices, Springer Netherlands, no. 9 in Reviews: Methods and Technologies in Fish Biology and Fisheries, 277–293, URL http://link.springer.com.silk.library.umass.edu/chapter/10.1007/978-1-4020-9640-2_17.

Tremblay, Y., Robinson, P. W., Costa, D. P., 2009. A Parsimonious Approach to Modeling Animal Movement Data, PLoS ONE 4 (3), e4711, doi:10.1371/journal.pone.0004711, URL http://dx.doi.org/10.1371/journal.pone.0004711.

Vuduc, R., Choi, J., 2013. A Brief History and Introduction to GPGPU, in: Modern Accelerator Technologies for Geographic Information Science, Springer, Boston, MA, 9–23, doi:10.1007/978-1-4614-8745-6_2, URL https://link.springer.com/chapter/10.1007/978-1-4614-8745-6_2.

Winship, A. J., Jorgensen, S. J., Shaffer, S. A., Jonsen, I. D., Robinson, P. W., Costa, D. P., Block, B. A., 2012. State-space framework for estimating measurement error from double-tagging telemetry experiments, Methods Ecol. Evol. 3 (2), 291–302, doi:10.1111/j.2041-210X.2011.00161.x.

Woillez, M., Fablet, R., Ngo, T.-T., Lalire, M., Lazure, P., de Pontual, H., 2016. A HMM-based model to geolocate pelagic fish from high-resolution individual temperature and depth histories: European sea bass as a case study, Ecological Modelling 321, 10–22, doi:10.1016/j.ecolmodel.2015.10.024, URL http://www.sciencedirect.com/science/article/pii/S0304380015005098.

Zemeckis, D., Liu, C., Cowles, G., Dean, M., Hoffman, W., Martins, D., Cadrin, S., 2017. Seasonal movements and connectivity of an Atlantic cod (*Gadus morhua*) spawning component in the western Gulf of Maine, ICES Journal of Marine Science 74 (6), 1780–1796, doi:10.1093/icesjms/fsw190.

Zemeckis, D. R., Hoffman, W. S., Dean, M. J., Armstrong, M. P., Cadrin, S. X., 2014. Spawning site fidelity by Atlantic cod (*Gadus morhua*) in the Gulf of Maine: implications for population structure and rebuilding, ICES J. Mar. Sci. 71 (6), 1356–1365, doi:10.1093/icesjms/fsu117, URL http://icesjms.oxfordjournals.org/cgi/doi/10.1093/icesjms/fsu117.

List of Figures

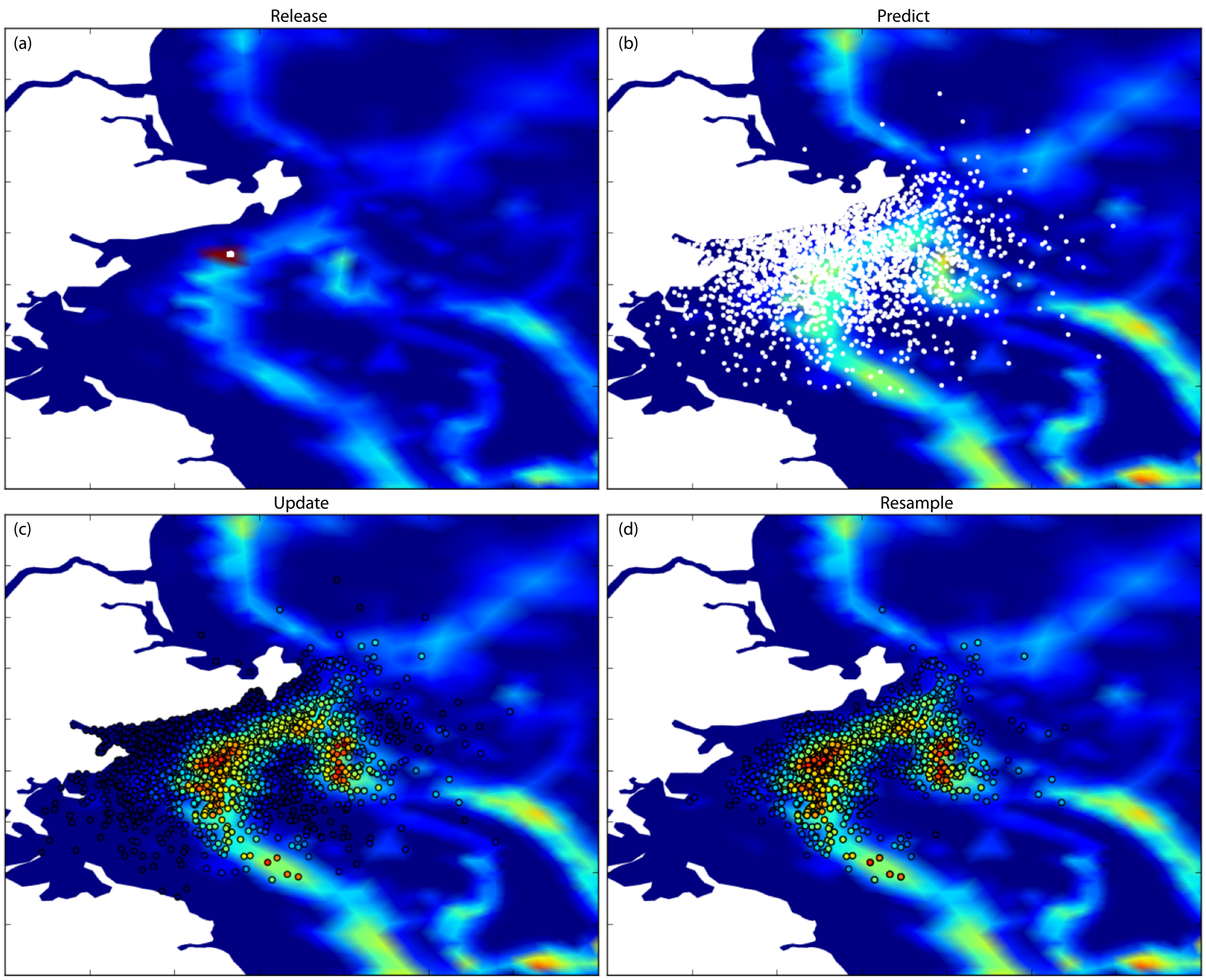


Figure 1: Demonstration of the steps of the particle filter: release, prediction, update, and resample. This is an example of cod in the Gulf of Maine. Color indicates values of the daily likelihood distribution L_{dt}.

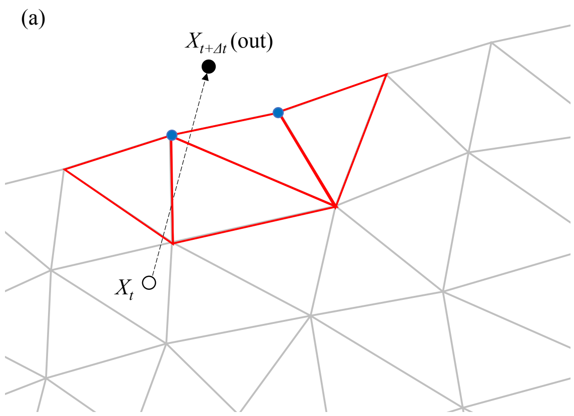


Figure 2: Boundary treatment of the particles during the prediction step. After the tentative movement established by the horizontal random walk (black particles), each particle is then classified as being outside or inside the domain. (a) A particle not found in all of the triangular cells (red triangles) surrounding the two nearest mesh vertices (blue dots) is characterized as being outside of the domain, and is subsequently restored to the location where it resided prior to the step. (b) A particle found in any of the triangular cells (green triangles) surrounding the two nearest mesh vertices (blue circles) is characterized as being inside of the domain and is allowed to remain in the new location.

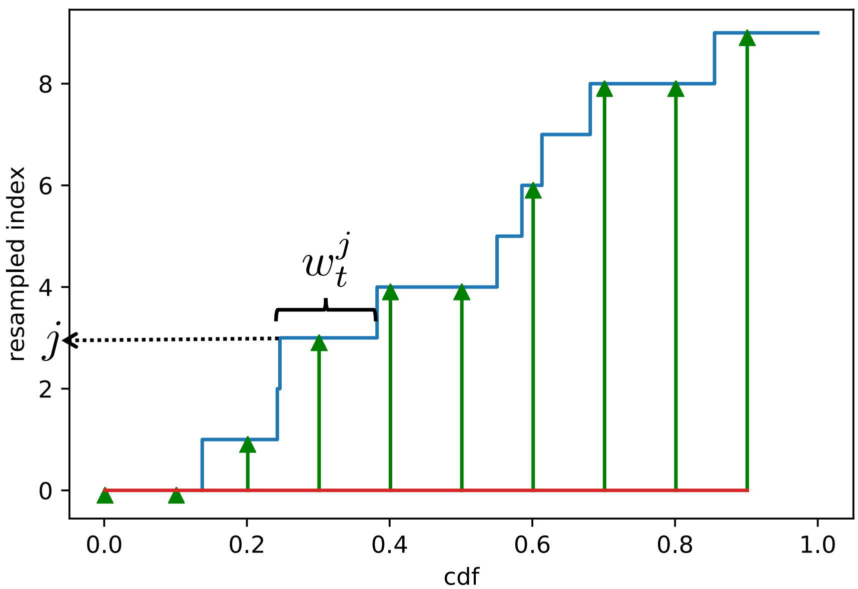


Figure 3: Schematic plot of the resampling process for $N = 10$ particles. The blue line is the cumulative density function (cdf), and the vertical axis is the particle index. Green arrows represent the equal divisions to determine which particles are sampled.

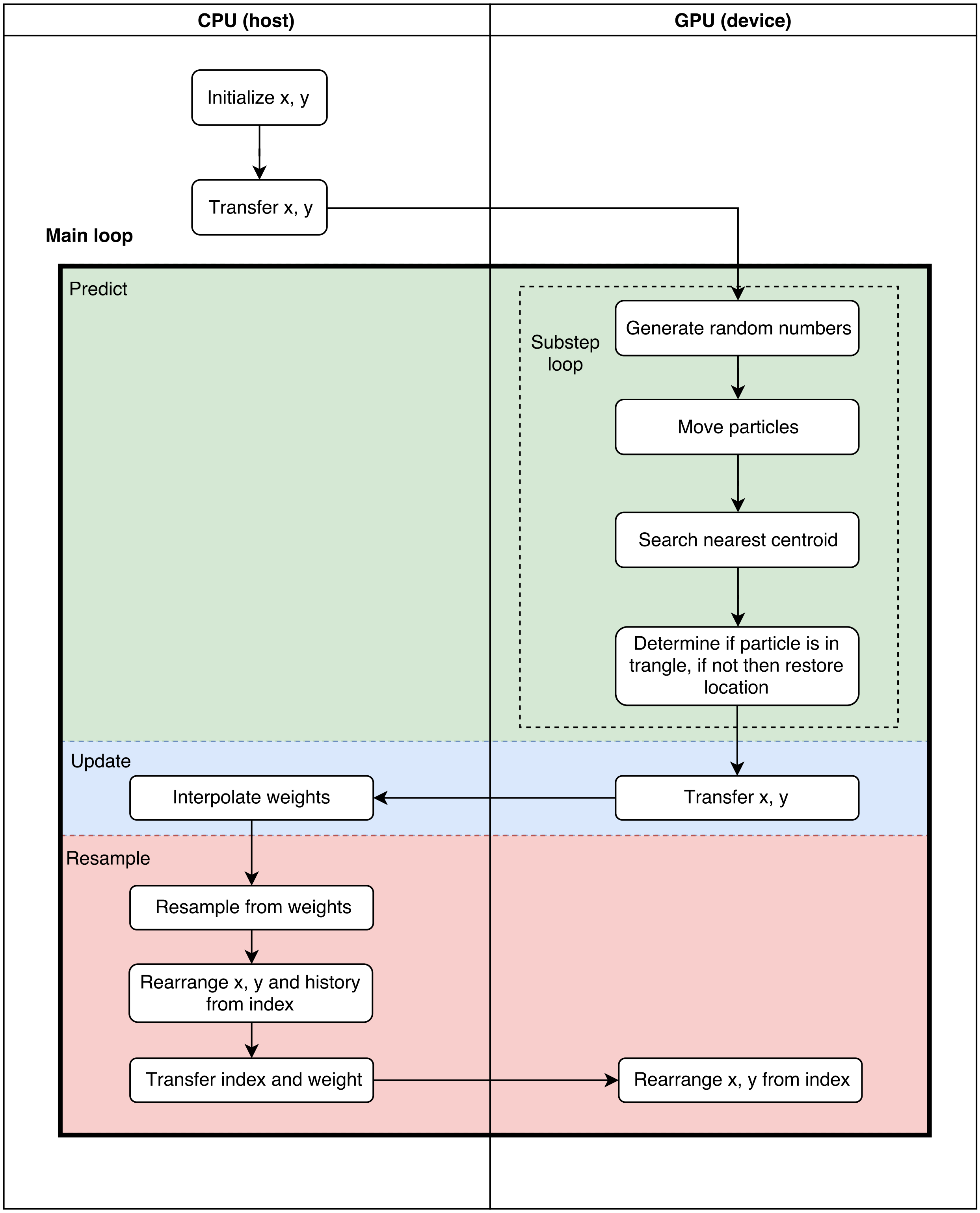


Figure 4: Flow chart of the parallel particle filter geolocation on graphics processing units (GPUs).

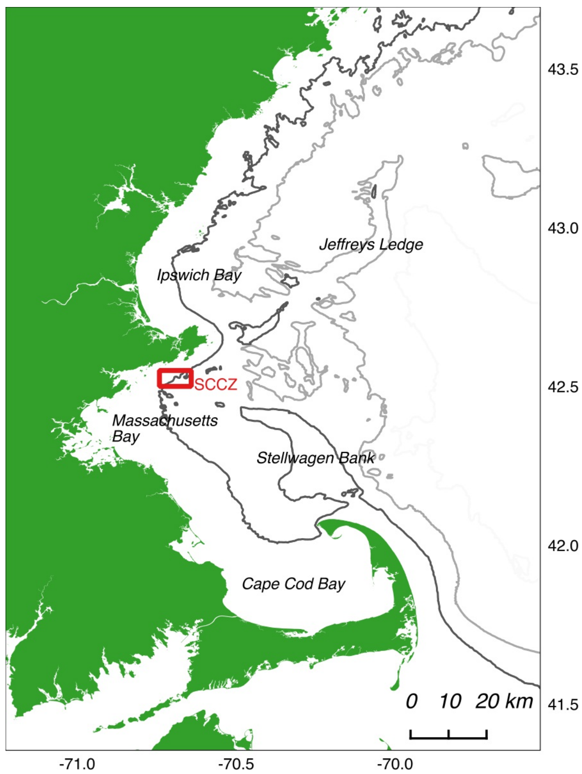
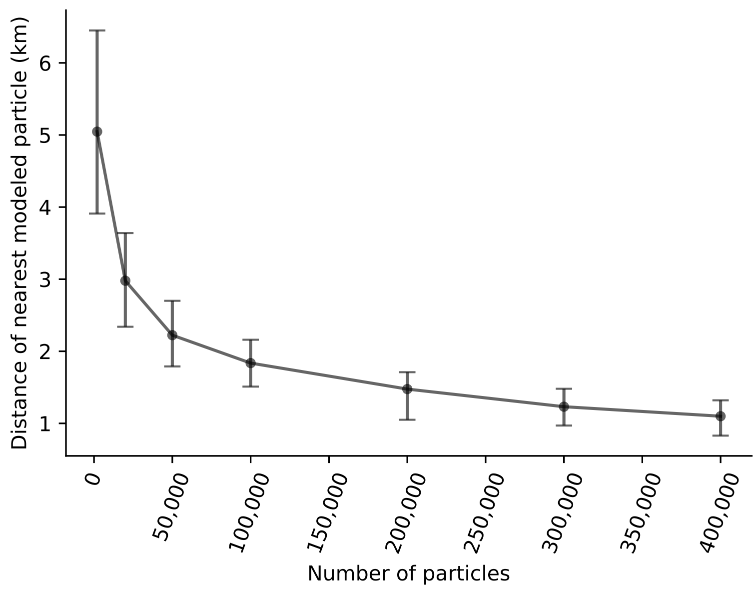
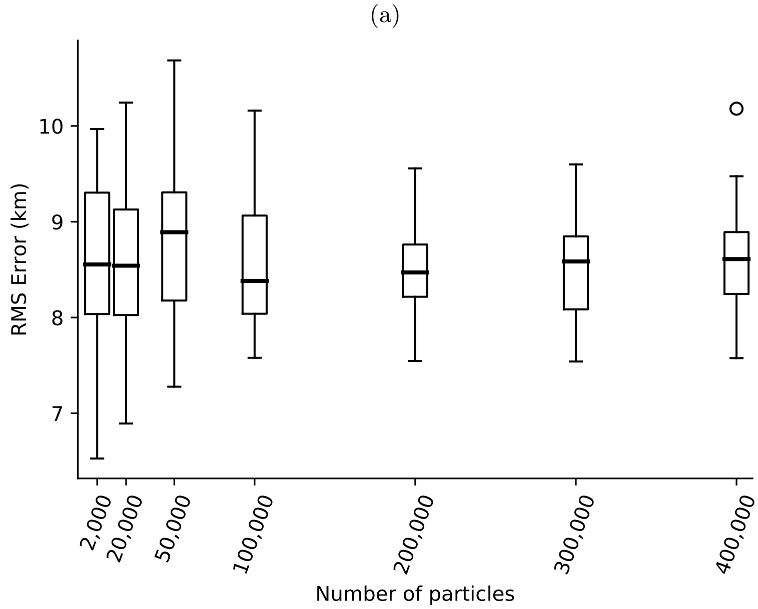


Figure 5: Map of western Gulf of Maine showing the Cape Cod Bay, Stellwagen Bank, Jeffreys Ledge, and the Spring Cod Conservation Zone (SCCZ) as the red rectangle. Selective isobaths of 50 m, 100 m, and 200 m are also shown as lines of decreasing shades of gray with greater depth.

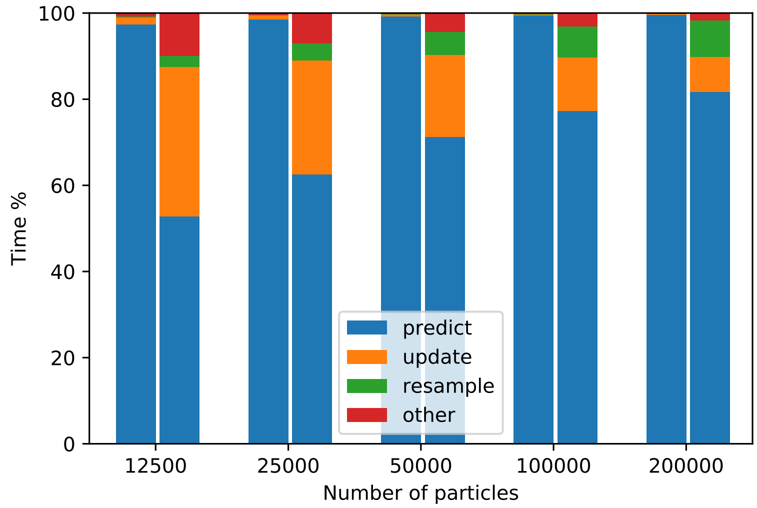


(a)

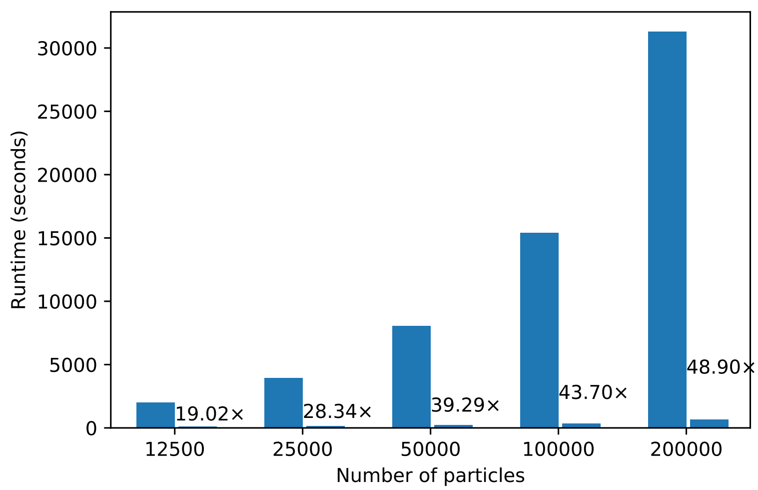


(b)

Figure 6: (a) Error bar plot of the distance between the nearest modeled particle and the associated acoustic location, showing the mean values (solid dots) and range (whiskers). (b) Box plot of RMS Error of the most probable track (MPT) in relation to the number of particles used in a particle filter geolocation run, over 30 model runs for each particle number, showing median values (thick black horizontal line), 25% and 75% percentile values (box outline), outliers (hollow circle), and the highest and lowest value within 1.5 times the interquartile range (whiskers).



(a)



(b)

Figure 7: Comparison of the time percentage for each step (a) for the PF geolocation between serial CPU (left bars) and GPU (right bars) and total run time and speed-up factors (b).

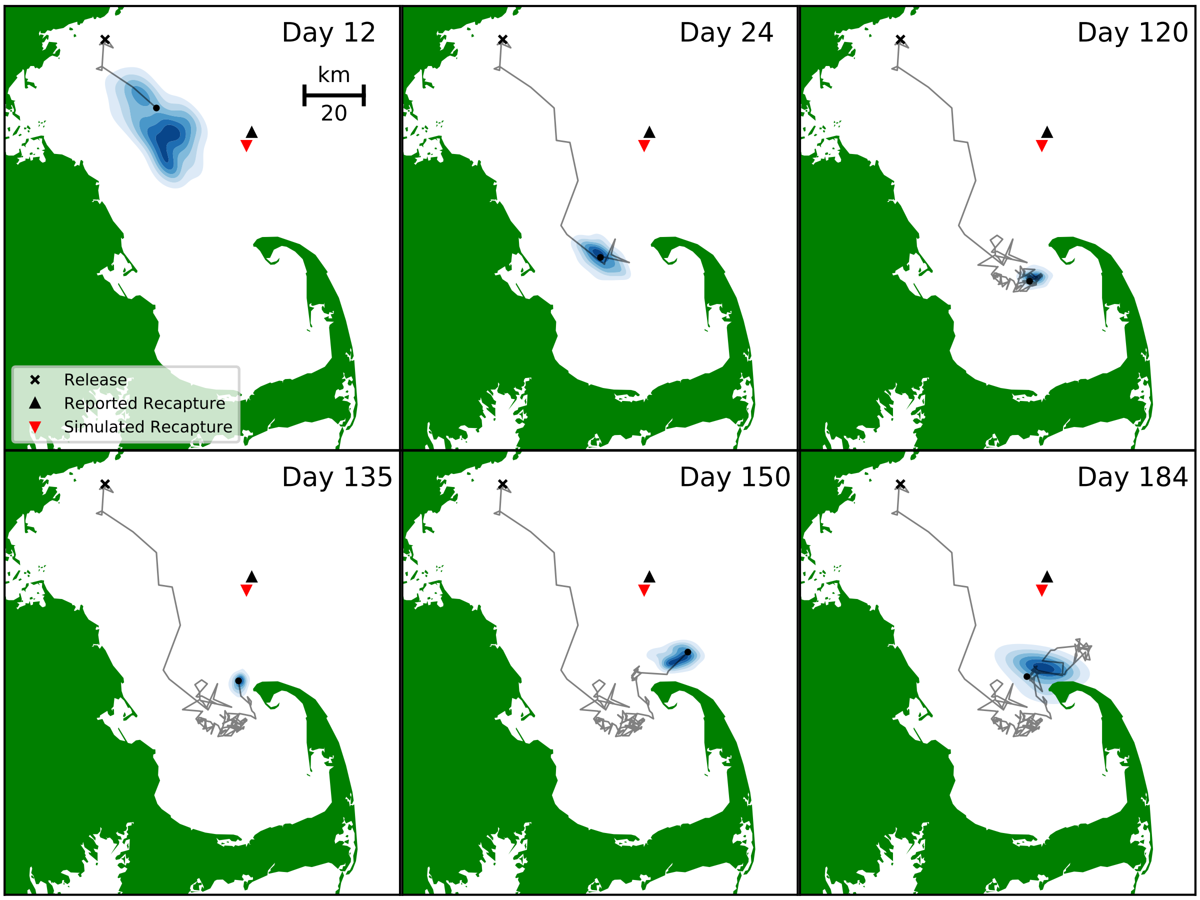


Figure 8: Progression of the daily posterior distribution (color rendering) and the most probable track (MPT, black line) for cod #13. Black cross: release location, black triangle: reported recapture location, red triangle: simulated recapture location.

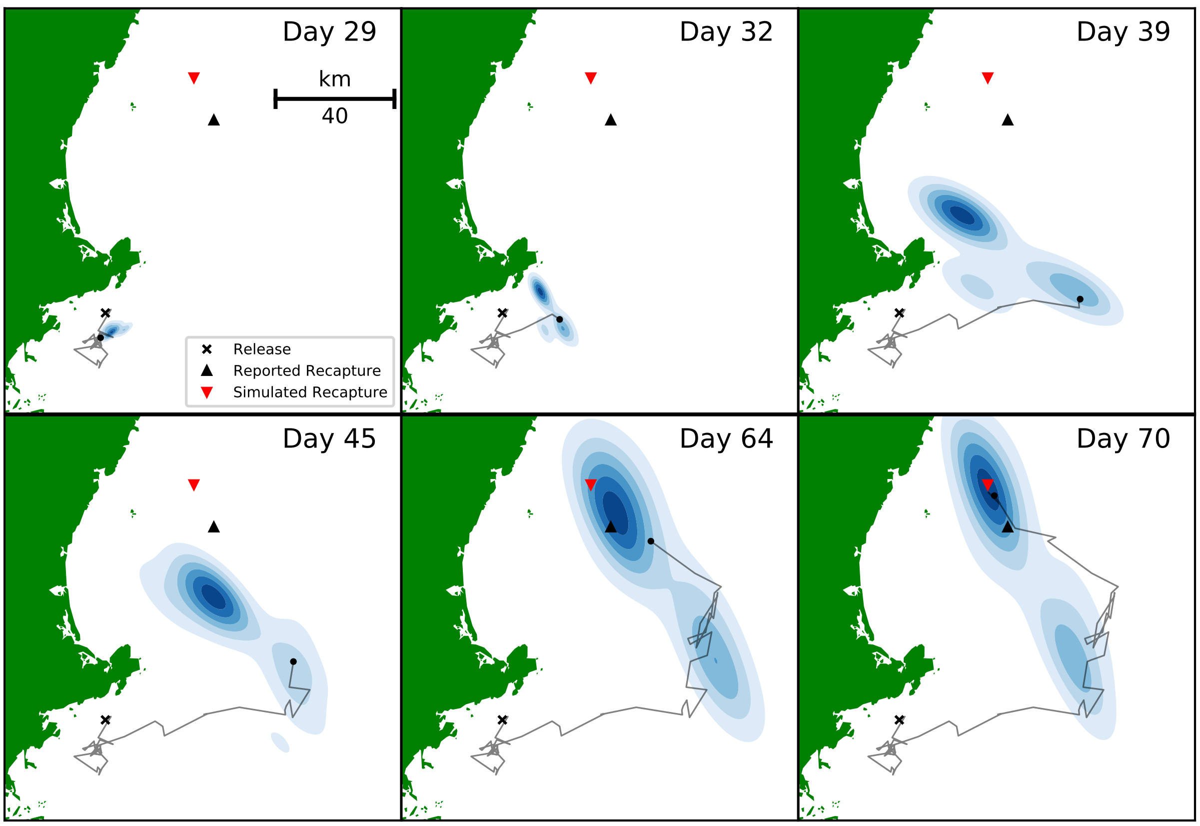
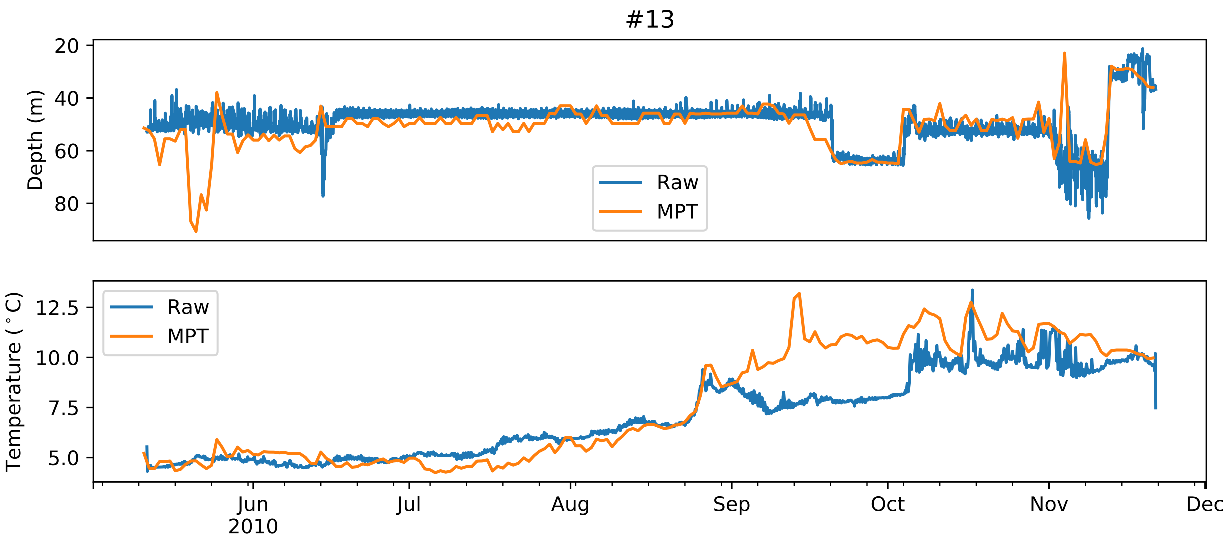


Figure 9: Progression of the daily posterior distribution (color rendering) and the most probable track (MPT, black line) for cod #17. Black cross: release location, black triangle: reported recapture location, red triangle: simulated recapture location.

(a)



(b)

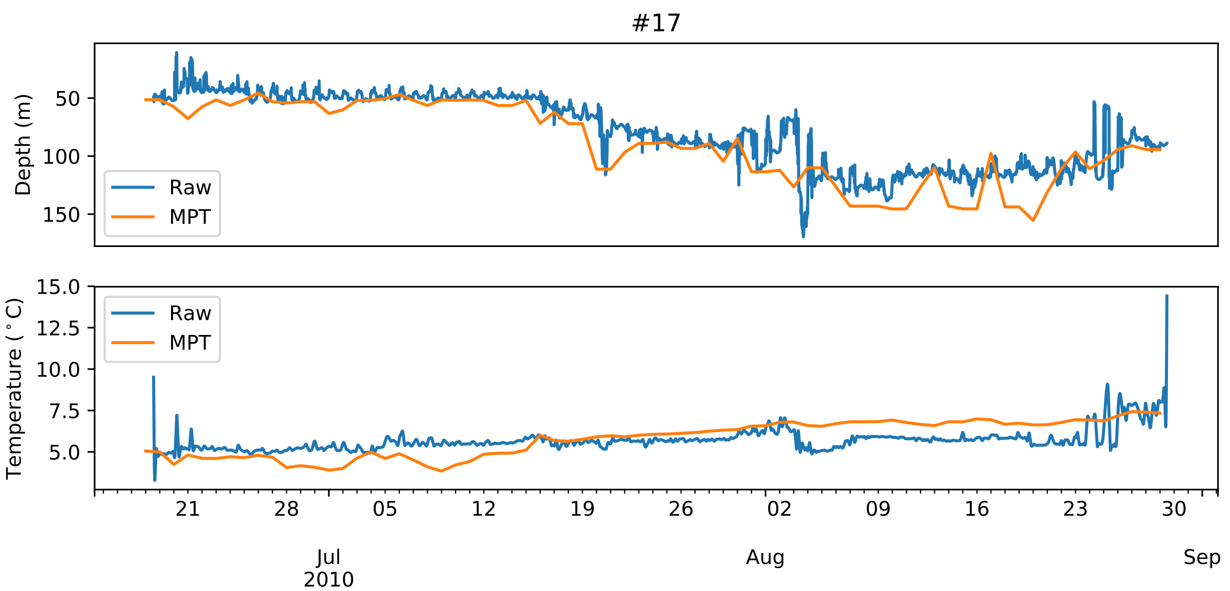


Figure 10: Comparison of the raw depth and temperature time series data recorded by the data storage tags (DSTs; blue line) and the daily depth and temperature data reconstructed from environmental database along the most probable track (MPT; orange line) for (a) cod #13 and (b) #17.

Table 1: Skills of the most probable track (MPT) of the PF geolocation method for mooring and double-tagging

Data Source	Mooring	Double-tagged fish
# tag deployments	14	10
# geolocation days with known locations	762	222
E2 Error range (km)	0.01–27.53	0.29–46.77
E2 RMS (km)	14.95	18.19
E2 Median (km)	9.71	6.0
E2 Mean $\pm$ S.D. (km)	12.03$\pm$8.87	12.47$\pm$13.28
E3 %days within 95% credible area	61.9	100

Table 2: Hardware specifications and PF-Throughput in geolocation-days per wall clock hour (d/h) on ten GPUs.

NVIDIA GPU Model	Architecture Generation	Compute Cores	Base Clock (MHz)	Memory Band-width (GB/s)	Single Precision GFLOPS	PF-Throughput (d/h)
Tesla V100	Volta	5120	1455	900.0	14899.0	1705
Titan X	Pascal	3584	1417	480.0	10157.0	1090
Tesla M60	Maxwell	4096	899	320.0	7365.0	877
Tesla K80	Kepler	4992	560	480.0	5591.0	657
GeForce GTX 1050 Ti	Pascal	768	1290	112.1	1981.4	638
GeForce GTX 1050	Pascal	640	1354	112.0	1733.1	483
Tesla K40c	Kepler	2880	745	288.0	4291.0	413
GeForce GTX 750 Ti	Maxwell	640	1020	86.4	1305.6	391
GeForce GTX 560 Ti	Fermi	384	1645	128.0	1263.4	315
Tesla C2050	Fermi	448	1150	144.0	1030.4	228