International Journal of
Geo-Information

Article

Morphing of Building Footprints Using a Turning Angle Function

Jingzhong Li [1,*], Xingong Li [2] and Tian Xie [1]

[1] School of Resource and Environmental Sciences, Wuhan University, Wuhan 430072, China; xtnyforbless@hotmail.com

[2] Department of Geography and Atmospheric Science, University of Kansas, Lawrence, KS 66045, USA; lixi@ku.edu

* Correspondence: 00009232@whu.edu.cn; Tel.: +86-150-7231-5257

Academic Editor: Wolfgang Kainz
Received: 27 April 2017; Accepted: 13 June 2017; Published: 14 June 2017

Abstract: We study the problem of morphing two polygons of building footprints at two different scales. This problem frequently occurs during the continuous zooming of interactive maps. The ground plan of a building footprint on a map has orthogonal characteristics, but traditional morphing methods cannot preserve these geographic characteristics at intermediate scales. We attempt to address this issue by presenting a turning angle function-based morphing model (TAFBM) that can generate polygons at an intermediate scale with an identical turning angle for each side. Thus, the orthogonal characteristics can be preserved during the entire interpolation. A case study demonstrates that the model yields good results when applied to data from a building map at various scales. During the continuous generalization, the orthogonal characteristics and their relationships with the spatial direction and topology are well preserved.

Keywords: turning angle function; morphing; building footprint; continuous generalization

1. Introduction

In online and web-mapping contexts, maps are commonly used in browsers and mobile devices with different sizes, resolutions, and color models. This application requires the support of multiple representations of map contents and forms. There are two strategies for multiple representations: multiscale spatial databases (MSDBs) using database technology and online continuous generalization using efficient scale transformation algorithms. Most current map applications are based on limited and discrete MSDBs with pregeneralized map contents and several transitions from one scale to the next [1]. For example, Google Maps provides 20 levels of detail (LOD), which are pregeneralized in China to support user zooming to different scales, but the platform does not satisfy the requirements for mapping at an arbitrary scale. To overcome this deficiency, continuous generalization using an on-the-fly technology has emerged [2,3]. This approach allows users to zoom to different map scales interactively at arbitrary levels with smooth and continuous changes [4].

Continuous generalization denotes the use of various generalization techniques in real time to generate geographic information at arbitrary scales with smooth and continuous changes. This process creates a temporary, generalized dataset exclusively for visualization and not for storage or other purposes. There are three solutions to online continuous generalization: the traditional cartography generalization algorithm-based solution, the LOD-based solution, and the shape interpolation-based solution. The first solution implements continuous generalization by selecting and transforming the traditional cartography generalization algorithm for data processing in a network environment. For example, Sester and Brenner [2] defined a set of elementary generalization operations (EGOs) and used them for the continuous generalization of building areas on small mobile devices. The LOD-based solution relies on the accumulation of change to construct a multiscale entity model [5]. This model considers the spatial representation from one scale to another as an accumulation of a set of changes. The difference between two consecutive representations is recorded in a linear order, and target representation is achieved through the gradual addition or subtraction of "change patches". The third solution achieves continuous generalization based on scale interpolation by obtaining a map representation at a meaningful intermediate scale and interpolating between two anchor scales [6,7]. The interpolation result is coarser than that at the larger scale and finer than that at the smaller scale, thereby reflecting the fusion of the two basic scales.

In this paper, a solution for the continuous generalization of building plans is presented based on shape morphing. Shape morphing is also known as shape blending, shape metamorphosis, or shape fusion, and it is an important technique in computer graphics and computer vision. The morphing process can be defined as a gradual (i.e., over time) and smooth transformation of one key shape into another [8,9]. The essence of this method is shape interpolation. Considering the orthogonal boundary property of building plans, we propose turning the angle function-based morphing model (TAFBM) to continuously generalize building features. The remainder of the paper is organized as follows: Section 2 reviews the related works on morphing; Section 3 investigates the continuous generalization model of building features using the TAFBM technology; the experiment is detailed in Section 4; Section 5 presents the conclusions and suggestions for future improvements.

2. Related Work

According to different input data formats, morphing methods can be divided into two types: raster morphing and vector morphing. Raster morphing treats maps as images and is used to warp two map images and generate intermediate-scale images [10–13]. Vector morphing treats map features as basic elements and is used to generate intermediate-scale shapes of corresponding features [1,14,15]. This paper focuses on vector morphing, particularly the morphing of building features.

Two methods are commonly used in the literature to specify morphing schemes for vector data [14]. The first approach transforms two input geometric graphics into two implicit functions and generates the intermediate shapes by fusing the implicit functions with different parameters. For example, Hughes [16] presented a scheduled Fourier transform to perform morphing in which both basic states were represented as a combined function of high frequency and low frequency, and the morphing effect was achieved by the fusion of the two functions. He et al. [17] decomposed the two basic states into a set of frequency wavelet bands, applied a smooth interpolation to each band, and reconstructed the results to generate the intermediate state.

The second popular approach involves an explicit morphing scheme. Previous works have mainly focused on two aspects: finding meaningful correspondence between "features" of two key shapes and producing smooth interpolated shapes according to "trajectories" along which the features, which are associated with the intermediate shapes, "flow" from one key shape to another. In turn, the meanings of feature and trajectory depend on the descriptions of the key shapes and consequently their geometrical and topological properties [18]. Generally, the process of feature correspondence involves the detection of the characteristic points and establishment of correspondence between the geometric features of the source and the target shapes. In the field of computer graphics, the correspondence relationship is commonly computed based on physical energy minimization or geometric shape similarity [14,19,20]. These methods use local geometric properties, such as the vertex angle, edge length, or triangle area, to establish correspondence between two input shapes. However, these local geometric properties are not sufficient to describe the hidden geographical properties within geometric shapes [21].

The process of path interpolation in morphing ensures that the intermediate shapes gradually change along a specified trajectory, that the boundaries of the intermediates exhibit no shrinkage, and that the interiors are not distorted. The simplest method is linear interpolation, which is only suitable for simple shapes. Sederberg et al. [22] proposed an intrinsic interpolation method which can avoid shrinkage by interpolating the edge lengths and vertex angles of two input shapes. The interpolation method is presented to improve the effect of boundary interpolation using rigid motion and compatible triangulation [23–25].

The aforementioned morphing technologies mainly involve pure geometric shapes without considering the potential geographic features underneath the geometric details of geographic elements, such as the orthogonal geometric features of a building polygon. However, continuous map generalization is a process of geographic feature abstraction rather than focusing on pure geometric detail change [21]. In the area of map generalization, Van Kreveld [15] introduced two simple algorithms to morph a polyline into a straight-line segment. Cecconi and Galanda [26] obtained a one-to-one correspondence between two polyline vertices by inserting abundant vertices on a polyline with few vertices based on the proportional chord length principle. Nöellenburg et al. [6] presented a morphing method based on dynamic programming. Whited and Rossignac [27] used ball maps [28] to detect the corresponding points of two smooth objects, and five different trajectories (Hat, Linear, Tangent, Circular and Parabolic) were used for morphing. Recently, Deng and Peng [7] considered the entire structures of linear features during the process of feature correspondence and used straight-line trajectories to generate intermediate-scale linear features.

Most of these morphing methods are suitable for the continuous generalization of geographic features with smooth boundaries but not for building features with orthogonal boundaries. This paper presents a morphing method based on the turning angle function to continuously generalize building features. First, we transform the representation of vector coordinates in the spatial domain based on the turning angle function. Second, we construct the correspondence relationship between two turning angle functions, which consider both the geometric characteristics and changing regularity of the building boundaries. Finally, we compute the turning angle function at the intermediate scale and transform it to the corresponding geometric shape.

3. Methodology

3.1. Conceptual Model

For the same building entity at different scales, a is the original building footprint of detailed representation on a large scale $\mathrm{T_a}$, b is the simplified footprint of coarse representation at small scale $\mathrm{T_b}$, and Mid is the intermediate representation at the in-between scale $\mathrm{T_{Mid}}$. The morphing model can be denoted as $\mathrm{Mid = f(a, b, g)}$, where f is the shape interpolation function based on the turning angle function and g is a scale-related parameter that controls the degree of interpolation. When $0 \leq \mathrm{g} \leq 1, \mathrm{Mid}$ is monotonically and continuously related to g. A small value of g indicates that Mid is more similar to b, whereas a large value of g indicates that Mid is more similar to a. When $\mathrm{g = 1, Mid = a}$; when $\mathrm{g = 0, Mid = b}$; and when $0 < \mathrm{g} < 1$, Mid is between a and b. The relationship between g and the scale is as follows:

$$\mathrm{g} = \mathrm{h}(\frac{\frac{1}{\mathrm{T_{Mid}}} - \frac{1}{\mathrm{T_a}}}{\frac{1}{\mathrm{T_b}} - \frac{1}{\mathrm{T_a}}}) \tag{1}$$

where h is a function based on the particular relationship between Mid and $\mathrm{T_{Mid}}$ in continuous generalization applications. When $\mathrm{T_{Mid}} = \mathrm{T_a}$, g = 1 = h(0), and when $\mathrm{T_{Mid}} = \mathrm{T_b}$, $\mathrm{g = 0 = h(1)}$.

$$\begin{cases} \mathrm{H(0) = 1} \\ \mathrm{h(1) = 0} \end{cases} \tag{2}$$

In this conversion, the degree of morphing is controlled by the scale $\mathrm{T_{Mid}}$. The conceptual model can be implemented in three main steps: transformation of the vector data to the turning angle function, feature matching, and trajectory interpolation at the intermediate scale.

3.2. Transformation of Vector Data into the Turning Angle Function

The shape boundary of the vector format has N points. The steering angle function of the shape is the cumulative function of the steering angle; the detailed procedure is as follows. First, select a point P_0 within N as the start point. Second, construct the independent variable t, which is the arc length between the start point P_0 and every other point $P_i(i = 1,2,\cdots,N)$ along the shape boundary in the counterclockwise direction. Because the arc length between two adjacent points P_{i-1} and P_i is $t_{P_{i-1}P_i}$, t is represented as $\sum_{j=1}^{i} t_{P_{j-1}P_j}$, and its domain is $[0,\text{perimeter}]$. Third, construct the dependent variable θ, which is the accumulation of turning angles along the shape boundary from P_0. Here, we define the azimuth angle of side P_0P_1 as the turning angle of start point P_0, and the start orientation of this azimuth angle is the positive x-axis. The turning angle is positive in the counterclockwise direction and negative in the clockwise direction. Then, the turning angle function is represented as follows.

$$\theta = \theta(t)\begin{cases} t\epsilon\left[\sum_{j=1}^{i} t_{P_{j-1}}t_{P_j}, \sum_{j=1}^{i} t_{P_j}t_{P_{j+1}}\right], j \in [1,i] \\ t\epsilon\left[0, t_{P_0}t_{P_1}\right], j = 0 \end{cases} i \in [1,N] \tag{3}$$

Figure 1 is an example that illustrates the principal of the turning angle function, where the illustration on the left is an orthogonal polygon with equal sides of one unit. If we select point I as the start point P_0, polygon α can be represented in the form of the turning angle function, as shown on the right side of the figure. In this case, θ is the sum of the azimuth angle of each side and 2kπ (k is an integer). The initial value of k is zero; when θ is larger than 2π, it is equal to positive one, and when θ is smaller than zero, it will equal negative one. Additionally, different start points will result in different frequency curves. As shown in Figure 1, if point II is selected as the start point, the turning angle function and related frequency curve are different from those of point I. The selection of the start point will be discussed in the experimental section.

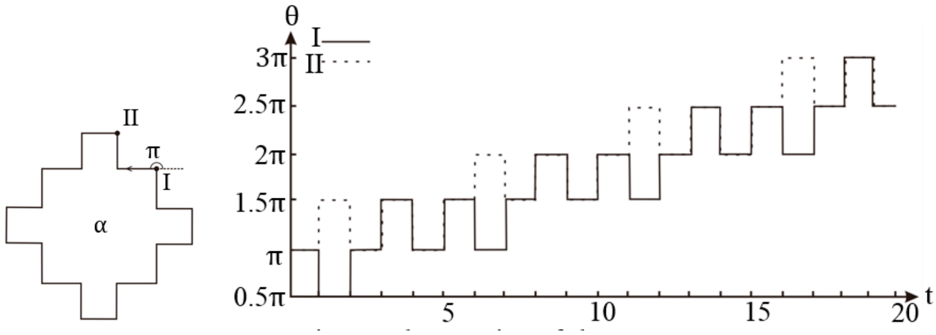


Figure 1. Turning angle function of shape α with different start points.

In an actual situation, an issue exists with the above turning angle function: different shapes of different perimeters have different domains of t. As shown in Figure 2, shapes α and β are two orthogonal polygons. Each side in α is one unit, and each side in β is three units. Their domains are [0,12] and [0,20], and their frequency curves overlap based on the turning angle function on the right side of the figure. To use the turning angle function for further analysis of the shape characteristic, we must unify the domains and convert them into [0,1] to construct a normalized turning angle function. The normalization method is shown in Equation (4). The post-normalization frequency curves are shown in Figure 3.

$$\theta = \theta(t), t \in \left[\frac{\sum_{j=1}^{i} t_{P_{j-1}P_j}}{\sum_{j=1}^{N} t_{P_{j-1}P_j}}, \frac{\sum_{j=1}^{i} t_{P_jP_{j+1}}}{\sum_{j=1}^{N} t_{P_{j-1}P_j}}\right] \tag{4}$$

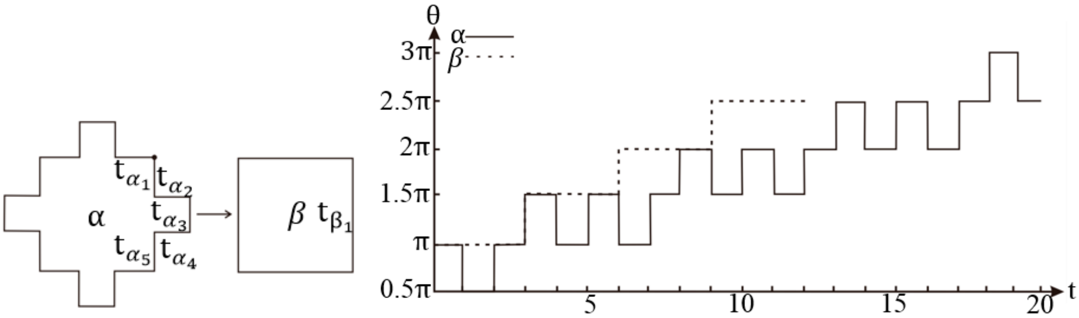


Figure 2. Overlap of the frequency curves of shapes α and β in different domains.

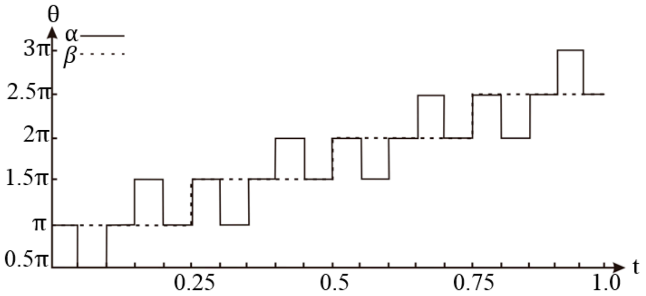


Figure 3. Overlap of the frequency curves of shapes α and β in a unified domain.

3.3. Matching of Sides

The turning angle function converts the geometric characteristics of a shape in the spatial domain into the frequency domain. By overlapping the frequency curves of two shapes, we can obtain their geometric differences and similarities, which will be used to guide the matching of sides. Shapes a and b have sides $\mathrm{t_a}$ and $\mathrm{t_b}$ and normalized turning angle functions $\theta = \theta_\mathrm{a}(\mathrm{t_a})$ and $\theta = \theta_\mathrm{b}(\mathrm{t_b})$, respectively. $\mathrm{t_a}$ and $\mathrm{t_b}$ satisfy the following conditions as homogeneous sides $\mathrm{t_{a_{same}}}$ and $\mathrm{t_{b_{same}}}$.

$$\begin{cases} \mathrm{t_a} \subseteq \mathrm{t_b} \\ \theta_\mathrm{a}(\mathrm{t_a}) = \theta_\mathrm{b}(\mathrm{t_b}) \end{cases} \tag{5}$$

In addition, $\mathrm{t_a}$ and $\mathrm{t_b}$ satisfy the following conditions as heterogeneous sides $\mathrm{t_{a_{diff}}}$ and $\mathrm{t_{b_{diff}}}$.

$$\begin{cases} t_a \subseteq t_b \\ \theta_a(t_a) \neq \theta_b(t_b) \end{cases} \tag{6}$$

According to the above definition, shape α in Figure 2 has eight heterogeneous sides: [0.05,0.1], [0.15,0.2], [0.3,0.35], [0.4,0.45], [0.55,0.6], [0.65,0.7], [0.8,0.85], and [0.9,0.95]. All other sides are homogeneous. The turning angle of the homogeneous sides may not be strictly equal at different scales due to the cartographic precision. Therefore, we must modify these matching conditions as follows:

$$\begin{cases} t_\mathrm{a} \subseteq t_\mathrm{b} \\ |\theta_\mathrm{a}(t_\mathrm{a}) - \theta_\mathrm{b}(t_\mathrm{b})| \leq \delta \end{cases} \tag{7}$$

And

$$\begin{cases} t_\mathrm{a} \subseteq t_\mathrm{b} \\ |\theta_\mathrm{a}(t_\mathrm{a}) - \theta_\mathrm{b}(t_\mathrm{b})| > \delta \end{cases} \tag{8}$$

where δ is the matching error, which influences the side-matching accuracy. The value of δ is determined by the generalization precision.

In the actual matching process, because of the normalized domain, the constraint on domain $\mathrm{t_a} \subseteq \mathrm{t_b}$ cannot be satisfied. For example, in Figure 4, the sides of shape γ in red cannot satisfy the above condition of $\mathrm{t_a} \subseteq \mathrm{t_b}$; therefore, these sides cannot be designated as homogeneous or heterogeneous sides. Furthermore, the difference in the shape perimeter may cause some mismatch. For example, during the transformation from γ to β, the sides in blue remain unchanged, but they are designated as heterogeneous sides. To solve these problems, we consider the geometric characteristics of the shape.

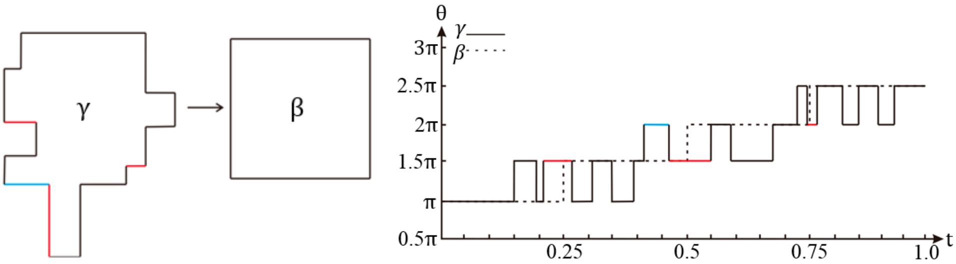


Figure 4. Matching of building sides based on the turning angle function.

The sides of shape a are given by $S_a\colon\{t_{a_i}|i\epsilon[0,N_a]\}$, where N_a is the number of sides of a. Additionally, the sides of shape b are given by $S_b\colon\{t_{b_j}|j\epsilon[0,N_b]\}$, where N_b is the number of sides of b. Considering the characteristics of scale ($T_a > T_b$), we have $N_a \geq N_b$, which implies that during the generalization process, one or more sides of shape a becomes a side in shape b. For example, in Figure 2, the sides $t_{\alpha_1}, t_{\alpha_2}, t_{\alpha_3}, t_{\alpha_4,}$ and t_{α_5} of shape α transform to side t_{β_1} of shape β. The sides of shape a can be divided into N_b subsets $S_{as}\colon\{t_{as_i} \mid i\epsilon Ns, Ns \subsetneq [0,N_a]\ \&i\epsilon[0,N_b]\}$ according to the segmentation condition, which will be described in the next paragraph. Now, we can formulate the matching problem as one-to-one mapping between S_{as} and S_b.

Because shapes a and b refer to the same geographical entity, they have similar spatial distributions. We can use the Euclidean distance as the basis to divide each shape into segments. The specific process is as follows. First, select a point $P_{b_i}(i \in [0,N_b])$ on shape b as the start point. Second, make a circle C with center point P_{b_i} and radius r (r is the distance threshold). Third, search for points on shape a within circle C, and select point P_{a_k} with the minimum distance as one splitting point of shape a. Then, push k into the splitting position of set U. Repeat this process until all points of shape b are traversed. Finally, we obtain a set U with N_b elements. Set U constitutes the segmentation condition. Using this method, we can obtain the closest points of $P_{b_0}, P_{b_1}, P_{b_2,}$ *and* P_{b_3} as $P_{\gamma_0}, P_{\gamma_1}, P_{\gamma_8},$ *and* $P_{\gamma_{13}}$ in Figure 5, and the splitting position set U = {0,1,8,13}, which divides the sides of shape γ into four subsets, as shown in Table 1. We can determine if each side in the four subsets is a homogeneous side or a heterogeneous side according to Equations (7) and (8). In Table 1, the sides in bold font are homogeneous sides, and the others are heterogeneous sides.

Table 1. Segmentation and classification of the shape sides.

	$\mathrm{T}_{\boldsymbol{as}}$	$\mathbf{T}_{\boldsymbol{b}}$
1	$\left\{t_{p_{\gamma 0} p_{\gamma 1}}\right\}$	$t_{p_{\beta_0} p_{\beta_1}}$
2	$\left\{t_{p_{\gamma_1} p_{\gamma_2}}, \boldsymbol{t}_{\boldsymbol{p}_{\gamma_2} \boldsymbol{p}_{\gamma_3}}, t_{p_{\gamma_3} p_{\gamma_4}}, \boldsymbol{t}_{\boldsymbol{p}_{\gamma_4} \boldsymbol{p}_{\gamma_5}}, t_{p_{\gamma_5} p_{\gamma_6}}, \boldsymbol{t}_{\boldsymbol{p}_{\gamma_6} \boldsymbol{p}_{\gamma_7}}, t_{\boldsymbol{p}_{\gamma_7} \boldsymbol{p}_{\gamma_8}}\right\}$	$t_{p_{\beta_1} p_{\beta_2}}$
3	$\left\{t_{p_{\gamma_8} p_{\gamma_9}}, \boldsymbol{t}_{\boldsymbol{p}_{\gamma_9} \boldsymbol{p}_{\gamma_{10}}}, t_{p_{\gamma_{10}} p_{\gamma_{11}}}, \boldsymbol{t}_{\boldsymbol{p}_{\gamma_{11}} \boldsymbol{p}_{\gamma_{12}}}, t_{\boldsymbol{p}_{\gamma_{12}} \boldsymbol{p}_{\gamma_{13}}}\right\}$	$t_{p_{\beta_2} p_{\beta_3}}$
4	$\{t_{p_{\gamma_{13}} p_{\gamma_{14}}}, \boldsymbol{t}_{\boldsymbol{p}_{\gamma_{14}} \boldsymbol{p}_{\gamma_{15}}}, t\ {p_{\gamma_{15}} p_{\gamma_{16}}}, \boldsymbol{t}_{\boldsymbol{p}_{\gamma_{16}} \boldsymbol{p}_{\gamma_{17}}}, t_{p_{\gamma_{17}} p_{\gamma_{18}}}, \boldsymbol{t}_{\boldsymbol{p}_{\gamma_{18}} \boldsymbol{p}_{\gamma_{19}}}, t_{\boldsymbol{p}_{\gamma_{19}} \boldsymbol{p}_{\gamma_0}}\}$	$t_{p_{\beta_3} p_{\beta_0}}$

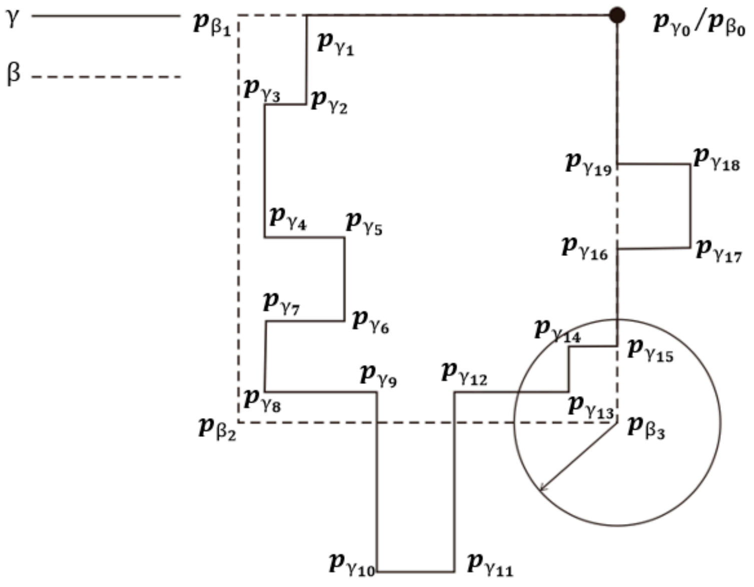


Figure 5. Illustration of the segmentation condition

3.4. Shape Interpolation

During the transformation from shape a to b in building feature generalization, the turning angles do not change, the number of homogeneous sides increases, and the number of heterogeneous sides decreases. Now, we present a model to express these changes and perform shape interpolation. The homogeneous side of shape a is $t_{\mathrm{same}_i}(i \in [0, \text{number of homogeneous sides}])$, and its increment is $\Delta t_{\mathrm{same}_i}$; the heterogeneous side of shape a is $t_{\mathrm{diff}_i}(i \in [0, \text{number of heterogeneous sides}])$, and its decrement is $\Delta t_{\mathrm{diff}_i}$. The transformation model from a to b is represented as follows:

$$\mathrm{Mid} = f(t_{\mathrm{same}_i}, t_{\mathrm{diff}_i}, \Delta t_{\mathrm{same}_i}, \Delta t_{\mathrm{diff}_i}) \tag{9}$$

where f is the turning angle function and t_{same_i} and t_{diff_i} can be obtained in the process of side matching. This model is equal to the mode of morphing $\mathrm{Mid} = f(a, b, g)$. During the transformation from shape a to b, the number of heterogeneous sides gradually decreases to zero, and the decrement is related to the scale parameter g.

$$\Delta t_{\mathrm{diff}_i} = -t_{\mathrm{diff}_i}(1 - g) \tag{10}$$

Similarly, the total increment of the homogeneous sides of shape a is as follows:

$$\Delta t_{\mathrm{sametotal}} = \sum_{i=1}^{NS} \Delta t_{\mathrm{same}_i} = t_{b_j} - t_{\mathrm{sametotal}} \tag{11}$$

where N_S is the number of sides of shape a, $t_{\mathrm{sametotal}}$ is the sum of homogeneous sides in shape a, t_{b_j} is the corresponding number of homogeneous sides in shape b, and $\Delta t_{\mathrm{sametotal}}$ is the increment of homogeneous sides in shape a. Notably, as the number of homogeneous sides in shape a increases, the increment decreases. In this model, the increment must be obtained from the segmented subsets of sides but not from a single side or the entire shape. The increment of the homogeneous sides $\Delta t_{\mathrm{same}_i}$ is also proportional to the length of side t_{same_i} and related to the scale parameter g.

$$\Delta t_{\mathrm{same}_i} = (1 - g)\frac{t_{\mathrm{same}_i}}{t_{\mathrm{sametotal}}}\Delta t_{\mathrm{sametotal}} \tag{12}$$

Finally, the continuous generalization model from shape a to b can be represented as a turning angle function at the intermediate scale.

$$\theta_{\mathrm{Mid}} = \theta(\mathrm{t})\begin{cases} \mathrm{t} = \mathrm{t}_{\mathrm{a}_{\mathrm{same}}} + (1-\mathrm{g})\dfrac{\mathrm{t}_{\mathrm{a}_{\mathrm{same}}}}{\mathrm{t}_{\mathrm{sametotal}}}(\mathrm{t}_{\mathrm{b}} - \mathrm{t}_{\mathrm{sametotal}}) & \mathrm{t} \in \text{homogeneous sides} \\ \mathrm{t} = \mathrm{g}\mathrm{t}_{\mathrm{a}_{\mathrm{diff}}} & \mathrm{t} \in \text{heterogeneous sides} \end{cases} \tag{13}$$

In Figure 6, shape a at the large scale has five sides {a_1,a_2,a_3,a_4,a_5}, while shape b at the small scale has one corresponding side b_1. After side matching, subset {a_1, a_5} includes the homogeneous sides, and {a_2,a_3,a_4} includes the heterogeneous sides. Given the parameter g = 0.5, the length of sides a_1, a_2, a_3, a_4, a_5, and b_1 are 41 units, 42 units, 39 units, 56 units, 59 units, and 136 units, respectively. According to the above interpolation model, we obtain $\Delta a_1 = 7.38, \Delta a_5 = 10.62, \Delta a_2 = -21, \Delta a_3 = -19.5, \text{and } \Delta a_4 = -28$. During the transformation from shape a to b, the number of homogeneous sides gradually increases until they are connected, and the number of heterogeneous sides gradually decreases to zero.

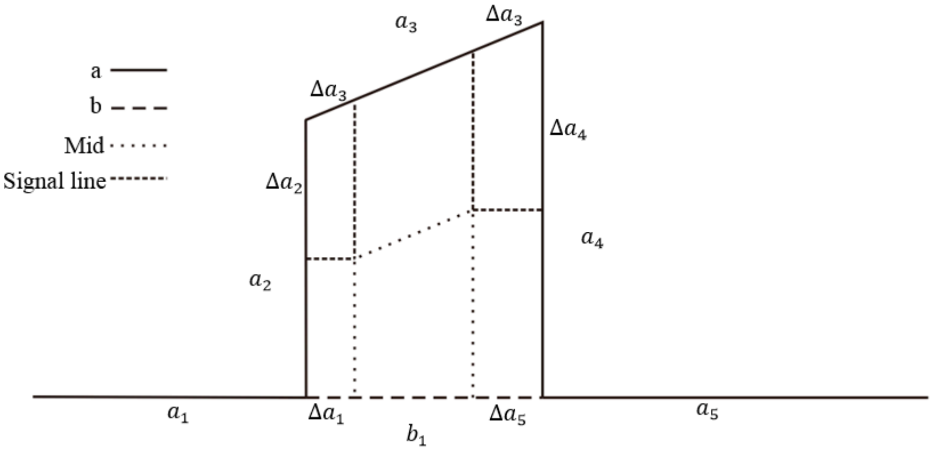


Figure 6. Interpolation of the intermediate scale based on the turning angle function-based morphing model (TAFBM).

4. Case Study

We evaluate our algorithms using two groups of experiments: one regarding the principle of selecting the start point and the other regarding the application of TAFBM for continuous generalization. All experiments are performed on a Dell OptiPlex 3020 MT PC (micro tower personal computer) with an Intel(R) Core(TM) i5-4590 CPU (central processing unit) and a memory of 8 GB in Microsoft Windows 7. The related TAFBM algorithms are implemented in C++ and compiled with VS2012.

4.1. Selection of the Start Point of the Turning Angle Function

The selection of start points is important in the representation of using the turning angle function. Here, we present four basic principles in the selection of a start point:

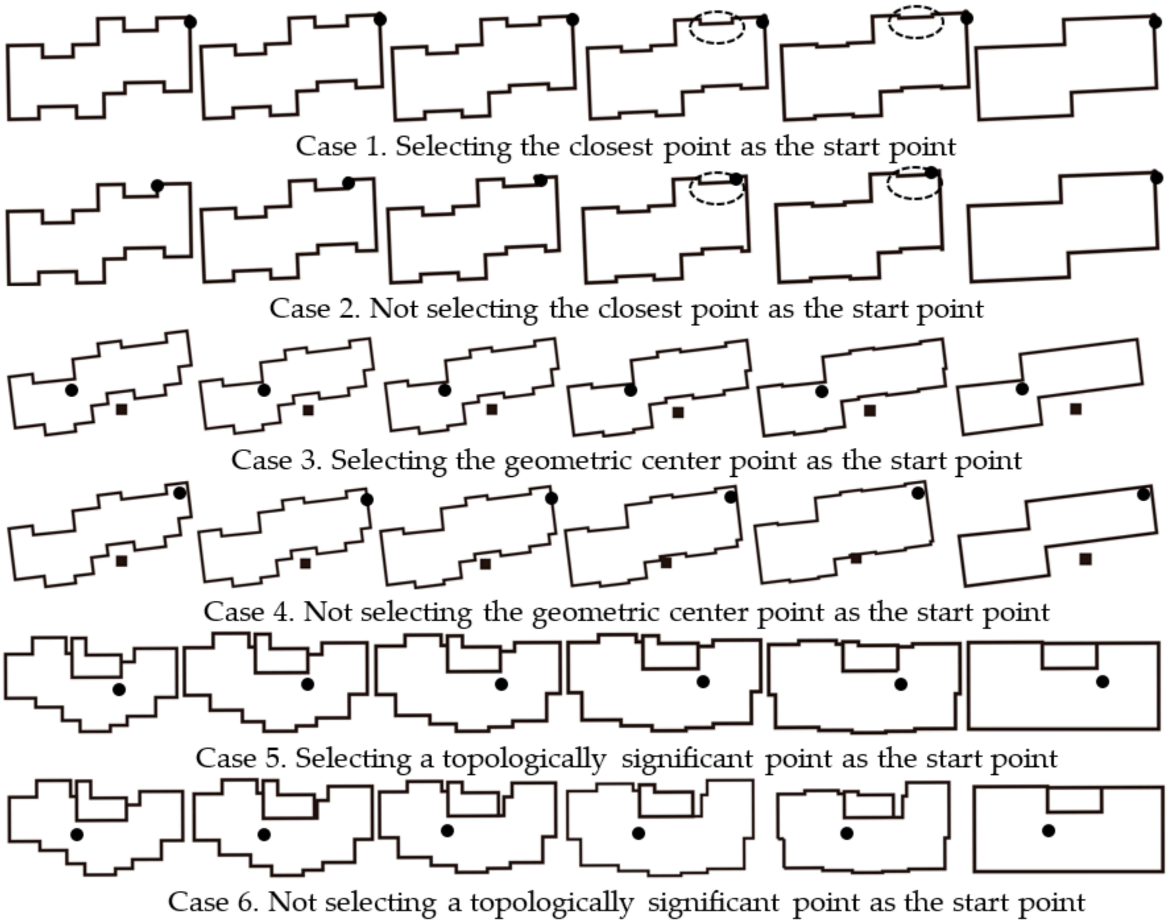
(P1) The geometric distance priority principle. We should select the point with the shortest distance between the original shape and the final shape as the start point.

(P2) The spatial position priority principle. We should select the point closest to the center of the shape as the start point.

(P3) The topological relationships priority principle. We should select a point with topological significance as the start point.

(P4) From P1 to P3, the priority increases. Generally, the topological relationship is more important than the spatial position, and the spatial position is more important than the geometric distance. Thus, principle 3 is given the highest priority, principle 2 has a medium priority, and principle 1 has the lowest priority.

Figure 7 is a set of contrast tests with different start points. In case 1, we select the closest point as the start point. We can observe that the slight concavity in the circle area remains smooth and continuously changes until it disappears. In case 2, the start point at the large scale is not the closest point. In this case, the slight concavity in the circle area prematurely disappears and results in an uncoordinated shape during the interpolation. In cases 3 and 4, the small square is an anchor point that does not change during the interpolation. We select the point near the geometric center and the closest point as the start points in cases 3 and 4, respectively. Our comparison shows that during the entire interpolation, the intermediate states in case 3 preserve their geometric positions well, whereas in case 4, their geometric positions drift relative to the anchor. In cases 5 and 6, we select a topologically significant point and a point near the geometric center as the start points, respectively. The former preserves the topological relationship during the entire interpolation, but in case 6, the topological relationship is invalid at some scales. From this case study, we observe that principles P1–P4 are feasible for the choice of a start point.


Case 1. Selecting the closest point as the start point

Case 2. Not selecting the closest point as the start point

Case 3. Selecting the geometric center point as the start point

Case 4. Not selecting the geometric center point as the start point

Case 5. Selecting a topologically significant point as the start point

Case 6. Not selecting a topologically significant point as the start point

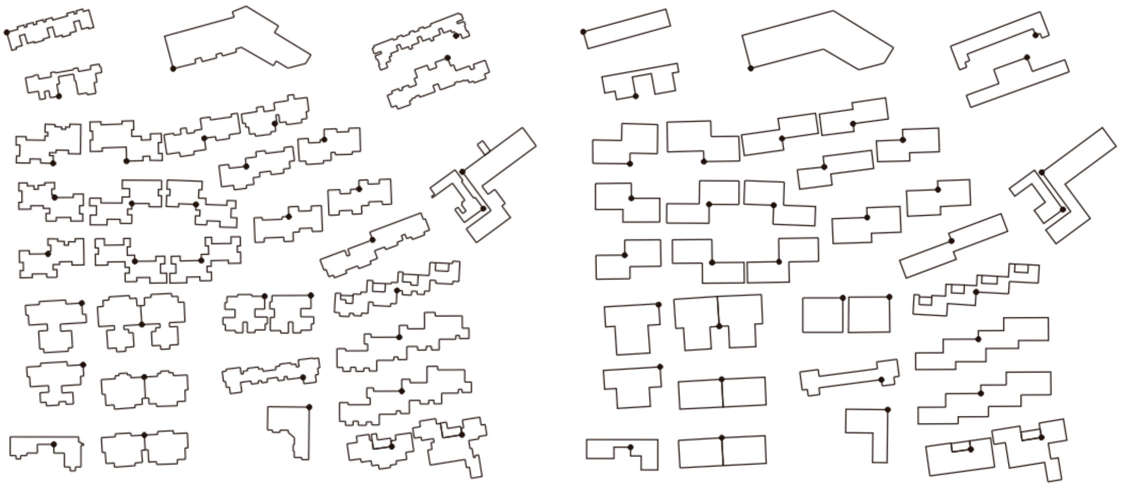
Figure 7. Morphing based on different start points.

4.2. Application of TAFBM for the Continuous Generalization of Building Features

The experimental data for continuous generalization include a pair of building datasets at the scales of T_a = 1:10,000 and T_b = 1:50,000 from the Shenzhen residential map of China. The two initial datasets are shown in Figure 8 and reflect the typical building patterns in China. Based on principles P1–P4, the start points are selected and marked with black circles. To apply the TAFBM model for continuous generalization, we must identify the particular form of the function in Equation (1). Here, we assume that parameters g and T_{Mid} are linearly dependent. Considering the constraints in Equation (2), Equation (1) becomes the following expression.

$$g = -\frac{\frac{1}{T_{Mid}} - \frac{1}{T_a}}{\frac{1}{T_b} - \frac{1}{T_a}} + 1 = -\frac{\frac{1}{T_{Mid}} - 10000}{40000} + 1 \quad (14)$$

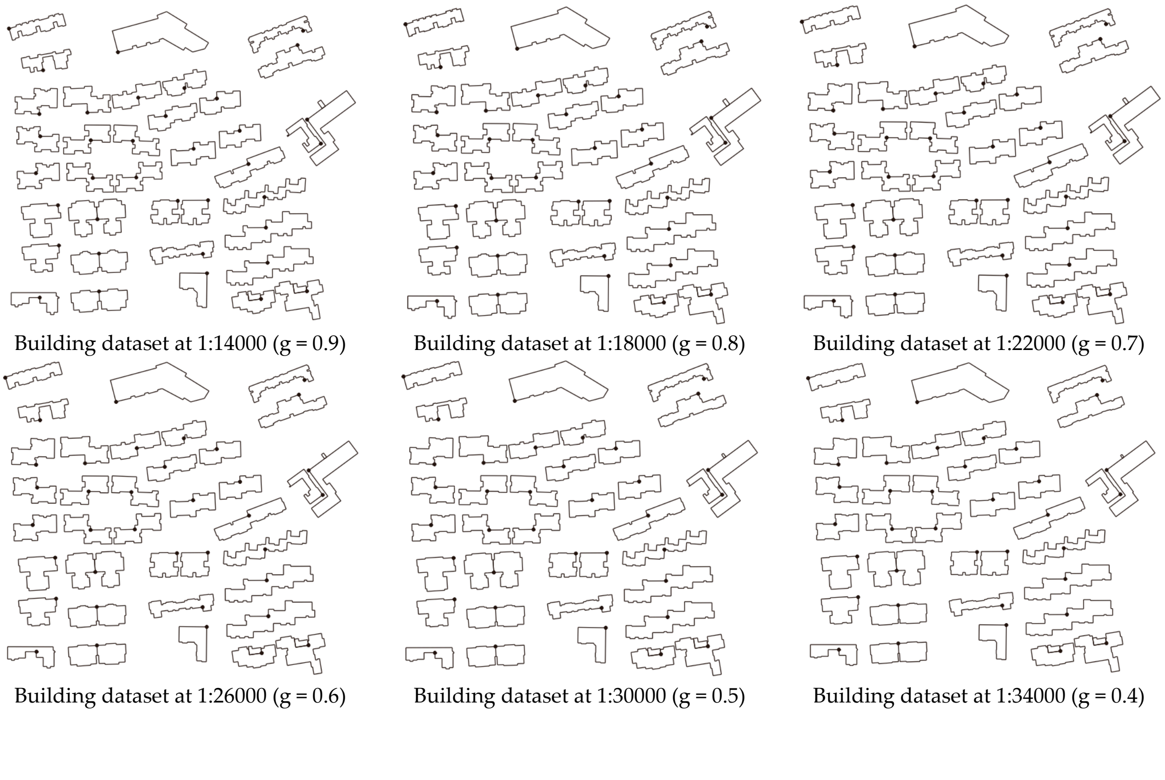
During the continuous generalization, parameter g is set to 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, and 0.1. Using Equation (14), we can obtain the corresponding scales of 1:14,000, 1:18,000, 1:22,000, 1:26,000, 1:30,000, 1:34,000, 1:38,000, 1:42,000, and 1:46,000. The results of continuous generalization are shown in Figure 9, where each single building plan smoothly and continuously changes, and the main geometric characteristics are preserved at every intermediate scale. Additionally, the relationship between the spatial direction and topology is well preserved.



Building dataset at the scale of 1:10000 (g = 1)

Building dataset at the scale of 1:50000 (g = 0)

Figure 8. Beginning and end states of the experimental data.



Building dataset at 1:14000 (g = 0.9)

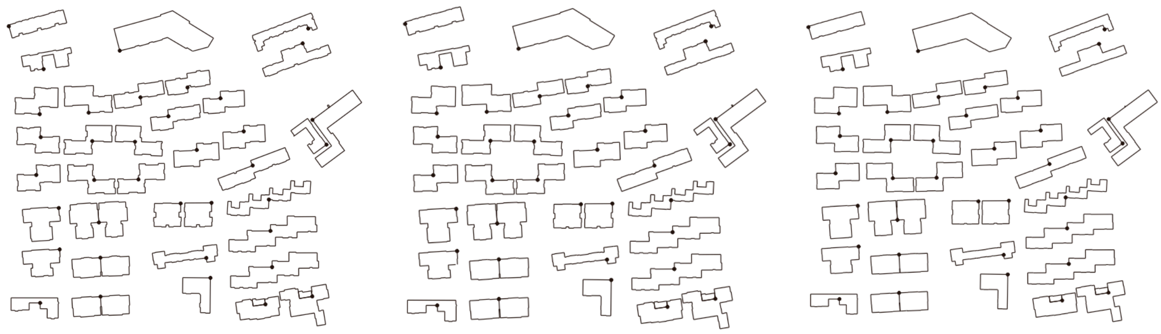
Building dataset at 1:18000 (g = 0.8)

Building dataset at 1:22000 (g = 0.7)

Building dataset at 1:26000 (g = 0.6)

Building dataset at 1:30000 (g = 0.5)

Building dataset at 1:34000 (g = 0.4)



Building dataset at 1:38000 (g = 0.3) Building dataset at 1:42000 (g = 0.2) Building dataset at 1:46000 (g = 0.1)

Figure 9. Continuous generalization of a building map using TAFBM.

For two polygons, a at 1:10,000 and b at 1:50,000, we can evaluate the effectiveness of scaling Mid to TMid using the turning function distance (TFD), which was proposed by [29]based on the Lp distance between the turning functions of two polygons. The value of TFD varies from 0–1. According to this method, if the TFD between the two polygons is small, their shapes are similar. We can define the similarity between a and Mid and b and Mid as follows.

$$\mathrm{a_{degree}} = 1 - \mathrm{TFD_a} \tag{15}$$

$$b_{degree} = 1 - \mathrm{TFD_b} \tag{16}$$

where TFDa denotes the TFD between a and Mid and $\mathrm{TFD_b}$ denotes the TFD between b and Mid.

Table 2 shows the similarity for intermediate scales of 1:10,000 and 1:50,000. According to the calculation results in Table 2, a large value of g or $\mathrm{T_{Mid}}$ corresponds to a high similarity between Mid and a (a_{degree}) and a low similarity between Mid and b ($\mathrm{b_{degree}}$). Thus, parameter g is significantly correlated with the scale.

Table 2. Similarity of shape at the intermediate scale based on the beginning and end states of a polygon.

g	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1/T$_{\mathbf{Mid}}$	46,000	42,000	38,000	34,000	30,000	26,000	22,000	18,000	14,000
a$_{\mathbf{degree}}$	0.312	0.398	0.477	0.541	0.614	0.673	0.746	0.851	0.938
b$_{\mathbf{degree}}$	0.946	0.899	0.826	0.758	0.684	0.611	0.529	0.438	0.356

We know that the TAFBM includes three steps: transformation of the vector data into the turning angle function, feature matching, and trajectory interpolation at the intermediate scale. Assuming that n polygons are involved in the transformation, the average number of vertices per polygon is m, and according to the principle of the TAFBM, the time complexity of each step is O(nm). The vertex number of building footprint m is generally small. Thus, the time consumption of our model is closely related to the number of polygons n. Table 3 shows the temporal statistics of the experiment illustrated in Figure 9, where the TAFBM effectively and efficiently provides building footprint morphing and continuous generalization. In general, our interpolation algorithm can generate approximately 10 intermediate building footprints of moderate complexity per second.

Table 3. Performance statistics of the experiment illustrated in Figure 9 (the run time is measured in seconds; n = 46; m = 15)

	Turning Function	Matching	Interpolation	Total
Run time	1.263	2.387	1.334	4.984

5. Conclusions

The morphing of building features based on a turning angle function is presented in this study. There are three main steps in the TAFBM: the conversion of vector data to the turning angle function, feature matching, and trajectory interpolation at the intermediate scale. In the process of conversion from the vector data to the turning angle function, the selection of the start point is crucial, and the principles P1–P4 proposed in this study are proven to be feasible. During the feature-matching process, the turning angle in the frequency domain and geometric characteristics in the spatial domain should be considered. In the trajectory interpolation process, the linear correlation model of g and an intermediate scale are sufficient. The case study shows that the TAFBM is suitable for the continuous generalization of building features with orthogonal characteristics. During progressive generalization, the geometric characteristics and their relationship to the spatial direction and topology are well preserved. Furthermore, the TAFBM is suitable for morphing in one-to-one mapping situations, but it only considers the generalization of a building ground plan. If the scale significantly changes, the model must include shape simplification, deletion, dimension reduction, aggregation, typification, and semantic fusion generalization operations. In the future, we will further develop the morphing method for the continuous generalization of residential features by considering changes in shape detail, geometric dimensions, mapping cardinality, and semantic conversion.

Acknowledgments: This research was partly supported by the China Scholarship Council (CSC) and a research grant from the National Natural Science Foundation of China (Grant No. 41671448).

Author Contributions: Li Jingzhong and Li Xingong conceived and designed the experiments; Xie Tian performed the experiments; Li Jingzhong and Xie Tian analyzed the data; Li Xingong contributed to the analytical tools; and Li Jingzhong wrote the paper.

Conflicts of Interest: We declare that we have no conflicts of interest.

References

1. Danciger, J.; Devadoss, S.L.; Mugno, J.; Sheehy, D.; Ward, R. Shape deformation in continuous map generalization. *Geoinformatica* **2009**, *13*, 203–221.
2. Sester, M.; Brenner, C. Continuous generalization for visualization on small mobile devices. In Proceedings of the 11th International Symposium on Spatial Data Handling (SDH'04), Leicester, UK, 23–25 August 2004; pp. 355–368.
3. Van Oosterom, P. Variable-scale Topological Data Structures Suitable for Progressive Data Transfer: The GAP-face Tree and GAP-edge Forest. *Cartogr. Geogr. Inf. Sci.* **2005**, *32*, 331–346.
4. Jones, C.B.; Ware, J.M. Map generalization in the web age. *Int. J. Geogr. Inf. Sci.* **2005**, *19*, 859–870.
5. Ai, T.H.; Li, Z.L.; Liu, Y.L. Progressive Transmission of Vector Data Based on Changes Accumulation Model. *Dev. Spat. Data Handl.* **2005**, 85–96, doi:10.1007/3-540-26772-7_7.
6. Nöellenburg, M.; Merrick, D.; Wolff, A.; Benkert, M. Morphing polylines: A step towards continuous generalization. *Comput. Environ. Urban Syst.* **2008**, *32*, 248–260.
7. Deng, M.; Peng, D. Morphing Linear Features Based on Their Entire Structures. *Trans. GIS* **2015**, *19*, 653–677.
8. Gomes, J.; Darsa, L.; Costa, B.; Velho, L. *Warping and Morphing of Graphical Objects*; Morgan Kaufman: San Francisco, CA, USA, 1999.
9. Heinz, F.; Güting, R.H. Robust high-quality interpolation of regions to moving regions. *GeoInformatica* **2015**, *20*, 385–413.
10. Reilly, D.; Inkpen, K. Map morphing: Making sense of incongruent maps. In Proceedings of the Graphics Interface 2004, London, ON, Canada, 15–19 May 2004; pp. 231–238.
11. Li, Z.L.; Wong, M. Animating basic operations for digital map generalization with morphing techniques. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, Beijing, China, 3–11 July 2008.
12. Pantazis, D.; Karathanasis, B.; Kassoli, M.; Koukofikis, A. Are the morphing techniques useful for cartographic generalization? In *Urban and Regional Data Management*; Krek, A., Rumor, M., Zlatanova, S., Fendel, E.M., Eds.; CRC Press: London, UK, 2009; pp. 195–204.
13. Pantazis, D.; Karathanasis, B.; Kassoli, M.; Koukofikis, A.; Stratakis, P. Morphing techniques: Towards new methods for raster based cartographic generalization. In Proceedings the Twenty-fourth International Cartography Conference, Santiago, Chile, 15–21 November 2009; pp. 15–21.
14. Efrat, A.; Har-Peled, S.; Guibas, L.J.; Murali, T.M. Morphing between polylines. In Proceedings of the 12th ACM-SIAM Symposium Discrete Algorithms, Washington, DC, USA, 7–9 January 2001; pp. 680–689.

15. Van Kreveld, M. Smooth generalization for continuous zooming. In Proceedings of the 20th International Cartographic Conference (ICC'01), Beijing, China, 6–10 August 2001; pp. 2180–2185.
16. Hughes, J.F. Scheduled Fourier Volume Morphing. In Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, 1 July 1992; pp. 43–46.
17. He, T.; Wang, S.; Kaufman, A. Wavelet-Based Volume Morphing. In Proceedings of the IEEE Visualization, Washington, DC, USA, 21 October 1994; pp. 85–92.
18. Yang, W.Y.; Feng, J.Q. 2D Shape Morphing via Automatic Feature Matching and Hierarchical Interpolation. *Comput. Graph.* **2009**, *33*, 414–423.
19. Sederberg, T.; Greenwood, E. A physically based approach to 2D shape blending. *ACM Comput. Graph.* **1992**, *26*, 25–34.
20. Van, O.R.; Veltkamp, R.C. Parametric search made practical. *Comput. Geom.* **2004**, *28*, 75–88.
21. Ai, T.H. The drainage network extraction from contour lines for contour line generalization. *ISPRS J. Photogramm. Remote Sens.* **2007**, *62*, 93–103.
22. Sederberg, T.; Gao, P.; Wang, G.; Mu, H. 2-D shape blending: An intrinsic solution to the vertex path problem. *Comput. Graph.* **1993**, *27*, 15–18.
23. Alexa, M.; Cohen-Or, D.; Levin, D. As-rigid-as-possible shape interpolation. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 23–28 July 2000; pp. 157–164.
24. Surazhsky, V.; Gotsman, C. Intrinsic morphing of compatible triangulations. *Int. J. Shape Modell.* **2003**, *9*, 191–201.
25. Erten, C.; Kobourov, S.G.; Pitta, C. Intersection-free morphing of planar graphs. In Proceedings of the 11th International Symposium on Graph Drawing, Perugia, Italy, 21–24 September 2003.
26. Cecconi, A.; Galanda, M. Adaptive zooming in Web cartography. *Comput. Graph. Forum* **2002**, *21*, 787–799.
27. Whited, B.; Rossignac, J. B-morphs between b-compatible curves in the plane. In Proceedings of the ACM/SIAM Joint Conference on Geometric and Physical Modeling, New York, NY, USA, 12 January 2009; pp. 187–198.
28. Chazal, F.; Lieutier, A.; Rossignac, J.; Whited, B. *Ball-Map: Homeomorphism between Compatible Surfaces*; Georgia Institute of Technology: Atlanta, GA, USA, 2006.
29. Arkin, E.M.; Chew, L.P.; Huttenlocher, D.P.; Kedem, K.; Mitchell, J.S.B. An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 209–216.