Workshop on Visualisation in Environmental Sciences (EnvirVis) (2015)
A. Middel, G. Weber, and K. Rink (Editors)

Probability Maps for the Visualization of Assimilation Ensemble Flow Data

Thomas Höllt[1,2] Markus Hadwiger[2] Omar Knio[1] Ibrahim Hoteit[1]

[1]Physical Sciences and Engineering, King Abdullah University of Science and Technology, KSA
[2]Applied Math and Computer Science, King Abdullah University of Science and Technology, KSA

Abstract

Ocean forecasts nowadays are created by running ensemble simulations in combination with data assimilation techniques. Most of these techniques resample the ensemble members after each assimilation cycle. This means that in a time series, after resampling, every member can follow up on any of the members before resampling. Tracking behavior over time, such as all possible paths of a particle in an ensemble vector field, becomes very difficult, as the number of combinations rises exponentially with the number of assimilation cycles. In general a single possible path is not of interest but only the probabilities that any point in space might be reached by a particle at some point in time. In this work we present an approach using probability-weighted piecewise particle trajectories to allow such a mapping interactively, instead of tracing quadrillions of individual particles. We achieve interactive rates by binning the domain and splitting up the tracing process into the individual assimilation cycles, so that particles that fall into the same bin after a cycle can be treated as a single particle with a larger probability as input for the next time step. As a result we loose the possibility to track individual particles, but can create probability maps for any desired seed at interactive rates.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Forecasts of currents and surface flows in oceans allow the estimation of the transport of injected particles such as pollutants and sediments. However, theses forecasts may may be subject to various sources of uncertainties. The uncertainty space is often sampled, by running so called ensemble simulations, where each member in the ensemble relates to one sample. Sampling is usually rather coarse, with a typical ensemble containing tens to hundreds of members. The sampling can be constant over time, meaning that each member at any time step A corresponds to a single defined member in any other time step B. Tracing a particle in such data basically works as such that the particle is split into n subparticles, one for each member of the ensemble, in the first time step and then tracing theses particles over time using the corresponding members. This usually results in a manageable number of particle traces, computationally as well as visually.

Recently data assimilation [LR99, Pha01, HPB02, Eve06, HHG*13, HLP13] became more and more common in the forecasting community. In these approaches measured data is assimilated into the simulation when it becomes available to minimize the uncertainty and keep the simulation on track. Once the measured data is available the result of the corresponding time step is adjusted and the simulation is restarted with the updated quantities of interest at this time step. When the simulation is restarted, new (random) samples need to be generated from the estimated probability distribution of the state, conditioned on the observations [HPGL15]. Generally, the new samples do not have well defined correspondences to the samples before the assimilation step. As a result, behavior over time can only be described as a set of possible combinations of the different members, before and after each assimilation cycle. For an ensemble with n members and m assimilation cycles the number of all possible combinations is n^m. With typical numbers for n and m, it is infeasible to trace an individual pathline for each combination. The Red Sea dataset presented in this paper contains 50 members and 10 assimilation cycles. Tracing all possible combinations would result in $50^{10} \approx 97$ quadrillion pathlines, even for such a small dataset.

In this work we tackle this problem by tracing particles piecewise for each assimilation cycle and binning the final positions of the particles. A probability is assigned to each bin according to the number of particles it contains. In the next cycle only a single particle, weighted by the corresponding probability, is traced for each bin and member. Though, we lose the capability to trace the paths of individ-

DOI: 10.2312/envirvis.20151090

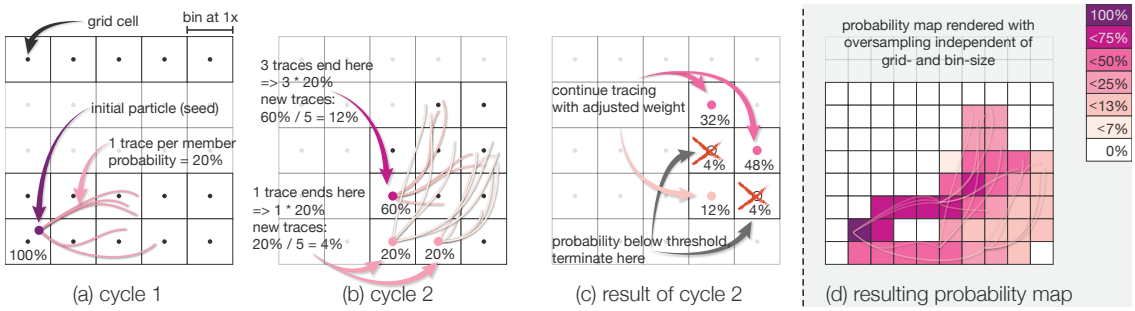


Figure 1: **The Probability Map Computation Pipeline.** (a) to (c) correspond to the first (CPU-based) part, the actual the pathline tracing, (d) to the second (GPU-based) part; rendering the traced lines and their attached probabilities. Note that the color scale is not linear to cover the rapidly dropping probability spectrum.

ual particles in the resulting set of weight adjusted pathlines, this approach allows us to create a map of the probabilities for each position in the grid to be reached by the particle during the time series at interactive rates.

2. Related Work

While the display of uncertainty information, such as error bars, is ubiquitous in visualization of abstract data for a while now, it is a relatively new development to present this information to the user of visualizations in a spatial context. Pang et al. [PWL97] present a general overview of early work on uncertainty visualization for scientific 3D data. It seems that the recent boom of research in this area, however, was sparked by Johnson and Sanderson's call *"to take the next step and make visually representing errors and uncertainties the norm rather than the exception"* [JS03].

Other factors that fueled this development are the efforts by the scientific computing community, especially in the area of numerical simulations, which nowadays usually output multidimensional, multivariate and multivalued data, so called ensemble data ([KDP01, LKP03, KKL*05, LPK05]). Ensemble simulations are especially useful for forecasting in the geophysical sciences. Numerical models are limited by computational resources and parameterization schemes and as such are not exact. Additionally, starting conditions can be uncertain due to scarcity of data and measurement errors. The inherent uncertainty can be mapped to multiple runs of the same or of different models, resulting in multiple possible outcomes forming the ensemble. Some of the first fully featured ensemble visualization applications, such as *Ensemble-Vis* by Potter et al. [PWB*09] and *Noodles* by Sanyal et al. [SZD*10], focus on visualization of weather and climate forecasts or similarly, ocean forecasts (i.e. *OVis* by Höllt et al. [HMC*13, HMZ*14]). These tools use multiple views on the data, some show aggregates of the multivalued part of the ensemble, for example by means of statistics, while other views let the user directly compare different results. Examples for a comparative approaches to ensemble visualization in various application areas are the works of Healey and Snoeyink [HS06], Matković et al. [MGKH09], Piringer et al. [PPBT12] and most recently Demir et al. [DDW14].

Recently multi-valued vector fields and flow analysis came into focus. Petz et al. [PPH12] and Pfaffelmoser et al. [PMW13] present feature based approaches. Their focus is the visualization of the positional uncertainty of important features, such as sources and sinks. Guo et al. [GYHZ13] couple flow line advection and analysis. This approach works well for tens of lines, but for the data presented in this work individual tracing does not seem feasible. Instead, our approach couples piecewise tracing of probability-weighted particles with visualization inspired by dense line visualization for vessel movement tracking as presented by Willems et al. [WvdWvW09, SWvdW*11, SWvdWvW11] and kernel density filtering approaches by Lampe and Hauser [LH11a, LH11b]. Whitaker et al. recently introduced an implementation of boxplots for contours [WMK13] and generalized it for curves [MWK14]. This kind of visualization would fit the presented problem very well, however, requiring extensive preprocessing, it is not applicable in an interactive scenario.

3. Probability Map Computation

Our pipeline for the computation of the probability maps is divided into two parts. In the first part pathline segments are traced on the CPU and stored as line segments, which are then rendered, using the attached probabilities, into a texture in the second part of the pipeline, using the GPU. The resulting texture corresponds to the probability map.

Computing the pathline segments works as follows:

First the particle is injected at the user-defined seed point (the colored dot in Figure 1 (a)). For each member a new *sub-particle* is created with the probability of the corresponding member attached. For the example we use equal probabilities for all sub-particles, defined by the probability of the original particle ($p_{t_0} = 1.0$ for the seed) divided by the number of members (n). Within an assimilation cycle each sub-article is traced with sub-pixel precision, based on the vector field of the corresponding member. For each (except the last) cycle we create a temporary 2D histogram, storing the number of pathlines that ended at each position, multiplied with the corresponding probabilities (the colored dots in Figure 1 (b)). The histogram is indicated by the boxes in Figures 1 (a) to (c), note that for the example we use 1x

resolution for the histogram, so that each point in the original grid falls into the center of the corresponding bin. The histograms differ from the final probability map as they are computed per cycle and only contain the end position of each pathline and not the complete pathlines. The histograms only function as the input for the next cycle. For each position in the histogram of a cycle t, where the total probability p_t is larger than a user defined threshold new sub-particles with the probability $p_{t+1} = p_t/n$ are created for cycle $t+1$. These new sub-particles are then traced using the vector fields corresponding to the member and the cycle $t+1$.

The last segment of a pathline of one cycle does not necessarily end in the center of a bin, where new pathlines are spawned for each cycle. To avoid gaps when connecting the segments of two cycles the final position of each pathline is adjusted to the center of the corresponding bin. In addition to the histogram all pathline segments are stored as a simple list of line segments with their corresponding probabilities, which is used later on to create the final probability map. At the moment we use a standard pathline algorithm without any optimization for computing the pathline segments.

The algorithm exhibits two user adjustable parameters. The first one is the resolution of the histogram, which is created based on the grid of the original dataset. We allow 1x, 2x, 4x and 8x resolutions. At 1x resolution one bin for each point in the original grid is created, for each of the following resolutions the number of bins is doubled in x- and y-direction resulting in four bins per grid point at 2x, 16 at 4x and 64 at 8x. Increasing the resolution has two consequences. First, precision and visual quality of the result is increased, however on the downside less particles fall in the same bins, requiring a higher workload, but also decreasing the average probability per bin. This brings us to the second parameter, the user defined threshold to discard sub-particles with very low probability. To allow early termination of the tracing process, when the probability is very low only particles with a probability higher than a user defined threshold are traced. At higher resolutions this threshold is hit more often than at lower resolutions, resulting in particles being dropped earlier. While this sounds problematic at first, we chose a very low threshold of 0.0001% probability as the baseline, resulting in rather few particles being dropped. A more detailed discussion follows in Section 4.

The second part of the pipeline is the creation of the actual probability map, as illustrated in Figure 1 (d). To create the probability map the line segments resulting from the first step are rasterized using the attached probabilities. The rasterization process is done in a texture backed offscreen GPU render pass. We use a simple fragment shader that renders the curves as solid white lines, using the probability as opacity value. By using additive blending the result is a texture where the alpha channel contains the sum of the probabilities of all lines for each texture element, referring to the total probability that the texture element is intersected by any of the pathline traces. The resolution of the texture is chosen independently of the dataset, but in general we use oversampling, to provide more detail. The texture is then used as an input for the visualization where the opacity value is used to look up the corresponding color in a user-selected colormap and to blend the probability on top of the topographic representation of the data.

4. Results

To illustrate our system we use a dataset of the Red Sea, covering the area between 9° N to 30° N, and 32° E to 77° E on a $0.1° \times 0.1°$ (corresponding to 9.6km to 11km) grid with 50 vertical layers. We use the timeframe of January 2004 with an assimilation cycle of three days, resulting in a dataset of 10 sampled time steps, each consisting of 50 members. For tracing and visualization we only consider the surface flow, which is a two dimensional vector field, consisting of 450×210 samples. All timings were measured using a quad-core Intel core i7 processor clocked at 4.0GHz and 24GB of main memory in combination with an AMD Radeon R9 M295X GPU with 4GB of graphics memory.

Figure 2 shows the resulting probability maps for a seed in the western Gulf of Aden close to the Bab-el-Mandeb, connecting it to the Red Sea. The Gulf of Aden area is particularly difficult to forecast as eddies are frequently generated in this area [YH15]. Meso-scale events such as eddies are hard to predict on their own and the strong variability of this area makes the prediction particularly challenging. As a result different simulation runs may vary significantly. This becomes very clear in the probability maps. By looking at the highest probability areas, which closely resemble

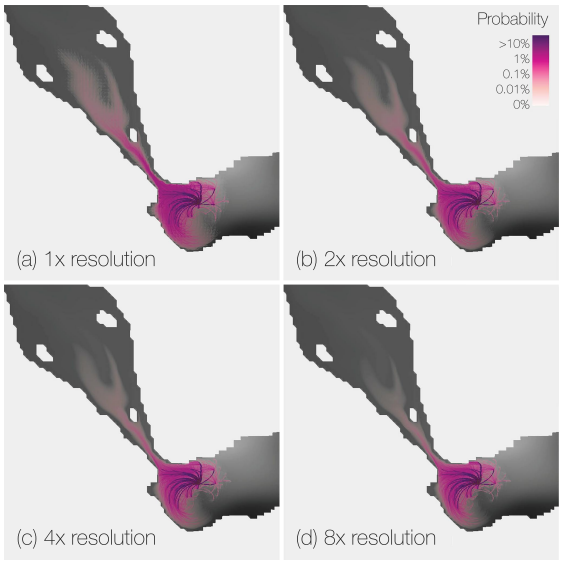


Figure 2: **Visual Comparison of Probability Maps** created using different resolution levels for the particle histogram. (a) shows the probability map created with a histogram at original data resolution, (b), (c) and (d) at 2x, 4x and 8x resolution, respectively.

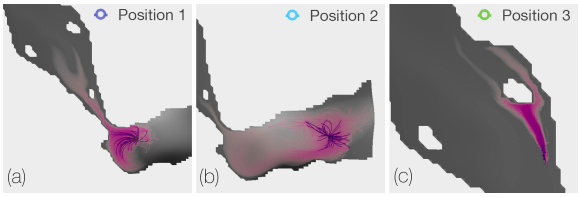
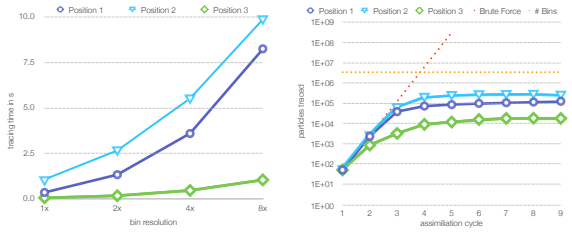


Figure 3: **Exemplary Traces** for three positions at 2x bin resolution. (a) and (b) show two positions seeded in the Gulf of Aden, (a) close to the Bab-el-Mandeb. (c) shows a position south of Farasan Island in the south of the Red Sea. (a) and (b) expose strong variation in the ensemble, while the probability map in (c) seems more uniform.

the seed point and the pathlines for the first assimilation cycle, the variation can clearly be seen, with some pathlines pointing in completely opposite directions. The comparison between different resolution levels of the bins used for computing the pathlines in Figure 2 shows the two main effects that can be observed when increasing the histogram resolution. While the underlying grid is visible, especially at the ends of the traces for the 1x and 2x resolutions ((a) and (b)) it becomes completely invisible at the 4x and 8x resolutions ((c) and (d)). However there is also a negative effect when increasing the resolution. Increasing the resolution means fewer particles will end up in the same bins. Especially in the late assimilation cycles more particles will have a probability lower than the threshold and will thus be terminated early. One way to counter this effect is to use a lower threshold for higher resolutions levels. This will, however, increase the computational cost, without guaranteeing the desired visual effect.

In the following we compare the performance of the pathline computation in a little more detail for three exemplary seed points in the dataset. As described in Section 3 the main performance improvement in our technique, compared to tracing all possible pathlines, comes from binning the particles and tracing multiple particles with equal segments at once. Hence it can be expected that computation times will be strongly dependent on the variation of the ensemble members. Therefore we picked positions with different characteristics for comparison. Figure 3 shows the probability maps at



(a) **Computation Times.** (b) **Traced Pathline Segments.**

Figure 4: **Pipeline Performance** for the positions shown in Figure 3 for all resolution levels (a) and the total number of traced pathline segments per assimilation cycle at 2x (b).

2x resolution for three exemplary positions. Position 1 (Figure 3 (a)) shows medium to strong variation and is rather typical for this dataset. Position 2 (Figure 3 (b)) exhibits very strong variation. Especially for the first cycle individual pathlines can easily be distinguished in the high probability areas (dark purple), corresponding to the first assimilation cycle. In stark contrast to this is the probability map for position 3 (Figure 3 (c)) which shows strong clustering. Individual pathlines are hardly visible, instead they are split up to follow two clusters on either side of Farasan Island.

Tracing performance for the three positions and different resolution levels can be seen in Figure 4 (a). As expected computation time is strongly dependent on the variation of ensemble members. When the variation is large, such as in position 2, a large amount of low probability pathlines needs to be computed, meaning higher computational costs as compared to position 3, where the variation is relatively low, resulting in more pathlines with higher probability and in turn less pathlines traced overall (Figure 4 (b)). In general at 1x resolution the probability maps for all three positions can be computed at interactive rates. At 2x, it took, 1.22, 2.53 and 0.15 seconds respectively to compute all pathlines, including preparation and transfer to the GPU and rendering the probability maps, which still allows the user to explore the dataset without too much waiting time, even for positions with very large variation. At the 4x and 8x resolution levels, especially for position 2, the computation time is hardly interactive with roughly 5 and 10 seconds, respectively. Overall 2x resolution seems to be a good compromise between computational cost and precision. Figure 4 (b) shows the number of weighted particles that are traced for every assimilation cycle. For all three positions one can see that while there is still a sharp rise in the first few assimilation cycles the curves flatten out significantly towards the later cycles. Here the binning approach significantly decreases the number of potential pathline segments. The large performance differences shown above between positions 1 and 2 and position 3 can be directly attributed to the number of individual traced particles, as this number is an order of magnitude larger for positions 1 and 2 than for position 3.

5. Conclusion

We have presented a novel, integrated approach for the computation and visualization of probability maps for ensembles of vector fields. By computing pathlines piecewise for each assimilation cycle and binning and weighting the final position of each particle we are able to estimate the probability that a position can be reached, from a given initial position, at interactive rates for each position in the grid.

In the future we will investigate the effect of increasing the histogram resolution and the interplay with the early termination threshold in more detail. We will also look into more elaborate rendering techniques for the probability map. Smoothing techniques, such as KDEs, should work well, since we are basically estimating a smooth distribution based on discrete samples.

Acknowledgements We would like to thank the anonymous reviewers for their constructive comments. Research reported in this publication was supported by the King Abdullah University of Science and Technology (KAUST).

References

[DDW14] DEMIR I., DICK C., WESTERMANN R.: Multi-charts for comparative 3d ensemble visualization. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014). 2

[Eve06] EVENSEN G.: *Data Assimilation: The Ensemble Kalman Filter*. Springer, 2006. 1

[GYHZ13] GUO H., YUAN X., HUANG J., ZHU X.: Coupled ensemble flow line advection and analysis. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2733–2742. 2

[HHG*13] HOTEIT I., HOAR T., GOPALAKRISHNAN G., ANDERSON J., COLLINS N., CORNUELLE B., KÖHL A., HEIMBACH P.: A MITgcm/DART ensemble analysis and prediction system with application to the gulf of mexico. *Dynamics of Atmospheres and Oceans 63* (2013), 1–23. 1

[HLP13] HOTEIT I., LUO X., PHAM D.-T.: Particle Kalman filtering: A nonlinear bayesian framework for ensemble kalman filters. *Monthly Weather Review 140* (2013), 528–542. 1

[HMC*13] HÖLLT T., MAGDY A., CHEN G., GOPALAKRISHNAN G., HOTEIT I., HANSEN C. D., HADWIGER M.: Visual analysis of uncertainties in ocean forecasts for planning and operation of off-shore structures. In *Proceedings of the IEEE Pacific Visualization Symposium* (2013), pp. 59–66. 2

[HMZ*14] HÖLLT T., MAGDY A., ZHAN P., CHEN G., GOPALAKRISHNAN G., HOTEIT I., HANSEN C. D., HADWIGER M.: Ovis: A framework for visual analysis of ocean forecast ensembles. *IEEE Transactions on Visualization and Computer Graphics 20*, 8 (2014), 1114–1126. 2

[HPB02] HOTEIT I., PHAM D.-T., BLUM J.: A simplified reduced order Kalman filtering and application to altimetric data assimilation in Tropical Pacific. *Journal of Marine Systems 36* (2002), 101–127. 1

[HPGL15] HOTEIT I., PHAM D.-T., GHARAMTI M., LUO X.: Mitigating observation perturbation sampling in the stochastic EnKF. *Monthly Weather Review in press* (2015). 1

[HS06] HEALEY C. G., SNOEYINK J.: VisTRE: A visualization tool to evaluate errors in terrain representation. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission* (2006), pp. 1056–1063. 2

[JS03] JOHNSON C. R., SANDERSON A. R.: A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications 23*, 5 (2003), 6–10. 2

[KDP01] KAO D. L., DUNGAN J. L., PANG A. T.: Visualizing 2d probability distributions from eos satellite image-derived data sets: a case study. In *Proceedings of the IEEE Visualization Conference* (2001), pp. 457–560. 2

[KKL*05] KAO D. L., KRAMER M., LOVE A. L., DUNGAN J., PANG A. T.: Visualizing distributions from multi-return lidar data to understand forest structure. *The Cartographic Journal 42*, 1 (2005), 35–47. 2

[LH11a] LAMPE O. D., HAUSER H.: Curve density estimates. *Computer Graphics Forum 30*, 3 (2011), 633–642. 2

[LH11b] LAMPE O. D., HAUSER H.: Interactive visualization of streaming data with kernel density estimation. In *Proceedings of the IEEE Pacific Visualization Symposium* (2011), pp. 171–178. 2

[LKP03] LUO A., KAO D. L., PANG A. T.: Visualizing spatial distribution data sets. In *Proceedings of the Symposium on Data Visualisation* (2003), pp. 29–38. 2

[LPK05] LOVE A. L., PANG A. T., KAO D. L.: Visualizing spatial multivalue data. *IEEE Computer Graphics and Applications 25*, 3 (2005), 69–79. 2

[LR99] LERMUSIAUX P. F. J., ROBINSON A. R.: Data assimilation via error subspace statistical estimation. Part I: Theory and schemes. *Monthly Weather Review 127* (1999), 1385–1407. 1

[MGKH09] MATKOVIĆ K., GRACANIN D., KLARIN B., HAUSER H.: Interactive visual analysis of complex scientific data as families of data surfaces. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1351–1358. 2

[MWK14] MIRZARGAR M., WHITAKER R., KIRBY M.: Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (Dec 2014). 2

[Pha01] PHAM D.-T.: Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Mon. Wea. Rev. 129* (2001), 1194–1207. 1

[PMW13] PFAFFELMOSER T., MIHAI M., WESTERMANN R.: Visualizing the variability of gradients in uncertain 2d scalar fields. *IEEE Transactions on Visualization and Computer Graphics* (2013). 2

[PPBT12] PIRINGER H., PAJER S., BERGER W., TEICHMANN H.: Comparative visual analysis of 2d function ensembles. *Computer Graphics Forum 31*, 3 (2012), 1195–1204. 2

[PPH12] PETZ C., PÖTHKOW K., HEGE H.-C.: Probabilistic local features in uncertain vector fields with spatial correlation. *Computer Graphics Forum 31*, 3pt2 (2012), 1045–1054. 2

[PWB*09] POTTER K., WILSON A., BREMER P.-T., WILLIAMS D., DOUTRIAUX C., PASCUCCI V., JOHNSON C. R.: EnsembleVis: A framework for the statistical visualization of ensemble data. In *IEEE Workshop on Knowledge Discovery from Climate Data: Prediction, Extremes and Impacts* (2009), pp. 233–240. 2

[PWL97] PANG A. T., WITTENBRINK C. M., LODHA S. K.: Approaches to uncertainty visualization. *The Visual Computer 13* (1997), 370–390. 2

[SWvdW*11] SCHEEPENS R., WILLEMS N., VAN DE WETERING H., ANDRIENKO G., ANDRIENKO N., VAN WIJK J. J.: Composite density maps for multivariate trajectories. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2518–2527. 2

[SWvdWvW11] SCHEEPENS R., WILLEMS N., VAN DE WETERING H., VAN WIJK J. J.: Interactive visualization of multivariate trajectory data with density maps. In *Proceedings of the IEEE Pacific Visualization Symposium* (2011), pp. 147–154. 2

[SZD*10] SANYAL J., ZHANG S., DYER J., MERCER A., AMBURN P., MOORHEAD R. J.: Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1421 – 1430. 2

[WMK13] WHITAKER R., MIRZARGAR M., KIRBY M.: Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2713–2722. 2

[WvdWvW09] WILLEMS N., VAN DE WETERING H., VAN WIJK J. J.: Visualization of vessel movements. *Computer Graphics Forum 28*, 3 (2009). 2

[YH15] YAO F., HOTEIT I.: Thermocline regulated seasonal evolution of surface chlorophyll in the gulf of aden. *PLoS ONE 10*, 3 (2015), e0119951. 3