Kinematic interpolation of movement data

Jed A. Long[1*]

[1]Department of Geography & Sustainable Development
School of Geography and Geosciences
University of St Andrews

[*]Corresponding author email: jed.long@st-andrews.ac.uk

Pre-print of published version.

Reference:
Long, JA. 2015. Kinematic interpolation of movement data. International Journal of Geographical Information Science.

DOI:
http://dx.doi.org/10.1080/13658816.2015.1081909

Abstract

Mobile tracking technologies are facilitating the collection of increasingly large and detailed datasets on object movement. Movement data are collected by recording an object's location at discrete time intervals. Often, of interest is to estimate the unknown position of the object at unrecorded time points to increase the temporal resolution of the data, to correct erroneous or missing data points, or to match the recorded times between multiple datasets. Estimating an objects unknown location between known locations is termed path interpolation. This paper introduces a new method for path interpolation termed kinematic interpolation. Kinematic interpolation incorporates object kinematics (i.e., velocity and acceleration) into the interpolation process. Six empirical datasets (two types of correlated random walks, caribou, cyclist, hurricane, and athlete tracking data) are used to compare kinematic interpolation to other interpolation algorithms. Results showed kinematic interpolation to be a suitable interpolation method with fast moving objects (e.g., the cyclist, hurricane, and athlete tracking data), while other algorithms performed best with the correlated random walk and caribou data. Several issues associated with path interpolation tasks are discussed along with potential applications where kinematic interpolation can be useful. Finally, code for performing path interpolation is provided (for each method compared within) using the statistical software R.

1. Introduction

New devices are providing scientists with unprecedented data on the movements of many different types of objects (e.g., humans, vehicles, and wildlife). Movement data are typically collected by recording an objects location at discrete time intervals, typically represented by the triple $\{x, y, t\}$, where x and y are spatial coordinates, and t is the time when the coordinates were recorded (Hornsby and Egenhofer 2002). Modern devices record movement data at increasingly detailed spatial and temporal resolutions, moving towards a continuous representation of the movement trajectory (Laube *et al.* 2007). With the rapid growth in availability of movement data, the field of GIScience has made significant contributions to methods for storing, indexing, visualizing, and analysing movement data, but yet there remain many areas for improving movement related research (Long and Nelson 2013a, Purves *et al.* 2014).

Movement analysis must consider that movement data are represented discretely, and thus the data represent only a sample of the object's true trajectory. When analysing movement data, problems can arise where data are missing, erroneous, or sampled with an irregular frequency (Laube and Purves 2011). In these cases, there is often a desire to estimate an objects unknown location using known data. The process of estimating unknown locations of a moving object along its trajectory is termed *path interpolation.* Methods for path interpolation have many practical applications in the analysis of moving objects. For example, researchers are commonly wishing to increase the temporal resolution of moving–object databases (Güting and Schneider 2005), often termed up–sampling (Turchin 1998). Up–sampling (and down–sampling) is useful, for example, when examining scale–dependencies in movement pattern indices (Laube and Purves 2011). Many methods for analysing joint movement patterns (e.g., Laube *et al.* 2005, Shirabe 2006, Benkert *et al.* 2008, Long and Nelson 2013b) require that the temporal sampling of multiple datasets match-up. In cases where multiple movement datasets do not perfectly match, path interpolation methods are useful for aligning the sampling resolutions of two or more movement datasets. Finally, path interpolation is used widely to deal with missing or erroneous data, which are commonly encountered in practice (Tremblay *et al.* 2006, Lonergan *et al.* 2009).

To date several methods have been proposed for path interpolation. The most commonly applied interpolation method is linear interpolation, which assumes that movement follows the straight–line path (bee–line) between two known points. Linear interpolation is advantageous because it can be straightforwardly implemented. The straight-line path between two points also represents the most–likely path of movement derived from random walk models (Winter and Yin 2010), making a strong theoretical argument for linear interpolation as well. Random–walk models have also been used to interpolate the movements of some animals, which are known to exhibit more random movement patterns (Wentz *et al.* 2003, Technitis *et al.* 2015). However, with many types of objects linear and random–walk models are inappropriate. For example, curvi–linear shapes (i.e., modelled using cubic–splines or Bézier curves) have been shown to be better at interpolating the movement trajectories of marine mammals (Tremblay *et al.* 2006). Similarly, Yu and Kim (2004, 2006) showed that polynomial curves improved interpolation of vehicle trajectories in comparison with linear interpolation methods.

The study of motion is commonly termed *kinematics* – which involves the use of a set of kinematic equations to describe the motion properties of an object (i.e., position, velocity, and acceleration) without considering the forces behind the motion. To date, the most significant advances in studying kinematic properties of moving–objects within GIScience have been extensions to Hägerstrand's (1970) classic time geographic model.

Kuijpers *et al.* (2011) provided the mathematical framework for altering the boundaries of the space–time cone and space–time prism to account for kinematic effects. Long *et al.* (2014a) extended the work of Winter and Yin (2010, 2011) to construct a probabilistic time–geographic model for calculating internal kinematic movement probabilities for the kinematic space–time cone. The approaches developed by both Kuijpers *et al.* (2011) and Long *et al.* (2014a) both fail to demonstrate how to delineate a kinematic path within a kinematic space–time prism. Current approaches to path interpolation fail to adequately consider the kinematic properties of the object, despite the fact that in many cases the kinematic characteristics of an object will influence the movement trajectory.

In this paper a new method is proposed for kinematic path interpolation that can be used to estimate the kinematic trajectory of an object from movement data. Here, it is hypothesized that kinematic path interpolation will be useful with datasets representing the movement of fast–moving objects, where data are collected with relatively fine sampling resolutions, for example with vehicles, cyclists, or athletes. The algorithm for performing kinematic path interpolation is derived and six empirical datasets are demonstrated to compare kinematic path interpolation with common existing approaches. Finally, a discussion of the results and general points for utilizing kinematic interpolation in other applications is provided.

2. Methods

The methods section begins with the development of the proposed kinematic interpolation method. Next, descriptions of four other commonly employed interpolation methods (Table 1) are provided for comparison: linear interpolation, constrained random walk, Bézier curves, and Catmull-Rom curves. These methods were chosen as they reflect the diversity of currently available methods for path interpolation, and are employed in a variety of situations. A contrived example is used to demonstrate each approach. Then, six empirical datasets are described which are used in order to evaluate the effectiveness of each interpolation method. The methods for evaluating each interpolation method are introduced, followed by a discussion of the computational efficiency of each method.

[Table 1 here]

2.1. *Kinematic Path Interpolation*

Consider the well documented situation where a moving object's position is recorded at discrete time intervals denoted $\mathbf{z}(t)$. The goal of path interpolation is to estimate the object's unknown location at some time t_u between known locations $\mathbf{z}(t_i)$ and $\mathbf{z}(t_j)$ where $t_i < t_u < t_j$. In order to perform kinematic interpolation we assume that the object has a known (or estimated) instantaneous velocity at time t_i (resp. t_j), denoted $\mathbf{v}(t_i)$ (resp. $\mathbf{v}(t_j)$). Kinematic interpolation builds from the equations that define kinematic motion in one dimension, that is $\mathbf{z}(t)$ and $\mathbf{v}(t)$ are segmented into 2 (or 3) independent dimensions (i.e., $\mathbf{z}(t) = (z_x(t), z_y(t))$ and $\mathbf{v}(t) = (v_x(t), v_y(t))$. Kinematic motion equations are straightforwardly extended to the 2-dimensional case by solving the system independently for each of the x and y components. The following kinematic equations can be used to describe kinematic motion in one dimension:

$$\text{position: } z(t_u) = z(t_i) + \int_{t_i}^{t_u} v(t)\ dt \tag{1}$$

$$\text{velocity: } v(t_u) = v(t_i) + \int_{t_i}^{t_u} a(t)\ dt \tag{2}$$

$$\text{acceleration: } a(t_u) = f(t_u) \tag{3}$$

where $f(t_u)$ is a function that describes the object's acceleration between t_i and t_j. Acceleration can change instantaneously, thus there are infinitely many ways in which we could describe an object's acceleration (and subsequently motion) via $f(t_u)$. Here it is proposed that $f(t_u)$ be modelled as a linear function of time in order to describe object acceleration as a smooth motion (i.e., no abrubt changes in speed or direction). This means that $f(t_u)$ describes the change in velocity between the two known locations as a monotonously increasing or decreasing linear function depending on $v(t_i)$ and $v(t_j)$. Specifically, $f(t_u)$ takes the form:

$$f(t_u) = b + m(t_u - t_i) \tag{4}$$

where b and m are two unknown parameters that represent the intercept and slope of the acceleration function. Back substitution of (4) into (1), (2), and (3) results in the equations:

$$z(t_u) = z(t_i) + v(t_i)(t_u - t_i) + \frac{b}{2}(t_u - t_i)^2 + \frac{m}{6}(t_u - t_i)^3 \tag{5}$$

$$v(t_u) = v(t_i) + b(t_u - t_i) + \frac{m}{2}(t_u - t_i)^2 \tag{6}$$

$$a(t_u) = b + m(t_u - t_i) \tag{7}$$

Given known values for $z(t_i)$, $z(t_j)$, $v(t_i)$, and $v(t_j)$ equations (5–7) can be used to set up a system of two equations in order to solve for the parameters b and m following:

$$v(t_j) - v(t_i) = b(t_j - t_i)^2 + \frac{m}{2}(t_j - t_i)^2 \tag{8}$$

$$z(t_j) - z(t_i) - v(t_i)(t_j - t_i) = \frac{b}{2}(t_j - t_i)^2 + \frac{m}{6}(t_j - t_i)^3 \tag{9}$$

Solving the system of equations described by equations (8) and (9) for b and m allows us to back–substitute b and m into (5) in order to interpolate the position of the object at time t_u. Once b and m are solved for, equation (5) can be applied recursively to estimate a continuous kinematic trajectory between $z(t_i)$ and $z(t_j)$. As stated previously, the process described by equations (1) to (9) is applied independently to each (spatial) dimension, and thus the parameters b and m are likely to be different across different dimensions.

2.2. *Linear interpolation*

Linear interpolation is conducted by estimating an object's unknown location along the straight–line path between two known locations (the bee–line). Linear interpolation has been implemented in numerous studies (e.g., Wentz *et al.* 2003, Delgado *et al.* 2014, Nelson *et al.* 2015), due to its straightforward calculation and interpretation. An unknown location at a specified time $z(t_u)$ is calculated following:

$$\mathbf{z}(t_u) = \mathbf{z}(t_i) + \frac{t_u - t_i}{t_j - t_i}\left(\mathbf{z}(t_j) - \mathbf{z}(t_i)\right) \tag{10}$$

Linear interpolation is a special case of kinematic interpolation where it is assumed that $\mathbf{v}(t_i) = \mathbf{v}(t_j) = \frac{\mathbf{z}(t_j) - \mathbf{z}(t_i)}{t_j - t_i}$ (i.e., constant motion) such that the parameters $b = m = 0$.

2.3. *Constrained Random Walk*

Random walks can be used to interpolate a moving object, whereby an interpolated position is dependent on the previous position. Random walks are generated by taking random samples from two distributions: a step-length distribution (l) and a turning angle distribution (θ) (Turchin 1998). The coordinates for a random walk position can be calculated simply as $z_x(t_u) = z_x(t_i) + l\cos(\theta)$ and $z_y(t_u) = z_y(t_i) + l\sin(\theta)$ (Turchin 1998). Such a random walk approach fails to consider the sequential nature of movement data, that is, that the object travels between two consecutive known locations $\mathbf{z}(t_i)$ and $\mathbf{z}(t_j)$. Thus, here a specific type of random walk is chosen, where the interpolation is constrained to the time geographic space-time prism (Hägerstrand 1970). Using the space-time prism to constrain random walks was first employed by Wentz *et al.* (2003), and further developed by Technitis *et al.* (2015), both in the context of interpolating paths from wildlife tracking data. As the object must move from location $\mathbf{z}(t_i)$ to $\mathbf{z}(t_j)$ the space-time prism is used to constrain the potential points included in the random walk (i.e., it is not completely random). To calculate such a constrained random walk, the potential point area for t_u is computed by intersecting the forward and past space-time cones from t_i to t_u and t_j to t_u respectively (Technitis *et al.* 2015). The constrained random walk algorithm is implemented by randomly selecting a location within the potential point area for t_u. When more than one point is to be interpolated, the time geographic constrained random walk algorithm accounts for the tendency of a random walk to wander around the first point and then make a bee–line to the second point (Wentz *et al.* 2003, Technitis *et al.* 2015) by randomly ordering the t_u's to be interpolated. The algorithm requires a parameter (v_{max}) representing the upper bound on mobility (i.e., velocity) which here was estimated as the maximum of either the 25th percentile of all segment velocities or 1.25× the observed segment velocity between t_i and t_j to account for differences between faster and slower movement periods. For more details on the time geographic constrained random walk see Technitis *et al.* (2015).

2.4. *Bézier Curve*

A cubic Bézier curve algorithm has been shown to be an effective interpolation method for some types of moving objects (Tremblay *et al.* 2006). Calculating the cubic Bézier curve requires definition of four anchor points, two of which are the origin $P_1 = \mathbf{z}(t_i)$ and destination point locations $P_4 = \mathbf{z}(t_j)$, and the other two (P_2, P_3) control the shape of the curve. Here, the approach taken for computing the Bézier control points (P_2, P_3) is based on the intial and exit velocities (see below). Such an approach makes the Bézier curve (as implemented here) comparable with kinematic interpolation, in that they use the same information.

$$P_2 = \mathbf{z}(t_i) + \mathbf{v}(t_i)\frac{1}{2}(t_j - t_i) \tag{11}$$

$$P_3 = \mathbf{z}(t_j) - \mathbf{v}(t_j)\frac{1}{2}(t_j - t_i) \tag{12}$$

$$\mathbf{z}(t_u) = (1-\delta)^3 P_1 + 3(1-\delta)^2 \delta P_2 + 3(1-\delta)\delta^2 P_3 + \delta^3 P_4 \tag{13}$$

where $\delta = \frac{t_u - t_i}{t_j - t_i}$, δ being simply the time we wish to interpolate scaled to unity.

2.5. *Catmull-Rom Curve*

Another curve-based option is the Catmull-Rom curve (Barry and Goldman 1988, Yuksel *et al.* 2011) which is a special type of cubic-spline that can be used to straightforwardly interpolate between known data points. The principal advantage of the Catmull-Rom curve approach in the context of path interpolation is that the observed movement locations are used directly as control points for the interpolated curve. Catmull-Rom curves can be used to estimate the location of an object at t_u based on four control points defined by $\mathbf{z}(t_{i-1}), \mathbf{z}(t_i), \mathbf{z}(t_j), \mathbf{z}(t_{j+1})$ (see Supplementary Material A for derivation).

2.6. *Example*

To demonstrate kinematic interpolation, a contrived example is used to compare the interpolated locations from the kinematic algorithm to the other methods. In this contrived scenario, consider a sequence of four points, where an object begins at the point $\mathbf{z}(0) = (0, -3)$ with a velocity of 0 m/s and then moves to the origin $\mathbf{z}(1) = (0, 0)$ with a velocity of 3 m/s to the North $\mathbf{v}(1) = (0, 3)$. The object reaches position $\mathbf{z}(6) = (10, 10)$ after 5 s, it now has a velocity of 3 m/s to the East $\mathbf{v}(6) = (3, 0)$ and it continues on to location $\mathbf{z}(7) = (13, 10)$. Using each interpolation method the location of the object is estimated at 1/2 s intervals from $t_u = 1$ to 6 s in order to show the shape of the interpolated trajectory resulting from each method (Figure 1).

[Figure 1 here]

The differences between each interpolation method and the kinematic interpolation method proposed are readily observed from this contrived example. The linear interpolation algorithm follows the 'bee-line' path between the two known points (Figure 1a). The constrained random walk wanders within the space in between the known points as defined by the space-time prism (Figure 1b). The Bézier method results in a curved trajectory that is more exaggerated than the kinematic path, but one that seems to incorporate the initial and final velocities (Figure 1c). The Catmull-Rom curve is the closest to the kinematic curve, in this case, with only small differences observed (Figure 1d).

2.7. *Data*

Six empirical datasets are used to compare and contrast the new kinematic path interpolation algorithm with the other methods (Figure 2). The first two datasets are generated via simulations using correlated random walks that exhibit a low ($r = 0.2$) or high ($r = 0.9$) level of correlation in movement. Correlated random walks are commonly used to model animal movement, and have been used in many studies (e.g., Fauchald and Tveraa 2003, Rowcliffe *et al.* 2012, Long *et al.* 2014b) to compare different methods for analysing movement data. The third dataset tracks the movement of a caribou in northern British Columbia, Canada over a one year period. Caribou locations were recorded every 4 h using satellite telemetry. The fourth dataset represents the movement of a cyclist within an urban environment. Cyclist locations were recorded using a GPS with a 1/5 Hz sampling rate. The fifth dataset shows the movement of hurricane Katrina at 3 h intervals between 21:00 on 26-Aug-2005 and 21:00 29-Aug-2005. The point location of the hurricane was calculated as the centroid of the eye of the hurricane, obtained from the NOAA H*WIND data product (Powell *et al.* 1998, 2010). The final dataset shows the movement of an athlete playing Ultimate Frisbee. The athlete tracking data were recorded using a sport–specific GPS device (GPSports, Fyshwick, Australia) with a 5 Hz sampling rate.

[Figure 2 here]

2.8. *Interpolation Testing*

In order to test the effectiveness of each path interpolation algorithm an approach similar to previous interpolation studies (e.g., Wentz *et al.* 2003, Tremblay *et al.* 2006) is followed. A known point $z(t_k)$ (or sequence of points, $k \in \{1, 2, 3, ...\}$) is removed and then estimated via interpolation from the surrounding points. Thus, the procedure utilizes a sequential moving window approach to removing k known points and subsequently re-estimating them by each of the five interpolation methods (a similar approach to Long *et al.* 2014a). As the number of consecutive fixes to be estimated (k) is increased the interpolation task becomes more difficult.

With some tracking devices instantaneous velocities may be recorded alongside the location points, but typically this is not the case. In cases where instantaneous velocities are unknown, the initial velocity $\mathbf{v}(t_i)$ and final velocity $\mathbf{v}(t_j)$ can be estimated using the distance between two consecutive fixes and dividing it by the time difference. Specifically here instantaneous velocities are estimated for $\mathbf{v}(t_i)$ and $\mathbf{v}(t_j)$ using the observed tracking data. That is, $\mathbf{v}(t_i) = \frac{\mathbf{z}(t_i)-\mathbf{z}(t_{i-1})}{t_i - t_{i-1}}$ and $\mathbf{v}(t_j) = \frac{\mathbf{z}(t_{j+1})-\mathbf{z}(t_j)}{t_{j+1} - t_j}$.

To evaluate interpolation performance two measures of overall assessment are used. The first measure is the root mean squared error (rmse) of the error between the interpolated locations and the known points, where error is defined simply as the spatial euclidean distance between an interpolated point estimate and the known location. The second measure is the proportion of the points in the interpolation where a given method performed best (P_{best}). The P_{best} measure is comparative, allowing direct comparison between the five methods for each interpolated location. To test across a range of interpolation difficulties, the interpolation testing procedure described above was implemented on each of the six datasets using values of k ranging from $k = 1, ..., 10$, representing increasing interpolation difficulty.

2.9. *Examination of Computational Efficiency*

The complexity of each method was investigated, along with a time-trial, to examine the computational efficiency of each method. All analysis was conducted using the statistical computing software R (R Development Core Team 2015), and the code for each algorithm is available in the Supplementary Material. To test the computational efficiency of each algorithm a scenario was derived (similar to that in the contrived example) where the number of points to be interpolated was set to $k = 1{\times}10^6$, $1{\times}10^7$, $1{\times}10^8$, and $2{\times}10^8$. Each scenario was run 100 times, and the average of these runs is reported in Supplementary Material B. In the case of the constrained random walk, the algorithm is much slower, and for comparison $k = 1{\times}10^1$, $1{\times}10^2$, $1{\times}10^3$, and $2{\times}10^3$ was used. The results were realised on a standard desktop PC (Intel QuadCore i7 3770 CPU @ 3.40 GHz, with 16 Gb of RAM, on Windows 7) running R version 3.1.2.

3. Results

3.1. *Interpolation Testing*

For the first correlated random walk (CRW1) the linear method had the lowest rmse, while the time geographic constraind random walk had the highest rmse, for all values of k (Figure 3a). As would be expected, the level of rmse increased with k for all methods, and this was consistent across the different datasets. With the more correlated (i.e., smoother) random walk (CRW2) the linear, kinematic, and Catmull-Rom curve methods provided nearly identical results, and the kinematic and Catmull-Rom curve methods resulted in lower rmse at higher values of k (Figure 3b). With the caribou data, the linear method resulted in the lowest rmse, followed by the Catmull-Rom method. With the caribou data, the Bézier curve method (Figure 3c) showed the highest rmse. With the cyclist data the kinematic and Catmull-Rom methods performed nearly identically, with the lowest rmse, followed by the Bézier, linear, and constrained random walk (Figure 3d). In the hurricane Katrina dataset, again the Catmull-Rom and kinematic methods performed similarly, but for larger k these two methods resulted in much lower rmse than the other three methods (Figure 3e). Finally for the athlete dataset at low k values the linear, Catmull-Rom, and kinematic methods perform nearly identical, but as k increases, the kinematic and the Catmull-Rom methods have lower rmse than the linear, and other methods (Figure 3f).

[Figure 3 here]

Looking at the P_{best} measure of fit, in CRW1 linear had the best fit for about 40% of the interpolation points, while constrained random walk was best in about 30% of the points (Figure 4a). With CRW2 linear had the best fit about 30–40% of the points, while Bézier was best at higher values of k (Figure 4b). The linear method performed even better with the caribou data, having the P_{best} estimate upwards of 40% of the interpolations, while the constrained random walk had 30% and the other methods around 10% each (Figure 4c). In the cyclist data, the linear and Catmull-Rom were very close for $k = 1$ with P_{best} about 30% each, however as k increases the kinematic method produced similarly good results (Figure 4d). With the hurricane Katrina dataset a more unpredictable pattern emerges; at low values for k the linear method had the highest P_{best} values, while for higher k the Catmull-Rom, and kinematic method performed better (Figure 4e). In the athlete data, the linear method performed best with $P_{best} \simeq 40\%$ with $k = 1$, but at higher k the Catmull-Rom and kinematic methods again were best performing approximately equally well (Figure 4f).

[Figure 4 here]

3.2. *Computational Efficiency*

Each of the five methods employed here are $O(n)$ complex. The linear, Bézier curve, Catmull-Rom curve, and kinematic methods are all relatively fast and easy to compute (see Supplementary Material B). The linear method is the fastest, followed by the Catmull-Rom curve, then kinematic, and finally the Bézier curve, but the differences between these four are negligible in practical scenarios (e.g., $1{\times}10^6$ interpolations in < 1 second). The constrained random walk method, however, takes much longer (e.g., the time taken for interpolating 10 points using the constrained random walk method was comparable to interpolating $1{\times}10^6$ points using the other methods). This difference is likely due to the additional requirement of intersecting the forward and past space-time cones for each interpolation which is a computationally expensive operation (more information on the performance of constrained random walk algorithm can be found in Technitis *et al.* 2015). Thus, kinematic interpolation is a fast and computationally efficient interpolation method, in-line with, or better than, existing approaches.

4. Discussion

With different types of movement processes different models are expected to be more appropriate. As hypothesized, the kinematic interpolation method performed best with fast moving objects where kinematic properties are known to influence movement (e.g., cyclists and athletes). With cyclists it is somewhat surprising that kinematic outperformed linear interpolation given the linear shape of cyclist movement along road networks. This result is owed to the effect of changes in speed (e.g., slowing down or speeding up), which are appropriately modelled via kinematic interpolation and are ignored in the linear method. It was also found that kinematic interpolation may be useful for other types of objects, for example hurricane movements as shown here. Recent studies have focused on analysing spatial–temporal patterns in large collections of hurricane tracks (Dodge *et al.* 2012, Buchin *et al.* 2012). Here kinematic interpolation may provide a useful tool for up–sampling such analysis or comparing data with differing temporal resolutions. Here, the similar, but different outcomes of three curve-based interpolation algorithms: Bézier curves, Catmull-Rom curves, kinematic interpolation, are clearly demonstrated. Our results suggest that Catmull-Rom and kinematic curves are nearly equivalent, and most appropriate with fast-moving objects, producing very similar rmse values in the empirical examples shown. Bézier curves are likely more useful only in specific scenarios, which is surprising given that they performed well in the study by Tremblay *et al.* (2006).

It may be unsurprising that the curve-based methods (i.e., Bézier and Catmull-Rom) and kinematic method out-performed linear interpolation. One reason for this is that the Bézier, Catmull-Rom, and kinematic methods all take into consideration the surrounding points in some way. Here, the surrounding points (i.e., $\mathbf{z}(t_{i-1})$ and $\mathbf{z}(t_{j+1})$) were used to estimate the initial velocities used in both the Bézier and kinematic methods. In the Catmull-Rom algorithm, the points $\mathbf{z}(t_{i-1})$ and $\mathbf{z}(t_{j+1})$ are used directly in the calculation. The constrained random walk, also uses ancillary information in the form of the v_{max} paramater. In this sense the comparisons made here are somewhat unfair to the linear method, as it uses the least amount of information in its calculation.

Movement data are typically recorded as discrete (i.e., x, y, t) points and analysis methods are then influenced by the granularity (i.e., temporal resolution) at which data are collected. The equations for kinematic motion, as implemented here, assume acceleration to be a linear function of time. This assumption is reasonable when movement data are recorded at relatively high sampling resolutions (i.e., the cyclist data and athlete data here). The idea that two consecutive points in a movement dataset can be related to each other through kinematic equations can be thought of as *kinematic dependence*. With many movement data examples, the assumption of kinematic dependence is unrealistic, for example here with caribou data collected at temporal resolution of 4 *h*. With data where kinematic dependence is not present, it is unlikely that kinematic interpolation will be useful. However, note that kinematic dependence is not entirely dependent on the tracking interval. With some objects with coarse tracking intervals their kinematics may still be relevant, for example, with hurricanes as shown by the hurricane Katrina example. This is related to the joint effects of the objects velocity and size, along with the acceleration and turning ability of the object. With athletes this concept is often characterized using the term *agility*.

With the datasets employed here instantaneous velocities were not known and were estimated from the data. However, when instantaneous velocity is estimated from the data, the velocity estimation is highly dependent on the temporal resolution of the movement data (Laube and Purves 2011). The process of estimating instantaneous velocities likely influenced the resulting interpolations; using an example where instantaneous velocities were known would improve the performance of kinematic interpolation. In studying marine mammals it is useful to deploy a "dead reckoning" tag (which usually consists of an accelerometer, a magnetometer, a time-depth recorder, and other components) alongside GPS devices to help interpolate between fixes, as marine mammals can only be tracked via GPS when they surface (Wilson *et al.* 2007, Nordstrom *et al.* 2013). Several approaches have been proposed for modelling the interpolated path combining GPS and accelerometer data (e.g., Wilson *et al.* 2007, Liu *et al.* 2014). Similarly, the GPS units commonly used to study the movements of athletes often incorporate high frequency tri-axial accelerometers into the unit (Barbero-Alvarez *et al.* 2010, Coutts and Duffield 2010). New approaches capable of integrating kinematic interpolation with accelerometer data may provide improved estimates of fine–scale movements from tracking data where both GPS and accelerometer data are recorded.

Perhaps the most interesting potential development from this work is the potential to build upon the work of Kuijpers *et al.* (2011) and Long *et al.* (2014a) in order to study kinematic probabilities within the kinematic space–time prism. Using simulated kinematic trajectories, kinematic probabilities can be defined within the kinematic space-time prism, where probability is defined by the amount of energy or *work* (from classical mechanics, Goldstein *et al.* 2001) required to undergo a movement trajectory. Trajectories requiring more *work* can be modelled as having lower movement probability in order to construct a probabilistic kinematic space–time prism, similar to that proposed by Winter and Yin (2010, 2011). Such developments will have the benefit of being grounded in underlying movement theory (i.e., kinematic motion equations), and would be appropriate for several types of fluid movement (e.g., athletes and hurricanes described here).

The results highlight the value of the traditional linear method for interpolation. Simply put, in many scenarios the linear method performed as well as or better than other more complex methods. With animal tracking data, especially land mammals whose movement are frequently modelled via random walks (Codling *et al.* 2008), the linear method remains a suitable choice. Tremblay *et al.* (2006) showed that the Bézier interpolation out–performed linear interpolation with marine mammals, however, the results (i.e., CRW2) suggest the kinematic or Catmull-Rom approach may be a suitable alternative to Bézier curve methods with marine mammals and other species exhibiting curvi–linear movement patterns. With moving object datasets that are both fast moving and represented by data at an appropriately fine temporal resolution, kinematic interpolation is an appropriate alternative method for interpolating object locations.

5. Conclusion

This paper proposes a new method for interpolating movement data trajectories, one that is based on object *kinematics*. Using several empirical datasets reflecting different types of movement scenarios, this research highlights where kinematic interpolation improves, or is comparable to, existing approaches. Further, this research demonstrates situations where existing and simpler methods (i.e., linear interpolation) may be more appropriate. Kinematic interpolation represents a suitable interpolation method with fast moving objects, where movement data is collected at a relatively high temporal resolution. Examples include cyclists, motorists, and athlete tracking data. Similarly, kinematic interpolation may be useful for tracking large objects (e.g., hurricanes) that display kinematic effects over broad spatial-temporal extents, or with datasets where curvi–linear movement patterns are present. Finally, perhaps the biggest contribution of kinematic interpolation is the opportunity for future research developing calculations for kinematic movement probabilities for the space-time prism. To assist other researchers wishing to perform path interpolation, code is provided (in the statistical software R) for each method described herein.

Acknowledgements

References

Barbero-Alvarez, J.C., *et al.*, 2010. The validity and reliability of a global positioning satellite system device to assess speed and repeated sprint ability (RSA) in athletes.. *Journal of science and medicine in sport / Sports Medicine Australia*, 13 (2), 232–5.

Barry, P.J. and Goldman, R.N., 1988. A recursive evaluation algorithm for a class of

Catmull-Rom splines. *ACM SIGGRAPH Computer Graphics*, 22 (4), 199–204.

Benkert, M., *et al.*, 2008. Reporting flock patterns. *Computational Geometry*, 41 (3), 111–125.

Buchin, M., Dodge, S., and Speckmann, B., 2012. Context-aware similarity of trajectories. *In*: N. Xiao, M.p. Kwan, M.F. Goodchild and S. Shekhar, eds. *7th International Conference of GIScience*, 612 Columbus, OH: Springer, LNCS 7478, 43–56.

Codling, E.a., Plank, M.J., and Benhamou, S., 2008. Random walk models in biology. *Journal of the Royal Society Interface*, 5 (25), 813–834.

Coutts, A.J. and Duffield, R., 2010. Validity and reliability of GPS devices for measuring movement demands of team sports. *Journal of Science and Medicine in Sport*, 13, 133–135.

Delgado, M.D.M., *et al.*, 2014. A statistical framework for inferring the influence of conspecifics on movement behaviour. *Methods in Ecology and Evolution*, 5 (2), 183–189.

Dodge, S., Laube, P., and Weibel, R., 2012. Movement similarity assessment using symbolic representation of trajectories. *International Journal of Geographical Information Science*, 26 (1), 1563–1588.

Fauchald, P. and Tveraa, T., 2003. Using First-Passage Time in the Analysis of Area-Restricted Abd Habitat Selection. *Ecology*, 84 (2), 282–288.

Goldstein, H., Poole, C., and Safko, J., 2001. *Classical Mechanics*. Third San Francisco: Addison Wesley.

Güting, R.H. and Schneider, M., 2005. *Moving Objects Databases*. The Morgan Kaufmann Series in Data Management Systems New York, NY: Elsevier.

Hägerstrand, T., 1970. What about people in regional science?. *Papers of the Regional Science Association*, 24 (1), 7–21.

Hornsby, K. and Egenhofer, M.J., 2002. Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence*, 36 (1-2), 177–194.

Kuijpers, B., Miller, H.J., and Othman, W., 2011. Kinetic space-time prisms. *In*: *19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* Nov. 1-4, Chicago, IL: Association for Computing Machinery, 162–170.

Laube, P., Imfeld, S., and Weibel, R., 2005. Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science*, 19 (6), 639–668.

Laube, P., *et al.*, 2007. Movement beyond the snapshot Dynamic analysis of geospatial lifelines. *Computers, Environment and Urban Systems*, 31 (5), 481–501.

Laube, P. and Purves, R.S., 2011. How fast is a cow? Cross-scale analysis of movement data. *Transactions in GIS*, 15 (3), 401–418.

Liu, Y., *et al.*, 2014. Bayesian Melding of the Dead-Reckoned Path and GPS Measurements for an Accurate and High-Resolution Path of Marine Mammals. 1–26.

Lonergan, M., Fedak, M., and McConnell, B., 2009. The effects of interpolation error and location quality on animal track reconstruction. *Marine Mammal Science*, 25 (2), 275–282.

Long, J.A. and Nelson, T.A., 2013a. A review of quantitative methods for movement data. *International Journal of Geographical Information Science*, 27 (2), 292–318.

Long, J.A. and Nelson, T.A., 2013b. Measuring dynamic interaction in movement data. *Transactions in GIS*, 17 (1), 62–77.

Long, J.A., Nelson, T.a., and Nathoo, F.S., 2014a. Toward a kinetic-based probabilistic time geography. *International Journal of Geographical Information Science*, 28 (5), 855–874.

Long, J.A., *et al.*, 2014b. A critical examination of indices of dynamic interaction for wildlife telemetry studies.. *Journal of Animal Ecology*, 83 (5), 1216–1233.

Nelson, T., *et al.*, 2015. A time geographic approach for delineating areas of sustained wildlife use. *Annals of GIS*, 21 (1), 81–90.

Nordstrom, C.a., *et al.*, 2013. Northern fur seals augment ship-derived ocean temperatures with higher temporal and spatial resolution data in the eastern Bering Sea. *Deep Sea Research Part II: Topical Studies in Oceanography*, 94, 257–273.

Powell, M.D., *et al.*, 1998. The HRD real-time hurricane wind analysis system. *Journal of Wind Engineering and Industrial Aerodynamics*, 77-78, 53–64.

Powell, M.D., *et al.*, 2010. Reconstruction of Hurricane Katrina's wind fields for storm surge and wave hindcasting. *Ocean Engineering*, 37 (1), 26–36.

Purves, R.S., *et al.*, 2014. Moving beyond the point: An agenda for research in movement analysis with real data. *Computers, Environment and Urban Systems*, 47, 1–4.

R Development Core Team, 2015. *R: A language and environment for statistical computing.*. Technical report, R Foundation for Statistical Computing, Vienna, Austria.

Rowcliffe, J.M., *et al.*, 2012. Bias in estimating animal travel distance: The effect of sampling frequency. *Methods in Ecology and Evolution*, 3 (4), 653–662.

Shirabe, T., 2006. Correlation analysis of discrete motions. *In*: M. Raubal, H.J. Miller, A. Frank and M.F. Goodchild, eds. *GIScience 2006. LNCS, vol. 4197.*, Vol. 4197 Berlin: Springer-Verlag, 370–382.

Technitis, G., *et al.*, 2015. From A to B , randomly: A point-to-point random trajectory generator for animal movement. *International Journal of Geographical Information Science.*

Tremblay, Y., *et al.*, 2006. Interpolation of animal tracking data in a fluid environment. *Journal of Experimental Biology*, 209 (1), 128–140.

Turchin, P., 1998. *Quantitative analysis of movement: measuring and modeling population* *redistribution in animals and plants*. Sinauer Associates.

Wentz, E., Campbell, A., and Houston, R., 2003. A comparison of two methods to create tracks of moving objects: linear weighted distance and constrained random walk. *International Journal of Geographical Information Science*, 17 (3), 623–645.

Wilson, R.P., *et al.*, 2007. All at sea with animal tracks; methodological and analytical solutions for the resolution of movement. *Deep Sea Research Part II: Topical Studies in Oceanography*, 54 (3-4), 193–210.

Winter, S. and Yin, Z.C.C., 2010. Directed movements in probabilistic time geography. *International Journal of Geographical Information Science*, 24 (9), 1349–1365.

Winter, S. and Yin, Z.C.C., 2011. The elements of probabilistic time geography. *Geoinformatica*, 15 (3), 417–434.

Yu, B. and Kim, S.H., 2006. Interpolating and using most likely trajectories in moving-objects databases. *In*: S. Bressan, J. Kung and R. Wagner, eds. *DEXA 2006, LNCS 4080* Springer-Verlag, 718–727.

Yu, B. and Kim, S., 2004. Curve-based representation of moving object trajectories. *In*: *International Database Engineering and Applications Symposium (IDEAS)* IEEE Computer Society, p. 7.

Yuksel, C., Schaefer, S., and Keyser, J., 2011. Parameterization and applications of Catmull-Rom curves. *CAD Computer Aided Design*, 43 (7), 747–755.

Tables and Figures

Table 1: Five interpolation methods employed in this study along with rationale for selection and selected reference(s).

Method	Rationale	Selected Reference(s)
kinematic	– Proposed here. – Uses kinematic motion equations to define interpolation.	This Paper
linear	– Most popular method and most computationally straightforward. – Special case of kinematic interpolation.	(Wentz *et al.* 2003, Tremblay *et al.* 2006)
constrained random walk	– Potentially useful for interpolating animal trajectories. – As implemented here, relates to time geographic framework for movement analysis.	(Wentz *et al.* 2003, Technitis *et al.* 2015)
Bézier curve	– Demonstrated to be effective in interpolating marine mammals. – Should be appropriate with curvi-linear paths.	(Tremblay *et al.* 2006)
Catmull-Rom curve	– Special type of cubic interpolating spline commonly used in computer graphics. – Many potential parameterizations, but Catmull-Rom parameterization especially useful in path interpolation.	(Barry and Goldman 1988)

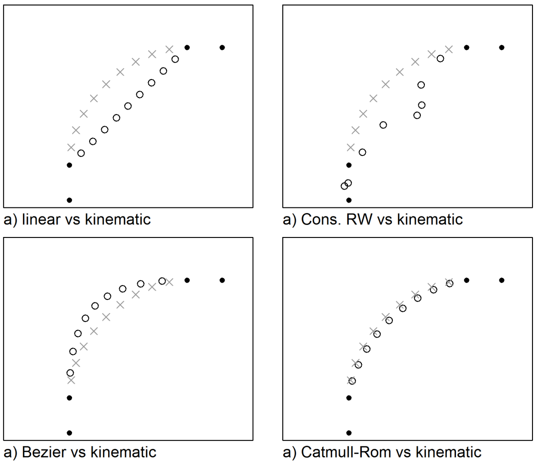

Figure 1. Contrived example of contrasting different path interpolation methods (open circles) with kinematic interpolation (grey crosses), known locations are denoted by filled circles. In the example, the object begins at the point $\mathbf{z}(0) = (0, -3)$ with a velocity of $0\ m/s$ and then moves to the origin $\mathbf{z}(1) = (0, 0)$ with a velocity of $3\ m/s$ to the North $\mathbf{v}(1) = (0, 3)$. The object reaches position $\mathbf{z}(6) = (10, 10)$ after $5\ s$, it now has a velocity of $3\ m/s$ to the East $\mathbf{v}(6) = (3, 0)$ and it continues on to location $\mathbf{z}(7) = (13, 10)$. The time difference between the initial and final point is $5\ s$ and the object's location is estimated every $1/2\ s$ in between.

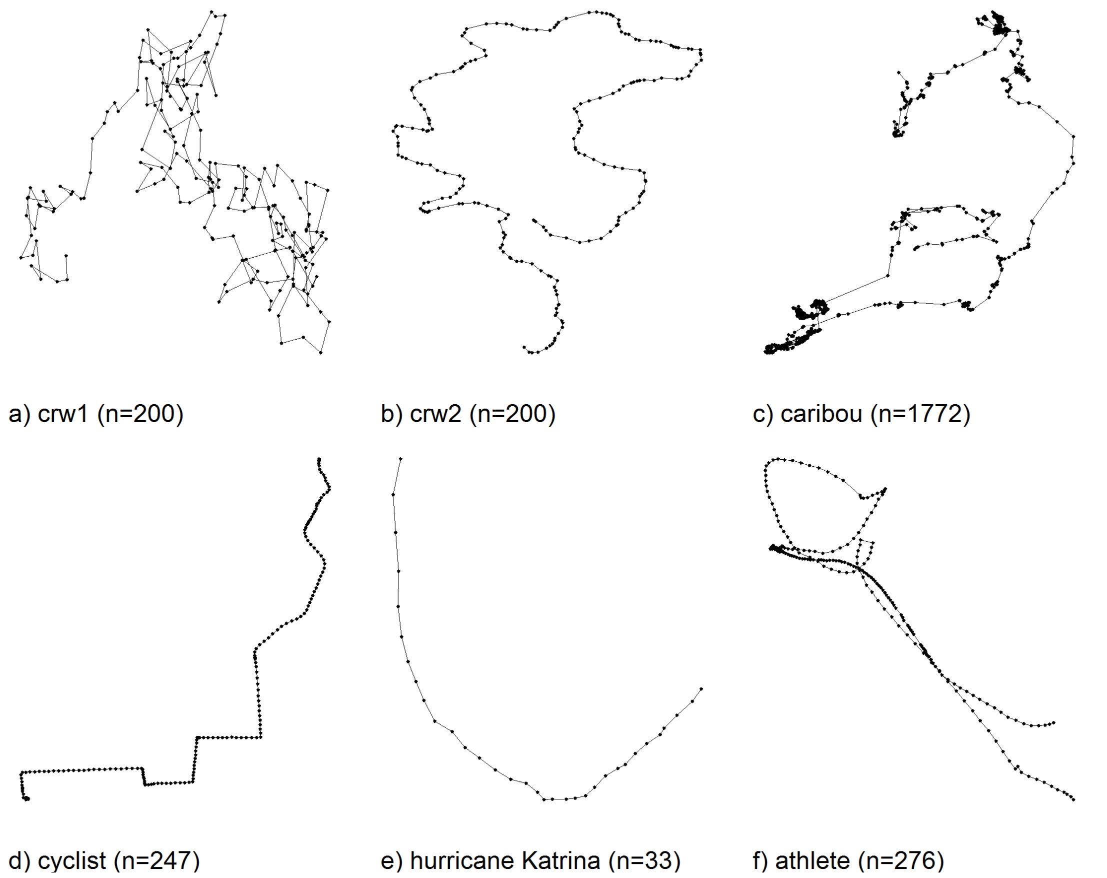

Figure 2. Six empirical datasets used for testing kinematic interpolation: a) crw1 ($n = 200$) , b) crw2 ($n = 200$), c) caribou ($n = 1772$), d) cyclist ($n = 246$), e) hurricane Katrina ($n = 33$), f) athlete ($n = 276$).

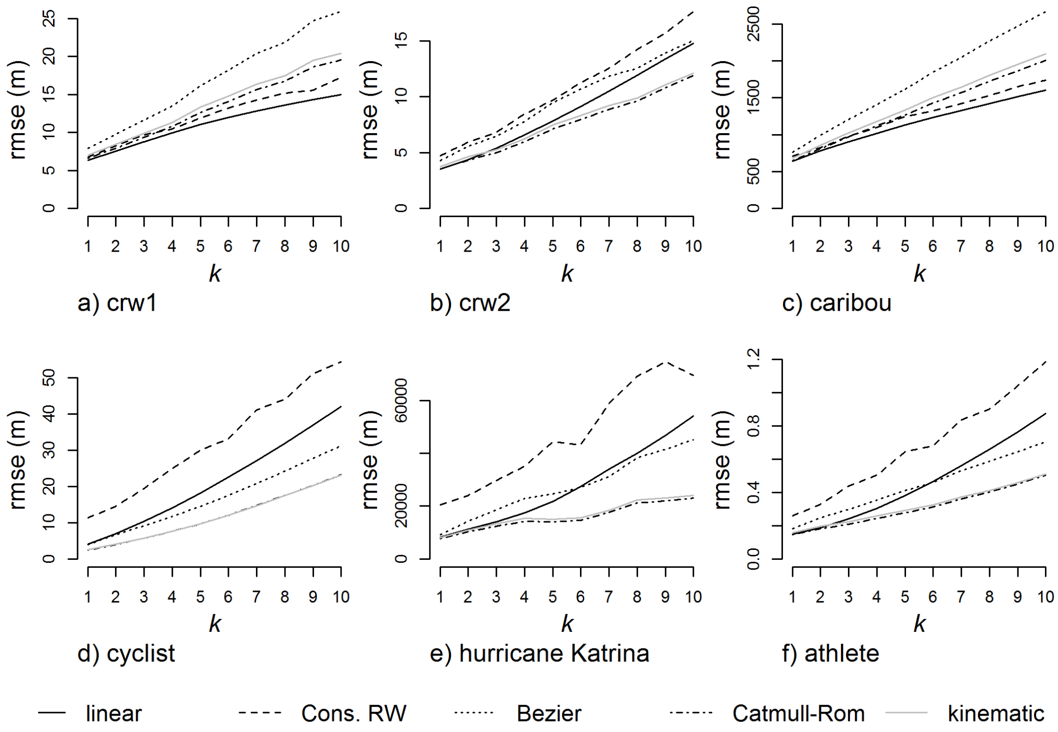

Figure 3. Root mean squared error (rmse; error defined as the euclidean distance between the interpolated and true location) for each interpolation method, for values of k from $1, ..., 10$, for each of the six datasets: a) crw1, b) crw2, c) caribou, d) cyclist, e) hurricane Katrina, f) athlete.

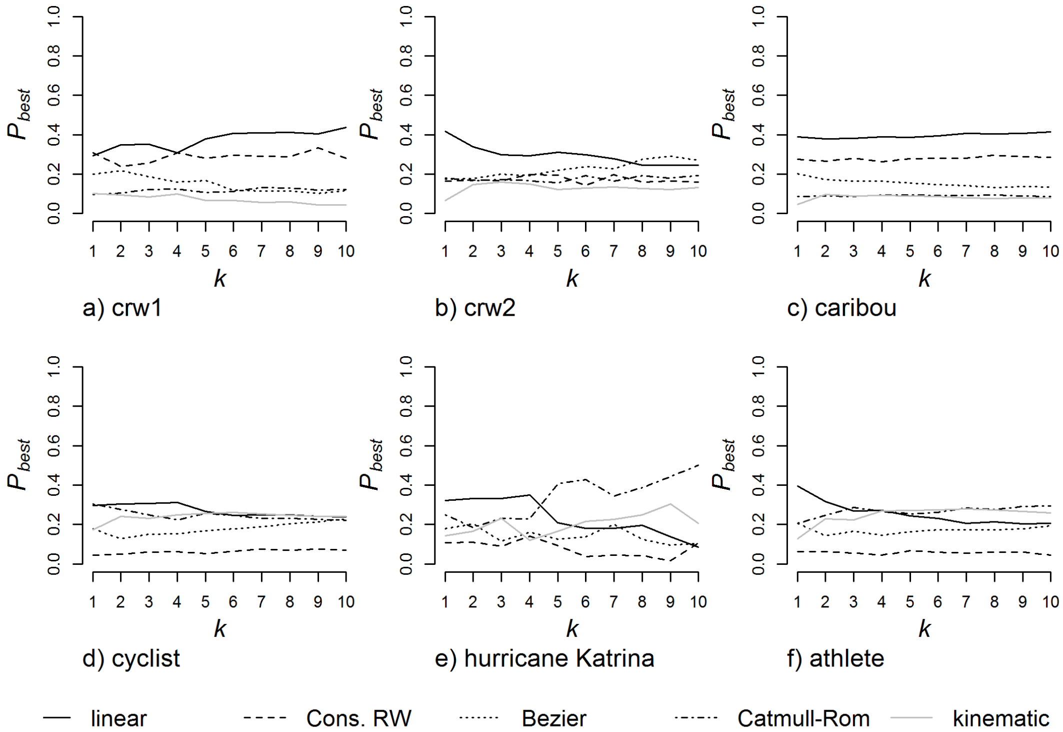

Figure 4. The proportion of interpolations where each method performed best (P_{best}), with values of k from $1, ..., 10$, for each of the six datasets: a) crw1, b) crw2, c) caribou, d) cyclist, e) hurricane Katrina, f) athlete.

Supplementary Material A

Derivation of Catmull-Rom curve.

Considering the four points $\mathbf{z}(t_{i-1}), \mathbf{z}(t_i), \mathbf{z}(t_j), \mathbf{z}(t_{j+1})$ where we wish to interpolate the position along the Catmull-Rom curve at t_u where $t_i < t_u < t_j$. The Catmull-Rom interpolation curve for path interpolation takes the following form:

$$\mathbf{z}(t_u) = \frac{t_j - t_u}{t_j - t_i} B1 + \frac{t_u - t_i}{t_j - t_i} B2$$

where

$$B1 = \frac{t_j - t_u}{t_j - t_{i-1}} A1 + \frac{t_u - t_{i-1}}{t_j - t_{i-1}} A2$$

$$B2 = \frac{t_{j+1} - t_u}{t_{j+1} - t_{i-1}} A2 + \frac{t_u - t_i}{t_{j+1} - t_i} A3$$

$$A1 = \frac{t_i - t_u}{t_i - t_{i-1}} \mathbf{z}(t_{i-1}) + \frac{t_u - t_{i-1}}{t_i - t_{i-1}} \mathbf{z}(t_i)$$

$$A2 = \frac{t_j - t_u}{t_j - t_i} \mathbf{z}(t_i) + \frac{t_u - t_i}{t_j - t_i} \mathbf{z}(t_j)$$

$$A3 = \frac{t_{j+1} - t_u}{t_{j+1} - t_j} \mathbf{z}(t_j) + \frac{t_u - t_j}{t_{j+1} - t_j} \mathbf{z}(t_{j+1})$$

Supplementary Material B

Computational efficiency test for each interpolation method, where k represents the number of interpolations performed. From this it can be seen that each of the methods are computed in $O(n)$ time. That is, a ten-fold increase in the number of points to be interpolated, is associated with a ten-fold increase in computation time, in all cases.

k	linear	Bézier	Catmull-Rom	kinematic	constrained RW*
1×10^6	0.017	0.247	0.061	0.124	0.067
1×10^7	0.136	2.261	0.538	1.227	0.676
1×10^8	1.637	23.438	5.283	12.622	6.820
2×10^8	2.981	46.415	10.631	25.137	13.872

*The constrained random walk algorithm is much slower than the other methods, and thus was compared using values of $k = 10, 100, 1000, 2000$.