CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Comparing the Locality Preservation of Z-order Curves and Hilbert Curves

Bachelor of Science Thesis in Software Engineering and Management

Alex Nordin

Adam Telles

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023

A study on leveraging space-filling curves to improve speed of querying large multidimensional datasets

Space-filling curves can be used to encode multiple dimensions of data into a singular integer representation while preserving the characteristics of that data. This study compares the merits of two types of space-filling curves, the Z-order and Hilbert curve, for this use case and the viability of improving software testing of applications dealing with huge multidimensional data using this method.

Supervisor: Christian Berger
Examiner: Robert Feldt

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023

Comparing the Locality Preservation of Z-order Curves and Hilbert Curves

Alex Nordin Adam Telles

***Abstract*—Developing and testing software in the automotive industry and in the research of autonomous vehicles requires the costly querying of multidimensional data recorded from such a vehicle's various sensors. Through encoding such data using space-filling curves, faster queries could be achieved by reducing multiple dimensions into a singular dimension, while exploiting the patterns that emerge in the one-dimensional representation to still get accurate search results. The aim of our study is to systematically compare key behaviors of Hilbert and Morton space-filling curves when applied to realistic automotive sensor data. We applied design science research to develop an experimental environment to investigate the proposed querying method and the comparative results in using either Morton or Hilbert curves with this method. This allowed us to establish some design heuristics for future applications employing this method. We found that asymmetry in data can have a strong deleterious or advantageous effect on event querying, and surprisingly little difference in the True Positive to False Positive ratio of search results between Morton and Hilbert curves. Overall, we prove the viability of this use of both Morton and Hilbert curves for up to eight dimensions of data.**

***Index Terms*—space-filling curve, software engineering, software testing**

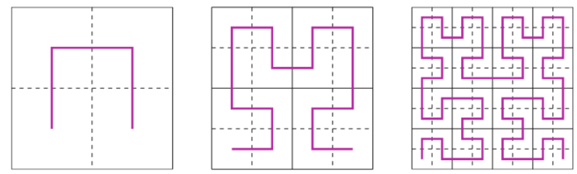

Fig. 1: Example of recursive construction of a Hilbert curve, illustrated by Bader [1].

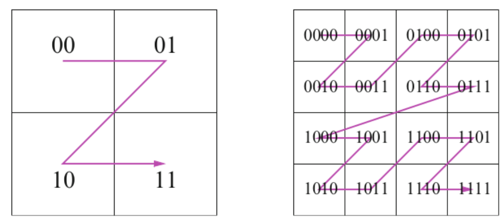

Fig. 2: Example of recursive construction of a Morton curve, illustrated by Bader [1].

I. Introduction

A. Background

A space-filling curve (SFC) is a one-dimensional representation of a multidimensional space. Bader [1] explains the promise of space-filling curves is to preserve the locality of data in the multidimensional space in the linear space. That is, if two points are close to each other in the multidimensional space, they will also be close to each other on the linear space-filling curve.

There are several different kinds of space-filling curves one could build. The Hilbert curve is especially known for its locality preservation, while the Morton curve (also referred to as the Z-order curve) is comparatively simpler in the way it navigates the multidimensional space. Both recursively divide a grid, such as a square or a hypercube, into smaller subsquares, while for each subsquare constructing a space-filling curve, and ultimately connecting all the subsquare curves into a space-filling curve mapping the entire grid. See Fig. 1 for Hilbert encoding in two dimensions and Fig. 2 for Morton encoding.

However, the locality preservation of SFCs is not perfect. In a 2-dimensional grid, a point coordinate can have four "nearest neighbors" i.e., the points spatially nearest to them. Jagadish [2] explains that for any point only two of these nearest neighbors will be preserved in the linear SFC representation. Given then that locality preservation cannot be perfect, it is necessary to mitigate the issues that might arise when making use of SFCs to increase the usability of their software and provide value to their stakeholders.

B. Problem Domain & Motivation

Space-filling curves have seen application in a wide range of scientific fields such as medicine by Kinney et al. [3], Yan and Mostofi's work in robotics [4] and data partitioning to enable efficient cluster computing as exemplified by Alis et al. [5]. Of particular interest to us is the application of space-filling curves for indexing and querying multidimensional data. With the advent of cloud storage, internet of things, the amount and size of data being processed every day globally has kept pace with the increase of computational power available to process that data.

Among these prolific data producers are autonomous cars. Equipped with a set of sensors, from cameras to LiDAR, autonomous cars output complex, multidimensional data potentially hundreds of times per second. Here at the University of Gothenburg, if a researcher wishes to study the data produced by an autonomous car, they would drive the car in some pre-determined way and record the output of the sensors chronologically. This would then yield them a time series of sensor values, and the timestamps at which these values were recorded.

To preserve the temporal relation of the sensor values to each other, a common strategy for storing this time series in a database is as a set of key-value pairs. The timestamp acts as the key, which holds the values from the sensors recorded at that time. This allows for fetching data by when it was recorded. However, this introduces a challenge when querying data for what we will call “areas of interest”, i.e., novel data produced by events such as a sharp turn or a rapid increase in speed. Locating these areas of interest (AoI) entails traversing every timestamp to check whether it contains what one is looking for. For multidimensional data, such a search has a time complexity of O(n) or worse, depending on the number of dimensions.

According to Yan and Mostofi [4] as well as Böhm et al. [6], using an algorithm to construct a space-filling curve of data instead yields great improvements in the speed of querying that data.

C. Research Goal & Research Questions

The aim of our study is to systematically compare key behaviors of Hilbert and Morton curves when applied to realistic automotive sensor data. The use case in question is key-only indexing of multidimensional data using either Morton or Hilbert curves, and exploiting the fact that certain maneuvers emerge as particular patterns or pattern clusters on the SFCs to identify AoIs. For this purpose, it becomes important for the curve to be expressive. By this, we mean that when a curve is constructed, it clearly expresses patterns that can be used to effectively identify AoIs.

There are multiple factors that could complicate this. There is a possibility that when translating multidimensional data into its single-dimensional representation, certain value ranges produce more or less expressive curves. Asymmetry in the data could lead to signals with a higher value compared to other signals, in a sense, overpowering the other signals and leading to a curve that does not sufficiently describe the car’s maneuvers for our use case. Consideration needs to be put into how to represent data to construct expressive curves, from which one can easily glean information.

RQ1: What are the value ranges for Autonomous Vehicle sensor data using uint64 and uint128 for technical implementation, that can be reasonably represented by Morton and Hilbert curves when the number of dimensions to represent increases?

RQ2: How do asymmetric representations of value ranges for multidimensional signals influence the expressiveness?

RQ3: How can software testing benefit from understanding the typical distance spread of areas-of-interest in the single dimension compared to the multidimensional space, and ratio of false positive AoIs for Morton vs. Hilbert curves on growing numbers of dimensions and growing AoIs?

D. Contributions

With the goal of establishing a deeper understanding of the merits of each of these space-filling curves we intend to establish heuristics and provide recommendations for selected application domains such as automotive for how to best model data for use with Morton and Hilbert curves, and the comparative merits of each curve in respect to locality preservation and expressiveness. We ground our investigation in a realistic use case, and realistic data, with the aim to ensure the relevance of our results to real-world applications. Namely, we will be making use of the SnowFox dataset, containing sensor readings from a car used in the study of autonomous vehicles at Gothenburg University and Chalmers University of Technology.

The sensor readings contained in SnowFox describe the car’s heading, speed, the angle of the steering wheel, acceleration, and angular velocity. These values were recorded over a drive around Lindholmen Campus in Gothenburg, Sweden. Constructing Morton or Hilbert curves with varying subsets of this data should allow us to detect events such as the car driving through a roundabout, by these events forming predictable patterns expressed on the SFC.

By manipulating factors described in our research questions, while adding the number of dimensions encoded into the SFCs, we will observe the changes to these patterns. We will study the change in expressiveness, and how AoIs manifest differently, thus exploring how we can best exploit our knowledge of how events manifest in patterns on the SFCs to query data for specific events.

By designing solutions, and engaging in the practice of software engineering itself to investigate a problem, we expect to contribute valuable knowledge to the field. We intend for this work to strengthen the understanding within software engineering on best practices for solving a particular, realistic, and prevalent problem when designing and testing multidimensional data. Increasing the speed of querying data for specific events with the help of SFCs could allow for better test coverage, by reducing the time it takes to run automated tests reliant on, for example, checking the sensor readings recorded at some point in time by an automated car.

How to build software to provide stakeholder value is the essence of software engineering, and by examining how software engineers can best utilize SFCs, we add to the current body of knowledge on how to provide value to our stakeholders.

E. Scope

This research focuses on the automotive domain, in particular the research of automated vehicles from a software engineering perspective. However, we expect that SFCs could be employed in the same way to aid event querying for other time-series data in other domains. Any application of this technique benefits from domain-specific knowledge to decide which signals are most important to preserve in the SFC encoding to achieve the best results.

The code we wrote to explore these heuristics was written in Python and was proven to run on versions 3.10 and 3.11 of CPython. Depending on the implementation of the two SFC algorithms used, different constraints could apply, in addition to language-specific constraints introduced by the coding language one chooses to employ.

The sensor readings in SnowFox are recorded as floating point numbers, necessitating adding an offset to the numbers to transform them into non-negative integer values without losing precision. This is required as the Hilbert and Morton algorithms only accept unsigned integers, which is why we limit ourselves to uint64 and uint128 in RQ1.

F. Structure of the Article

Section I provides a short introduction to the topic of Space-Filling Curves and our research. Further details about SFCs and previous research on SFCs in software engineering, computer science, and mathematics is given in Section II. Section III details the research methodology we employed. Section IV presents the results we obtained for each research question, while the analysis of those results and the discussion in relation to related work is written in Section V. Section VI contains our conclusions and thoughts on possible future work.

II. Background

Our research builds on the work of Berger and Birkemeyer [7], and so it is important to have some knowledge of that work in order to better understand some aspects of this work, notably the concept of the Characteristic Stripe Pattern (CSP). A CSP is the pattern of stripes that can be observed when plotting the distribution of all values of an SFC using a specialized kind of histogram, which we refer to as a CSP plot in this study. Throughout this thesis, when we plot the CSP of a curve, we place the values of the curve on the x-axis and limit the y-axis to go from one to zero. Sometimes we overlay the CSP histogram with a scatter plot mapping a specific event over time on a separate x-axis, to show how specific events fall within the boundaries of the stripes.

Signal	Available Resolutions
Heading	5Hz, 10 Hz, 20 Hz
groundSpeed	5 Hz, 10 Hz, 20 Hz, 50 Hz, 100 Hz
SteeringWheelAngle	5 Hz, 10 Hz, 50 Hz
IMU-Acceleration (Lateral, Longitudinal, Vertical)	5 Hz, 10 Hz, 50 Hz, 100 Hz
AngularVelocities (Yaw, Roll)	5 Hz, 10 Hz, 20 Hz, 50 Hz, 100 Hz

TABLE I: Signals in the SnowFox dataset.

Table. I shows the sensor signals available to us from the SnowFox dataset, a set of data collected from a vehicle used in the research of automated vehicles at the University of Gothenburg during a drive around Lindholmen, Gothenburg. This is the data used in our research, and is referred to extensively in this study, and so we include this table to inform the reader about the data at our disposal.

III. Related Work

The book, *Space-Filling Curves* (2013), by Bader [1] is widely cited in recent studies on SFCs, and provides a thorough overview of the foundational concepts of SFCs, including the differences in how Morton and Hilbert curves traverse the multidimensional space and how they encode and decode values which we will now explain in more detail.

According to Bader [1], given a 2-dimensional square grid, the Hilbert curve is constructed in three steps. The square is divided into four subsquares with side lengths half of those of the original. Then a Hilbert curve is drawn for each subsquare. Finally, the curves for each subsquare are rotated or reflected in such a way that they continuously map the entire space, as shown in Fig. 1.

In contrast, the Morton curve similarly divides the square into subsquares, then fits the line, however the line traverses the subsquares in an identical pattern each time, going from the top-left to the top-right to the bottom left to the bottom right as explained by Bader [1]. Following this “Z” pattern for each subsquare necessitates the curve to sometimes jump from one subsquare to another which is not directly adjacent, which is the case with the Hilbert curve.

Fig. 2 shows how a Morton curve can be constructed on a two-dimensional grid. Observe the jump the curve performs, traversing from the bottom-right of the top-right subsquare to the top-left of the bottom-left subsquare.

Of practical importance to software applications making use of Hilbert or Morton curves is how to encode multidimensional values into a single integer representation. This can be achieved by bit-interleaving as shown by Bader [1], or alternatively bit transposition or rotation as explained by Skilling [8].

As early as the 1990s, Jagadish [2] studied the clustering properties of Hilbert and Morton curves. Clustering refers to how well the curves preserve contiguous clusters of points in the linear representation. This is the same concept we have been referring to as locality preservation, and so we will continue to describe it using that term. Jagadish [2] studied locality preservation explicitly for its importance in applying SFCs in database design, as it affects how well queries perform on the curves, and concluded that in this measure Hilbert curves outperform Morton curves.

Building on this work, Moon et al. [9] expanded on Jagadish’s [2] work, for dimensions above two. Dai and Su [10] then further refines the measurement of locality preservation for two-dimensional Hilbert and Morton curves.

Crucially, these studies establish and refine the theoretical properties of SFCs, mostly in mapping two-dimensional data, but do not explore concerns in the practical application of SFCs.

Lawder and King [11] detail the creation of a fully working software utilizing the Hilbert curve to map multidimensional onto linear indexes. Their software is shown to work with up to sixteen-dimensional data. This use case is very similar to the one we base our study on, though we differ in that we are not implementing the SFC algorithm ourselves but are instead investigating design heuristics for applying SFCs with varying value ranges, dimensions, and so forth.

Another practical application of SFCs for linear indexing of multidimensional data is put forth in Lee et al.'s work [12], where they exploit Z-Order curves to perform a skyline query, the details of which are not relevant to our scope, but it is one example among many that show despite the relatively better performance of the Hilbert curve in terms of clustering/locality preservation established in aforementioned research there is still wide interest in the Morton curve in realistic application scenarios related to our research.

No prior studies, to our knowledge, delve into the practical considerations we propose to explore as described in our Research Questions. Additionally, though we foresee the results of our study can be more broadly applicable, the literature on using SFCs in the realm of automotive vehicles is lacking or nonexistent aside from the work done by Berger and Birkemeyer [7]. Further, existing research does not put a focus on the SFCs ability to eloquently express the physical maneuvers of autonomous vehicles, as described by their sensor signals, in a way that supports more effectively querying that data.

IV. Methodology

A. RQ1

The research methodology we use to answer our first research question is design science research (DSR). We base our specific implementation of DSR mainly on how Peffers et al. [13] and Hevner et al. [14] have explained it, where an artifact is designed and implemented. In our case, the artifact consists of an experimental environment where we through our software implementation can systematically test varying parameters in order to observe how the resulting SFCs are influenced. For this first research question, this means we observe how the resulting SFCs are affected by increasing dimensions. Another core feature of DSR is the ability to perform the study in stages and cycles. The early stages involve designing and implementing the artifact. While later stages allow us to analyze and collect new data that will be used in the next cycle, where we get a chance to improve the artifact, as described by Peffers et al. [13]. Fig. 3 shows the stages of our DSR which we intend to perform, as well as how the evaluation stages will affect the next cycle.

B. RQ2

Much like for RQ1, our second research question is also be worked on using DSR and the previously described experimental environment. In this case, the parameters that are studied are yet again the multidimensional signals. However, in this case, we are interested in observing how asymmetric value ranges affect the curve. This means that the resulting curves are compared when the SnowFox data is left as is, manipulated to make certain values larger or smaller, and normalized, which in this case means they are given the same value ranges while preserving their relative differences. In this study, normalizing values is done according to Equation 1.

$$normalized = value/max \tag{1}$$

C. RQ3

In the case of the SnowFox dataset, manual work is required if precise locations of roundabouts, lane changes, or other maneuvers are desired. This means for example manually reviewing video recordings or GPS data to see if such a maneuver occurred. If video or GPS data is not available due to privacy concerns, yet another manual and heuristic approach may be required to find roundabouts by examining the raw data and identifying patterns correlating to roundabouts. While some of these methods may continue to be the only way to achieve absolute certainty, with our experimental environment we aim to examine how varying parameters and SFCs can help reduce the time and effort required for event detection. We systematically test how for instance increasing dimensions and asymmetric data affect our ability to detect maneuvers and the ratio of true positives to false positives. Additionally, we try to find a method to optimize the speed at which we can detect events by reducing the amount of data that needs to be searched through. By doing this, we identify trends and lay foundations for how to process data and create SFCs which can provide benefits such as higher test coverage on big data and faster event detection, and what trade-offs certain parameters incur.

D. Data collection

The data which we run our artifact on is real-world data collected from an approximately hour-long car drive. The data includes readings such as X, Y, and Z acceleration and steering wheel angle. In the case of this data, our areas of interest are when the car drives through a roundabout.

One of the main features of the artifact is to produce output data such that we can analyze the SFC with respect to our performance criteria. This is our quantitative data, and it is in the form of graphs, models, and any other form which we find suitable for analyzing the output. The timeline for this project consists of two cycles, each four weeks long. The first cycle consists of creating the artifact and generating data relevant to RQ1 and RQ2. The second cycle is used for refinements to the artifact and answering RQ3.

E. Data analysis

As previously mentioned, the artifact is capable of producing data that we can analyze. The artifact is implemented in Python, and thus we make use of Matplotlib and other Python libraries to visualize data. We use descriptive statistics to compare the two curves and see how they are affected by varying input data and parameters. The resulting data and findings are analyzed together with academic colleagues. This

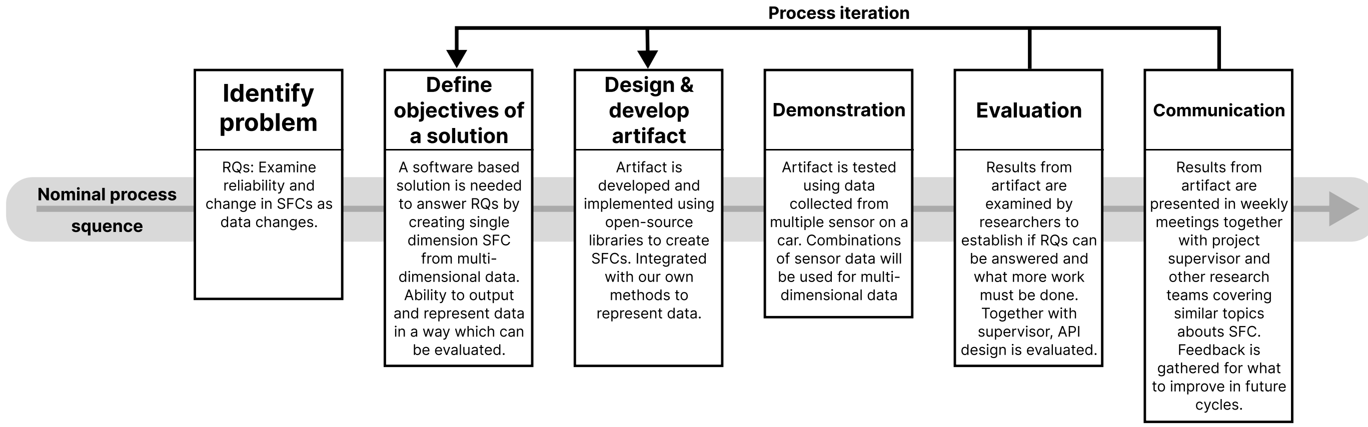

Fig. 3: Design science stages for this study. Adapted from Peffers et al. [13].

is the part of DSR where we get feedback and input on our work in order to improve for the second cycle.

F. *Limitations*

As seen in RQ1, we limit the scope of this research question to only include the data types uint64 and uint128. This is because smaller data types are not practical and larger ones are most likely not performant. We believe that the differences in precision between these two data types should be indicative of how SFCs are affected by more and less precise data, but this nonetheless limits the study.

V. Results

A. *Results for RQ1*

We began by examining the data we had available for this project. The data consisted of two parts, one was from a nearly hour-long drive with data from the following sensor readings: speed, steering wheel angle, heading, angular velocities (yaw and roll), and acceleration (latitude, longitude, and vertical). The number of roundabouts and when they occurred in the run was unknown to us at this point in time. In addition, we had access to a subset of data recorded with the same car, around 40 seconds long, going through a roundabout. We used this as a reference for what the sensor readings could be during a roundabout, to find similar events in the SnowFox data set.

When examining the data from the reference run we found that the steering wheel angle, when plotted, seemed to form a pattern easily inferred to correspond to a roundabout maneuver (as did the yaw rate which is closely correlated to the steering wheel angle). Fig. 4 shows what the steering wheel angle looked like in the reference run.

The next step was to begin the development of the experimental environment we'd need to encode the SnowFox data into SFC curves, plot the results, as well as ancillary helper functionality to expediate fast experimentation. We made use of two open-source implementations for creating the two SFCs in Python, chosen for their ability to encode greater than three dimensions, these were Schubotz's [15] and Galtay's [16] implementations.

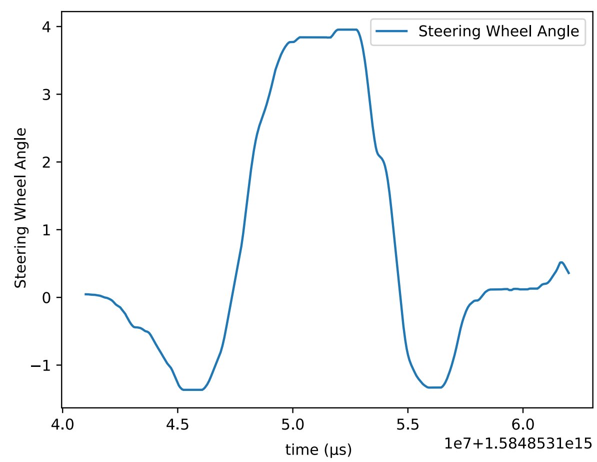

Fig. 4: Steering wheel angle in reference run.

When the functionality to encode into Morton or Hilbert curves was in place, we began a short phase of exploration to create an initial understanding of the behaviors at play. We noticed if we paired with timestamps from the original data, such that we could plot the SFC values over time, depending on what combination of data was used to create an SFC, the resulting code values and the shapes of the SFC could vary greatly. Fig. 5 shows the speed data from the reference run. If we create a Hilbert curve from the steering wheel and speed data, Fig. 6 shows what the resulting curve looks like.

As the data in the SnowFox dataset is recorded at floating point numbers, and the SFC encoding requires non-negative integer values, it should be mentioned that some pre-processing is required. The pre-processing we perform is to add plus 10 to each value, and then multiply each value by a factor of 10^4. This ensures non-negative values, and captures the four digits of precision of the original floating point values. Aside from the two points mentioned, these baseline transformations are somewhat arbitrary, but kept us in line with previous research done on the same dataset by Berger and Birkemeyer [7]. This

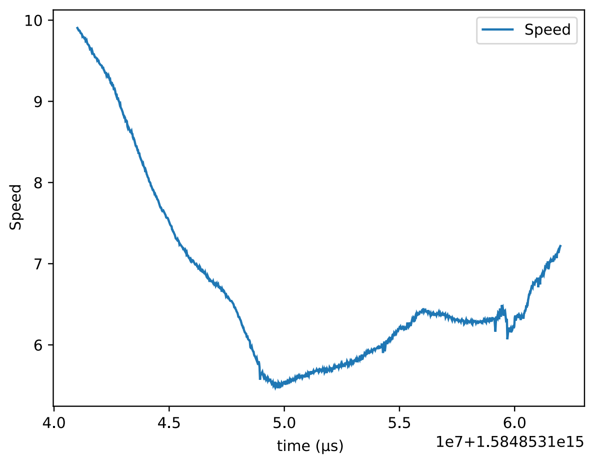

Fig. 5: Speed in reference run.

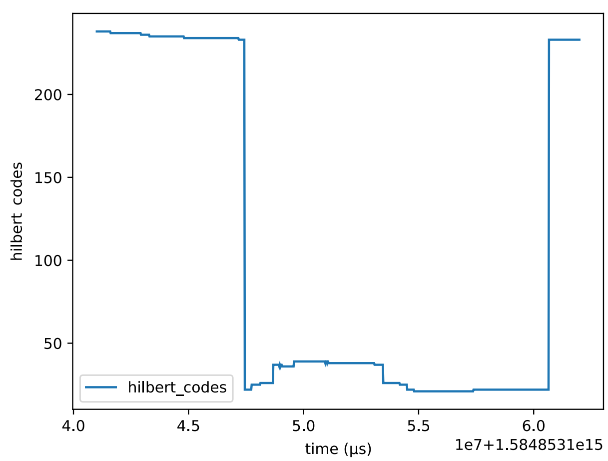

Fig. 6: Hilbert curve based on speed and steering wheel data from the reference run.

pre-processing is done for all consequent tests, and before any other manipulation of the data that is mentioned throughout the rest of the thesis.

With this functionality in place, we could begin to investigate RQ1. The tests we performed consisted of manipulating the absolute value ranges of the data by multiplying each signal by some factor x, adding another sensor signal, and performing the same alteration, making note for each test and its parameters how the CSP of the curve was affected.

We encountered issues with Pythons integer types, more precisely the lack of a 128-bit integer. We solved this problem by extending our experimental setup by writing a separate module in the Julia language, which natively supports the uint-128 type, into which we could import the Morton and Hilbert codes.

These implementation details aside, we encountered two noteworthy behaviors of both Morton and Hilbert curves pertaining to increasing value ranges. With increasing value ranges and dimensions, we noted that the events tended to either spread out, or coalesce into a few points.

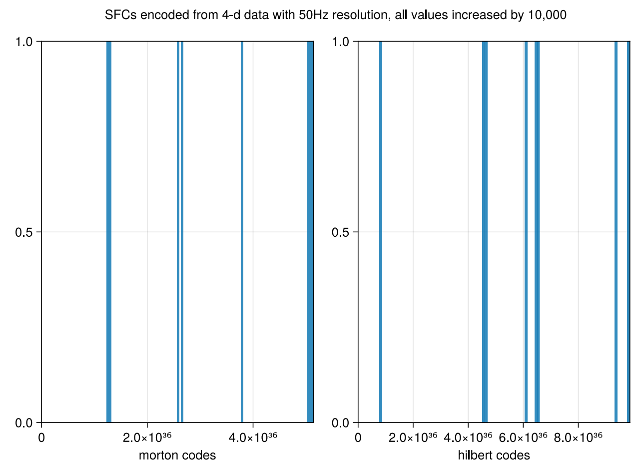

Fig. 7: The CSP formed by encoding 4 dimensions of modified SnowFox sensor readings into Hilbert and Morton curves.

Fig. 7 shows an example of AoIs spreading when we modified four sensor signals by multiplying each value by a factor of 10^4. The "bands" or "stripes" show where all events occur on the Hilbert curve. For four dimensions, we also tried increasing all values by 10^5, however this produced codes which were too large to be represented by uint-128.

Increasing the number of dimensions to seven, we were no longer able to increase the value ranges of the signals and produce either Morton or Hilbert codes that could be fit into uint-128. This means that we were right around the limit of what can be represented at seven dimensions using values in the range of 5664–237905. As seen in Fig. 8, encoding seven dimensions still produces a reasonably expressive CSP.

To better understand the viability of these curves for our intended use case, we extended the functionality in our Julia code to do a simple value-range based search. In simple terms, this search method constructs a histogram of SFC codes and collects the codes which occur around the most weighted edges of that histogram.

We performed a search of the data with this method, recording how many of the codes belonging to verified roundabout events were returned for increasing dimensions and value ranges. For brevity, we chose to present the results for 2, 4, and 7 dimensions only in Fig. II

B. *Results for RQ2*

Starting with two dimensions of sensor data, we performed the following steps for up to seven dimensions. First, encode two or more signals into Morton and Hilbert curves without tampering with the data, not counting the offset we use as a standard when importing the values. Second, produce new curves after increasing one of the signals by a factor of 1000.

Curve	Dims	Modified	Correct Codes Returned	Percentage Returned of All Values
Morton	2	No	18953 out of 29866	44.077
Hilbert	2	No	11969 out of 29866	34.695
Morton	2	Values increased by 10^9	15717 out of 29866	38.890
Hilbert	2	Values increased by 10^9	15304 out of 29866	39.458
Morton	4	No	20768 out of 29866	78.845
Hilbert	4	No	23602 out of 29866	88.296
Morton	4	Values increased by 10^4	9626 out of 29866	10.084
Hilbert	4	Values increased by 10^4	3573 out of 29866	11.507
Morton	7	No	7078 out of 29866	30.539
Hilbert	7	No	19943 out of 29866	63.894

TABLE II: Experiment with range based search, for elevated value ranges

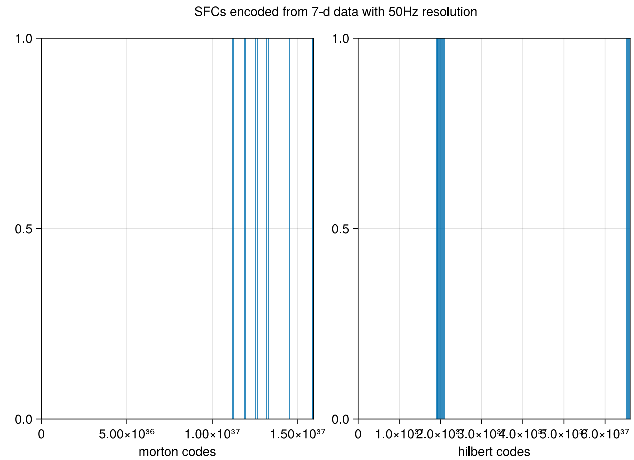

Fig. 8: Seven dimensions can still be encoded, but is at the limit of what we can encode from the SnowFox dataset without loosing precision.

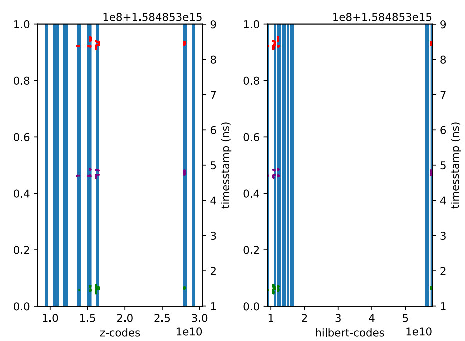

Fig. 9: CSP formed by encoding the steering wheel angle and yaw from the gyroscope

Constructing both Morton and Hilbert curves using two dimensions of the SnowFox data produces the CSP shown in Fig. 9. Steering wheel angle and gyroscope yaw were chosen as they correlate strongly to a roundabout maneuver. This matters to us in this instance, as we chose the roundabout maneuver to demonstrate how the patterns formed by specific maneuvers might change, though there are other maneuvers or events we could have chosen. The red, purple and green areas overlaid over the bands of the CSP show three separate roundabout maneuvers.

We performed the same experiment with three-dimensional data, encoding lateral acceleration along with the two signals chosen for the experiment with two dimensions. The patterns of the roundabouts emerged very similarly as they did with two dimensions. Adding longitudinal acceleration as a fourth dimension, while the roundabouts look similar on the Hilbert curve CSP, on the Morton curve they form an almost inverted pattern from what can be observed in two dimensions. There are overall fewer stripes emerging on the CSP, as seen in Fig. 11, before modifying to create asymmetry.

When introducing asymmetry, we see little difference in the CSP in Fig. 12, unlike what happened in two dimensions, though the value ranges wherein the bands of the CSP form are numerically larger.

We then increased the dimensions to seven, adding ground speed, vertical acceleration, and gyroscope roll. This combination of dimensions creates a pattern on the Morton curve that has a greater spread in areas of interest than the curves presented so far, as seen in Fig. 13. However, introducing asymmetry produces a pattern more similar pattern to previous results, judging by the CSP in Fig. 14.

We further tested the expressiveness of the curves by querying the encoded data using a simple search algorithm based on the CSPs showed in the above plots, verifying how many codes known to fall within a specific event (in this case roundabouts) were correctly returned and how much unrelated data was also returned, and how these results were changed by strategically introducing asymmetry. We chose to present the results for 4 dimensions here, as it proved particularly indicative for the reason that four dimensions are around the mid-point of what we have available and also allows us to introduce a significant amount of asymmetry without going

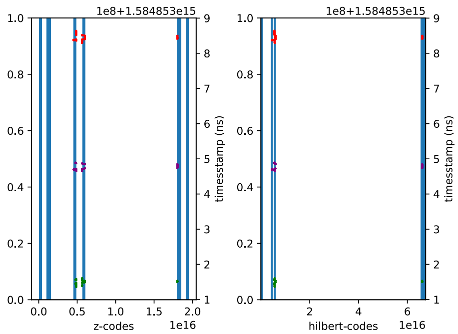

Fig. 10: CSP formed by encoding the steering wheel angle and yaw from the gyroscope after modifying all steering wheel data by factor 1000

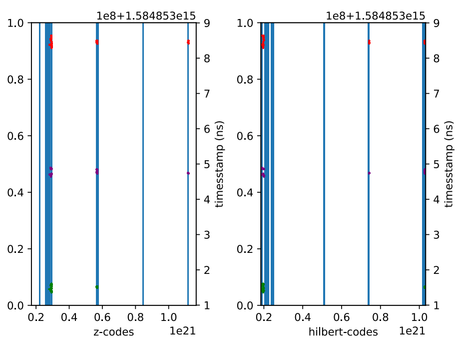

Fig. 11: CSPs on Morton and Hilbert curves constructed from four dimensions of data

outside the bounds of memory bounds of the datatype we used to store the results (uint128).

C. Results for RQ3

For RQ3, we compare how detected results compare to the true location of roundabouts, and how many found roundabouts overlap with a correct roundabout. However, to compare our experiment results to the correct roundabouts, we had to know when the true roundabouts occurred. The true locations of the roundabouts were initially collected simultaneously with all other data. The method for doing so was to, prior to the data collection starting, geo-boxes were manually placed on all roundabouts in the area which the car would pass through. A geo-box in this case means storing latitude and longitude coordinates for all roundabouts such that when using the GPS on the car, if the car is within the coordinates of a roundabout we can save the timestamps for when the car enters and exits the roundabout. This both required tedious and manual work and ultimately proved to not find all the roundabouts present in the dataset. To be certain of the correct timestamps for all roundabouts, more tedious and manual work had to be done. The solution was to have a person sit and watch the video recording from the nearly hour-long drive and note

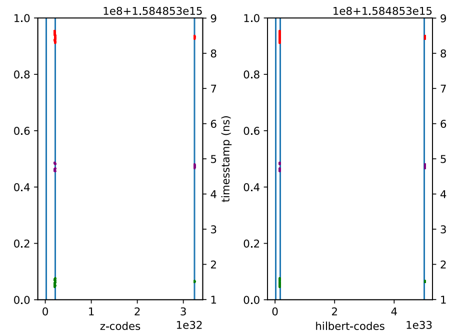

Fig. 12: CSPs on Morton and Hilbert curves constructed from four dimensions of data, with one asymmetrically large dimension

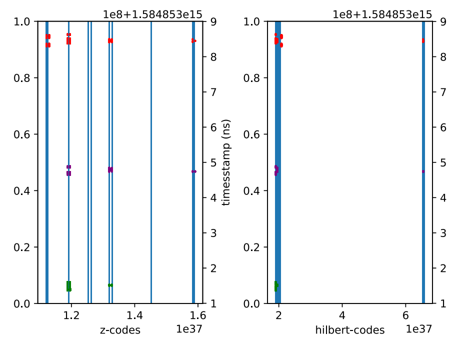

Fig. 13: CSPs on Morton and Hilbert curves constructed from seven dimensions of data

down when the car enters and exits a roundabout. A total of 15 roundabouts exist in the data.

We first attempted to find each roundabout by a search method that was based purely on value ranges, returning SFC codes, the values of which fell within the ranges indicated by the stripes of the CSP histogram. This method proved inadequate at first for finding roundabouts, though it was later refined and used to strengthen some of our findings, as seen in the results for RQ1 and RQ2. The second approach we developed was based on bounding-boxes, exploiting a repeated pattern we observed when plotting the data.

Fig. 15 shows a scatter plot for a Hilbert curve based on steering wheel angle and speed from the reference run (notice that in the scatter plots, the x-axis is SFC-code values and the y-axis is time). When plotting various SFCs with increasing dimensions, we noticed that in many cases we retained the left-right-left pattern for the clusters of data points. The red boxes in Fig. 15 show what were our AoIs, with their size and positioning based on the left-right-left pattern we have observed. The idea for our second approach then is that if a certain amount of data points land in each AoI, then we have found a roundabout.

We knew at this point that asymmetric value ranges have a

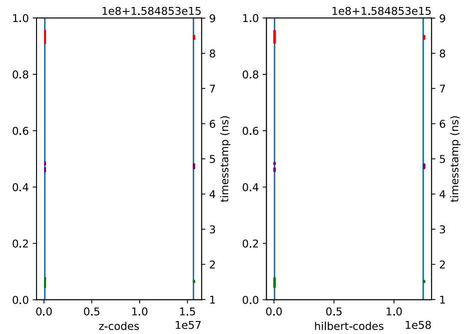

Fig. 14: CSPs on Morton and Hilbert curves constructed from seven dimensions of data, with one asymmetrically large dimension

Curve	Dims	Asymmetric	Correct Codes Returned	Percentage Returned of All Values
Morton	4	No	11044 out of 29866	30.590
Hilbert	4	No	26770 out of 29866	87.804
Morton	4	Steering Wheel Angle elevated by 10^4	9106 out of 29866	18.107
Hilbert	4	Steering Wheel Angle elevated by 10^4	10314 out of 29866	16.723

TABLE III: Experiment with range based search, for asymmetric value ranges

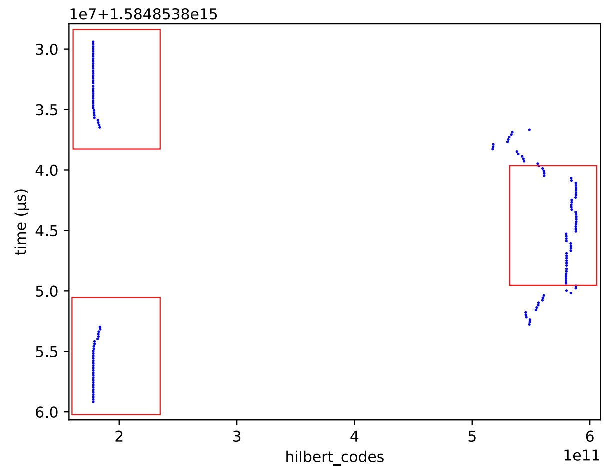

Fig. 15: Scatter plot of a Hilbert curve. Red boxes indicate approximate AoI size and positioning.

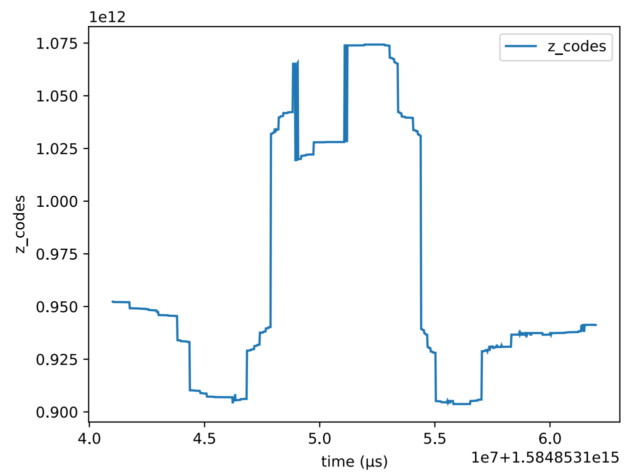

Fig. 16: Normalized Morton curve based on speed and steering wheel data from the reference run.

pronounced effect on the patterns that emerge on the SFC. We therefore found it useful to introduce functionality to normalize the value ranges making all values for a specific type of data a percentage of that data's max value, such that all data was in the range of 0-100 with 100 being the new max value. Fig. 16 shows what a Morton curve would look like when normalizing the value ranges in the steering wheel data and speed data. In contrast, we also created functionality to optionally introduce asymmetry as opposed to removing it, as another parameter in our tests. Fig. 17 shows what happens if we increase the steering wheel data by 10% and decrease speed data by 10%. We can observe that the Hilbert curve when plotted appears more similar to the raw steering wheel data from the reference run.

Once the true locations of all roundabouts were known, we could analyze how well our detection worked by checking how many true positive (TP) and false positive (FP) roundabouts were detected. We also examine the ratio between the TPs and FPs, which in this case is how many FPs are detected for every TP. Our experimental environment allows us to run experiments with varying parameters such as the type of SFC, number of dimensions, asymmetric or symmetric data, and adjust the size of our AoI boxes. The parameters would affect if we first manipulated any data or normalized it before creating an SFC, and what type of SFC would be encoded. The size of the AoI boxes refers to the red boxes in Fig. 15, where increasing the size of them could allow more data points to land in the boxes. This would increase the likelihood of examined data points being classified as a roundabout.

The experiments in Table IV compare how the curves and detection are affected by increasing dimensions. These experiments show that for increasing dimensions, most amounts of TP roundabouts are found when the dimensions are 3 or

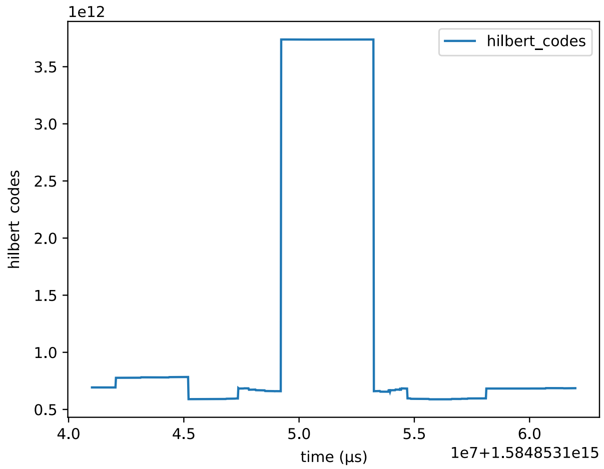

Fig. 17: Hilbert curve when steering wheel angle data is increased by 10%, and speed data decreased by 10%.

4. Additionally, when using a Morton curve we still manage to detect 10 TP roundabouts at 7 dimensions, whereas using Hilbert detects 5. If we exclude experiment 5, which only detected 2 TP and 3 total roundabouts, experiment 3 has the best TP/FP ratio at 2.7.

In Table V the experiments are intended to compare how symmetric and asymmetric data affect the SFCs and detection. Experiment 1 achieves a relatively low TP/FP ratio when normalizing all data prior to SFC creation. Experiments 3 and 4 both increase the values of steering wheel data by 20%. However, experiment 3 normalizes the data first, while experiment 4 does not. This results in experiment 3 not detecting any roundabouts at all, while experiment 4 detects 5 TP and 5 FP, resulting in a TP/FP ratio of 1.

The experiments in Table VI compare how altering the size of the AoI boxes affects detection. Experiments 1-4 show that increasing box sizes results in a higher number of TPs being detected, at the cost of a higher TP/FP ratio. However, this trend does not continue similarly in experiments 5 and 6.

For the set of experiments presented in Table VII, we compare detection when using Hilbert or Morton curves. For the majority of these experiments, we see Hilbert achieving a lower TP/FP ratio, except in experiment 6 where the TP/FP ratio of 0.33 is achieved using a Morton curve. Morton also achieves a higher number of TP roundabouts detected in three out of the four direct comparisons, though this is often at the cost of an increased TP/FP ratio.

The final set of experiments in Table VIII compare roundabout detection using data at different resolutions. We used the data at 50 Hz which contains 169633 entries and compares it to the same data set but down-sampled to 1 Hz, containing 3392 which is only approximately 2% of the 50 Hz variant. At 1 Hz, the full detection run took 21 seconds, while the 50 Hz run took over 107 minutes. However, the difference in the number of detections was only one roundabout, which in this case was a false positive.

VI. Analysis & Discussion

The results of our research indicate that despite the known advantages of Hilbert curves over Morton curves described by Bader [1], Jagadish [2] and others, they do not perform unambiguously better for the use case first proposed by Berger and Birkemeyer [7]. The superior locality preservation is certainly on display, but does not always lead to better search results for either of the approaches developed by us for the purposes of this study.

The results from our testing in regard to RQ3 show a few trends. For a growing number of dimensions, we find that there appears to be an optimal number of dimensions where our detection algorithm performs best, both in terms of the total number of TPs and the TP/FP ratio. For the SnowFox data set, we find that four dimensions (which in this case are the steering wheel angle, yaw rate, roll rate, and speed) perform best for both Hilbert and Morton, providing both the highest number of TPs and the lower TP/FP ratio for both curves. We suspect that the reason Morton performs better than Hilbert at seven and eight dimensions is not due to Morton being superior at expressing these dimensions, instead, we suspect the contrary that Morton is worse. This would be in line with previous findings, where superior locality preservation and expressiveness are found in Hilbert. The issue arises when the final four dimensions are included (x/y/z-velocity and heading), which in this case are not as useful for identifying roundabouts with our setup. Thus, the Hilbert curve's superior ability to represent this data harms the patterns we are searching for, resulting in decreased TP and FP, and increased TP/FP ratio at these high dimensions. This could mean that perhaps the ability to preserve the desired pattern indicative of a roundabout is more important than locality preservation for our use case, as this will ultimately be implementation specific. Additionally, when comparing Hilbert and Morton curves, we tend to see that Hilbert provides a lower TP/FP ratio, while Morton generally provides a higher number of TPs. While this is not always the case as exemplified by a few of our experiments, this could be indicative of which SFC to select depending on if the total number of TPs or TP/FP ratio is more important.

Morton curves perform better when it comes to the size of the integer representation "code" produced from each multidimensional space encoded, producing consistently smaller values. Therefore, if storing more dimensions in larger value ranges into smaller integers is a concern for an application, Morton curves have a slight edge to Hilbert curves in this regard. Keep in mind, we have confirmed that this holds only for the Morton and Hilbert algorithms we used in our research, by Schubotz [15] and Galtay [16] respectively. Galtay's implementation [16] is based on one of the implementations described by Skilling [8], while Schubotz's Morton encoding works by the same bijective bit-interleaving as all Morton algorithms, described by Bader [1] among others.

Experiment	SFC	Dimensions	Asym. Manip.	Normalize data	AoI multiplier	TP	FP	TP/FP ratio
1	Hilbert	1	None	Yes	1.0x	1	0	1 : 0
2	Hilbert	3	None	Yes	1.0x	10	30	1 : 3
3	Hilbert	4	None	Yes	1.0x	10	27	1 : 2.7
4	Hilbert	7	None	Yes	1.0x	5	23	1 : 4.6
5	Hilbert	8	None	Yes	1.0x	2	1	1 : 0.5
6	Morton	1	None	Yes	1.0x	1	0	1 : 0
7	Morton	3	None	Yes	1.0x	9	34	1 : 3.77
8	Morton	4	None	Yes	1.0x	11	36	1 : 3.27
9	Morton	7	None	Yes	1.0x	10	32	1 : 3.2
10	Morton	8	None	Yes	1.0x	7	23	1 : 3.28

TABLE IV: Experiments observing the results from increasing dimensions for Hilbert and Morton curves.

Experiment	SFC	Dimensions	Asym. Manip.	Normalize data	AoI multiplier	TP	FP	TP/FP ratio
1	Hilbert	4	None	Yes	1.0x	6	14	1 : 2.33
2	Hilbert	4	None	No	1.0x	7	27	1 : 3.85
3	Hilbert	4	Steering wheel 1.2x	Yes	1.0x	0	0	-
4	Hilbert	4	Steering wheel 1.2x	No	1.0x	5	5	1 : 1

TABLE V: Experiments observing the results from symmetric and asymmetric data for Hilbert curves.

Experiment	SFC	Dimensions	Asym. Manip.	Normalize data	AoI multiplier	TP	FP	TP/FP ratio
1	Hilbert	4	None	Yes	1.0x	6	14	1 : 2.33
2	Hilbert	4	None	Yes	1.5x	7	18	1 : 2.57
3	Hilbert	4	None	Yes	2.0x	8	21	1 : 2.62
4	Hilbert	4	None	Yes	2.5x	10	30	1 : 3
5	Hilbert	4	None	Yes	3.0x	9	35	1 : 3.88
6	Hilbert	4	None	Yes	3.5x	10	33	1 : 3.3

TABLE VI: Experiments observing the results from adjusting AoI box sizes for Hilbert curves.

We found that asymmetric value ranges have a very noticeable effect on the distribution of values on the SFCs, and therefore the "expressiveness" as shown by the CSPs. We judge that this could be an advantage or a hindrance to using SFCs for our use case, depending on the data in question. If pre-processing the data is feasible, one could manipulate certain signals one judges to strongly correlate to a specific event one wishes to search for, thus emphasizing it and constructing an SFC that is very expressive of that event. However, if pre-processing is not an option because of the computational effort and time involved, there is a chance for the opposite to occur. Overall, we would emphasize that a good understanding of the data one is encoding, and which signals are expressive of the events one would like to search for, is paramount in maximizing the results achieved by making use of SFCs discussed in this study.

Experiment	SFC	Dimensions	Asym. Manip.	Normalize data	AoI multiplier	TP	FP	TP/FP ratio
1	Hilbert	2	None	Yes	1.0x	4	4	1 : 1
2	Morton	2	None	Yes	1.0x	4	5	1 : 1.25
3	Hilbert	4	None	Yes	1.0x	6	14	1 : 2.33
4	Morton	4	None	Yes	1.0x	11	37	1 : 3.36
5	Hilbert	4	Steering wheel 1.2x	No	1.0x	5	5	1 : 1
6	Morton	4	Steering wheel 1.2x	No	1.0x	6	2	1 : 0.33
7	Hilbert	4	None	Yes	1.5x	7	18	1 : 2.57
8	Morton	4	None	Yes	1.5x	11	42	1 : 3.81

TABLE VII: Experiments comparing Hilbert and Morton curves.

Experiment	SFC	Dimensions	Asym. Manip.	Normalize data	AoI multiplier	Time to completion	TP	FP	TP/FP ratio
1	Hilbert	4 dims at 50 Hz	None	Yes	1.0x	107m 39s	6	14	1 : 2.33
2	Hilbert	4 dims at 1 Hz	None	Yes	1.0x	21s	6	13	1 : 2.16

TABLE VIII: Experiments comparing data resolution, measuring time from just before SFC creation until results are produced.

When introducing asymmetry, found that most of the experiments which detected more than half of existing roundabouts did not require any additional data manipulation besides at most normalizing the value ranges. Intentionally increasing the values of data that are deemed indicative of a roundabout did however improve TP/FP ratio, which can be seen in experiment 4 in Table IV and experiments 5 and 6 in Table VII. Additionally, the takeaway from experiments where AoI box sizes were increased could be that to a certain extent, increasing the sizes will provide more TPs at the cost of a worsening TP/FP ratio. Finally, our experiments comparing data resolutions have shown one method of increasing the speed at which events can be detected for our use case. That is, by reducing the size of the data set by nearly 98% we achieved results where the number of detections dropped by one, but only requiring approximately 0.31% of the run time to achieve results.

Answering RQ1: What are the value ranges for Autonomous Vehicle sensor data using uint64 and uint128 for technical implementation, that can be reasonably represented by Morton and Hilbert curves when the number of dimensions to represent increases? For constructing curves viable for the type of querying suggested in this value ranges as high as in the order of 10^{12} produced viable curves. This suggests representing sensor signals recorded with as much as thirteen digits of precision is entirely reasonable. However, if one wishes to store the resulting integer representation in a data type with a fixed number of bits such as uint128, the possible value ranges quickly narrow as dimensions increase. Of course, the use of arbitrarily sized integers renders this a nonissue. This restraint existed for us only to appropriately scope our work and might not exist for future research or industry application unless perhaps as a limitation of the chosen programming language.

Answering RQ2: How do asymmetric representations of value ranges for multidimensional signals influence the expressiveness? Having asymmetrically large or small dimensions heavily influences the patterns in which interesting data is expressed on the curve, in a way that may worsen or improve the expressiveness of the curve depending on the correlation that each dimension has to the class of events that is most important to preserve. Signals represented by greater value ranges in the multidimensional space exert a greater influence on the patterns that form in the singular space. This seems to hold for both Morton and Hilbert curves in equal measure.

Answering RQ3: How can software testing benefit from understanding the typical distance spread of areas-of-interest in the single dimension compared to the multidimensional space, and ratio of false positive AoIs for Morton vs. Hilbert curves on growing numbers of dimensions and growing AoIs? Utilizing SFCs to speed up queries of multidimensional data with up to eight dimensions seems to be viable, but not perfect. Exploiting the distribution of AoIs on both Morton and Hilbert curves, we were able to produce two different algorithms for finding "events" on the curves. Both these algorithms beat a purely heuristic approach to finding events by inspecting the multidimensional data, in both speed and the amount of TP events found. It is faster but not more accurate than reviewing footage and marking events manually, and not as accurate as using GPS data to find events, but with the benefit of higher anonymity specifically by not utilizing GPS location. The speed of these queries, and its stated advantages over the two other options mentioned, could potentially allow for greater test coverage of applications dealing with massive multidimensional data, if the TP/FP ration is improved over our implementations.

A. Threats to Validity

1) Internal validity threats: An internal validity threat that was taken into consideration is related to the implementation of the algorithm that generates SFCs. While we were confident in our abilities to create data output and visualization, the practical creation of SFCs from multidimensional data is an area where feel we lack knowledge. If we were to implement the SFC algorithms ourselves, we worry that there would be a high chance of them being suboptimal or incorrect, which could greatly impact results. To minimize the threat of errors in the implementation of SFC algorithms, we chose to not implement them ourselves. Instead, we made use of open-source and peer-reviewed implementations of SFC-generating algorithms, in large part those implemented by Schubotz [15] and Galtay [16]. This has also been an area where the feedback and iterative nature of DSR helped us discover any mistakes in our work.

2) External validity threats: As just mentioned, we have used Galtay's [16] implementation of Hilbert curves and Schubotz's [15] implementation of Morton curve generation in Python. While this did reduce the chances of us making mistakes in the implementation of curve creation, this could mean that projects using other implementations or even custom implementations could find different results than those we have.

Another external validity threat is related to the data types used in this study. While our input data was integer values when an SFC was created, we encountered compatibility issues regarding Python and Python libraries which were used to visualize data. As data values grew increasingly large, most of the libraries we used to visualize the data were unable to work with such large values. A workaround was to cast SFC codes (those referred to as 'hilbert codes' and 'z codes' in this study) to floating point data types. To ensure that this did not affect the results, we calculated the delta between two SFCs, where one used integers and the other floats. We found no difference between the values other than the data types. Additionally, we implemented our visualization in the Julia programming language as well, such that we could visualize and analyze the data without changing data types. While we believe our workaround is valid for this study, we raise the concern and threat this may pose to similar implementations of this work in other programming languages or differing technologies, which may not treat data types similarly to our case.

3) Construct validity threats: A construct validity threat related to this work is our ability to detect roundabout maneuvers when the first exit of a roundabout is taken. This specific type of maneuver can be difficult to detect since it is similar to that of a normal right turn taken by a car. Also, a roundabout maneuver where the first exit is taken lasts for a shorter duration and does not have an equally unique pattern to that where the second, third or fourth exit is taken. We believe that despite this, only being capable of detecting second, third, and fourth exit roundabout maneuvers will be sufficient to answer our research questions. This is because we can still observe emerging trends and changing patterns as our input data and parameters vary. Nonetheless, this is still a threat to validity that should be considered.

VII. Conclusion and Future Work

Space filling curves have been shown to have many novel applications, and this study investigates a very recently pioneered solution to the problem of the time cost of querying large multidimensional data in the automotive realm. We investigated the relative merits of two types of space filling curve, Morton and Hilbert, through the development of an experimental environment in which we tested the proposed use case on real automotive data collected for the purposes of autonomous vehicle research. We thus established heuristics for the use of space filling curves for this new use case, such as; the pronounced impact of asymmetry, the surprising similarity in querying performance of Morton and Hilbert curves in regard to True Positive/True Negative rates, and how dramatically one can downscale data resolution and still achieve good results.

Our findings suggest that despite the mathematically proven superior locality preservation of Hilbert curves, one could take advantage of the relatively quicker encoding algorithm of Morton curves and achieve similar results for our proposed use case. In addition, we showed that with the only discovered limitation being the memory capacity of the datatype used to store the output, increased value ranges is not a hindrance to how well the resulting curves can be queried, meaning one could safely store more precision without negative effects. However, asymmetry in data must be considered to have a powerful effect on the expressiveness of space filling curves, and we suggest data to be either normalized or pre-processed in order to achieve the best results in the problem domain in which this solution is deployed.

It is our strong belief that the application of space filling curves explored in this study has significant potential to improve querying, and therefore testing of applications dealing with large multidimensional data, in the automotive industry and beyond. However, the solution developed by us in this study is still immature, and so we urge future research in this field to focus on developing a solution for improving the True Positive/False Positive ratio which we achieved in this study.

Acknowledgments

We thank our supervisor Christian Berger and his colleague Lukas Birkemeyer for their support and the work that laid the foundation for this study. Additionally, we would like to thank the other bachelor students who were part of the larger group of students researching this broader research topic in conjunction with us for their collaboration, and camaraderie.

References

[1] M. Bader, *Space-Filling Curves*. Springer Berlin, Heidelberg, 2013. DOI: 10.1007/978-3-642-31046-1.

[2] H. V. Jagadish, "Linear clustering of objects with multiple attributes," in *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*, 1990, pp. 332–342.

[3] N. Kinney, M. Hickman, R. Anandakrishnan, and H. R. Garner, "Crossing complexity of space-filling curves reveals entanglement of s-phase dna," *Plos one*, vol. 15, no. 8, e0238322, 2020.

[4] Y. Yan and Y. Mostofi, "Efficient clustering and path planning strategies for robotic data collection using space-filling curves," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 4, pp. 838–849, 2016.

[5] C. Alis, J. Boehm, and K. Liu, "Parallel processing of big point clouds using z-order-based partitioning," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives*, International Society of Photogrammetry and Remote Sensing (ISPRS), vol. 41, 2016, pp. 71–77.

[6] C. Böhm, S. Berchtold, and D. A. Keim, "Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases," *ACM Computing Surveys (CSUR)*, vol. 33, no. 3, pp. 322–373, 2001.

[7] C. Berger and L. Birkemeyer, "ZEBRA: Z-order Curve-based Event Retrieval Approach to Efficiently Explore Automotive Data," in *Proceedings of the 35th IEEE Intelligent Vehicles Symposium*, 2023, pp. 1–8.

[8] J. Skilling, "Programming the hilbert curve," in *AIP Conference Proceedings*, American Institute of Physics, vol. 707, 2004, pp. 381–387.

[9] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the hilbert space-filling curve," *IEEE Transactions on knowledge and data engineering*, vol. 13, no. 1, pp. 124–141, 2001.

[10] H. Dai and H. Su, "Clustering analyses of two-dimensional space-filling curves: Hilbert and z-order curves," *SN Computer Science*, vol. 4, no. 1, p. 8, 2022.

[11] J. K. Lawder and P. J. H. King, "Querying multi-dimensional data indexed using the hilbert space-filling curve," *ACM Sigmod Record*, vol. 30, no. 1, pp. 19–24, 2001.

[12] K. C. Lee, W.-C. Lee, B. Zheng, H. Li, and Y. Tian, "Z-sky: An efficient skyline query processing framework based on z-order," *The VLDB Journal*, vol. 19, pp. 333–362, 2010.

[13] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, pp. 45–77, Jan. 2007.

[14] A. Hevner, A. R, S. March, *et al.*, "Design science in information systems research," *Management Information Systems Quarterly*, vol. 28, pp. 75–, Mar. 2004.

[15] R. Schubotz, *zCurve: Multi-dimensional indexing using Morton space filling curves.* Version 0.0.4, May 2021. DOI: 10.5281/zenodo.4777584. [Online]. Available: https://github.com/rmrschub/zCurve.

[16] G. Galtay. "Hilbertcurve." (), [Online]. Available: https://github.com/galtay/hilbertcurve. (accessed: 19.04.2023).