Simulation of realistic digitizing errors on geographical objects using constraints modeling the operator input process

Jean-François Girres[*1]
[1] Université Paul-Valéry Montpellier 3 – UMR GRED, France

*Corresponding author: jean-francois.girres@univ-montp3.fr

The geometry of vector objects in a geographical database is mainly obtained through manual input processes performed by an operator. It is well known that this capture process can cause errors - called digitizing errors - that affect the positioning of geographical objects or the computation of geometric measurements (i.e. length and area). To model these impacts, simulation methods, based on Monte-Carlo approaches, are generally proposed. However, several problems can occur by simulating digitizing errors with simple random processes (e.g. topological errors or non-respect of the object's shapes). Indeed, these problems would have been avoided by an operator performing a classical manual input process. Thus, in order to simulate digitizing error with a higher degree of realism, it seems important to understand and model some operator's behaviors. In this context, this paper proposes an approach to simulate digitizing errors using a set of constraints that try to reproduce the capture mechanisms of the operator. To contextualize these issues, principles and limitations of digitizing error simulation methods are presented in section 1, before detailing in section 2 the proposed constraints to model the operator input process. Finally, an experiment of the proposed methods is performed in section 3, before concluding.

I SIMULATION OF DIGITIZING ERROR : PRINCIPLES AND LIMITATIONS

The geometry of geographical vector objects is mainly produced through manual input process performed by an operator. This human capture process can generate accidental and systematic errors, called digitizing errors, that affect the positioning of the vertices of the geometry. These errors impact the positioning of objects, the computation of geometric measurements (length or area) or any analysis performed with the geometries of vector objects (e.g. buffers, overlaps). Thus, to model the positional uncertainty involved by digitizing error and assess its impact, several contributions (Keefer et al, 1988; Hunter and Goodchild, 1996) propose to use simulation methods, based on Monte-Carlo approaches. Using these approaches, simulated geometries are created by generating random errors in the positioning of each vertex of the geometry of the original objects. As proposed by Bolstad et al (1990), a normal law can be used to model the imprecision in the positioning of the vertices.

The simulation of errors on a geometry can be seen as the translation of the coordinates x and y of each vertex. As exposed on figure 1, two methods can be used to translate the vertices in a simulated geometry:

- in the first method (on the left), the coordinates of a vertex *P* are translated according to an angle *a* (generated randomly), at a distance *d* from the original vertex position (*d* is generated using a normal law)
- in the second method (on the right), the coordinates *x* and *y* of a vertex *P* are translated at distances *dx* and *dy* from the original vertex position

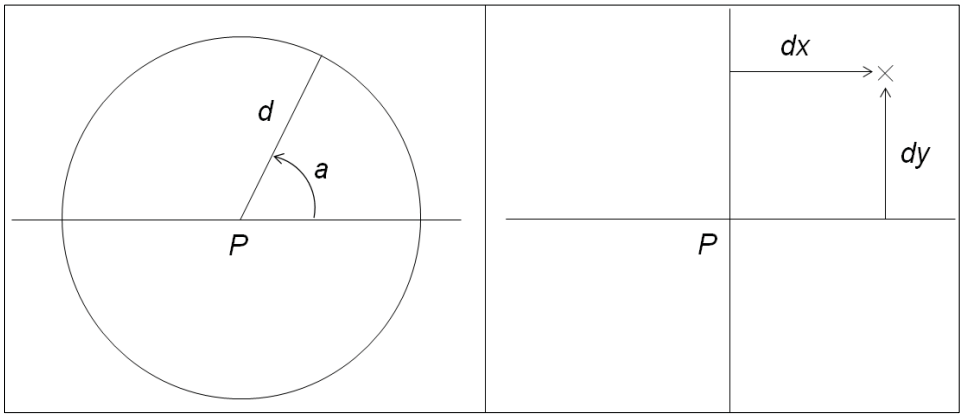

Figure 1: Two methods to translate the coordinates x and y of a vertex P

The first method, which is considered as more correct to represent accidental errors (Vauglin, 1997) is used in this study to simulate digitizing errors on a geometry. This operation is performed on each vertex of the original geometry in order to generate a simulated geometry. By repeating this operation on a large set of realizations (figure 2), it is then possible to study the sensibility of geometric measurements on simulated geometries (by comparing lengths or areas with the original geometry) and then assess the impact of digitizing error on measurements, as proposed by Goodchild et al (1999).

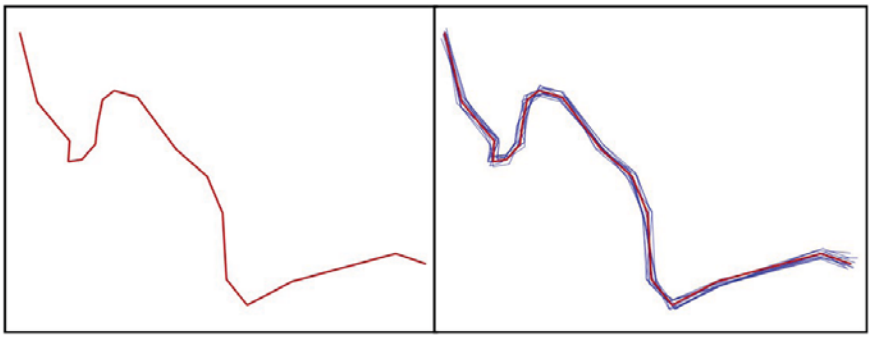

Figure 2: An original geometry (left) and 10 simulated geometries (right)

However, several problems, which would have been avoided by a "human" operator, can occur by simulating digitizing errors with simple random processes. The first type of problem is related to topological errors (e.g. self-intersection of edges) as exposed in figure 3. Several contributions have already been proposed to avoid such topological inconsistencies (Hunter and Goodchild, 1996; Bonin, 2002).

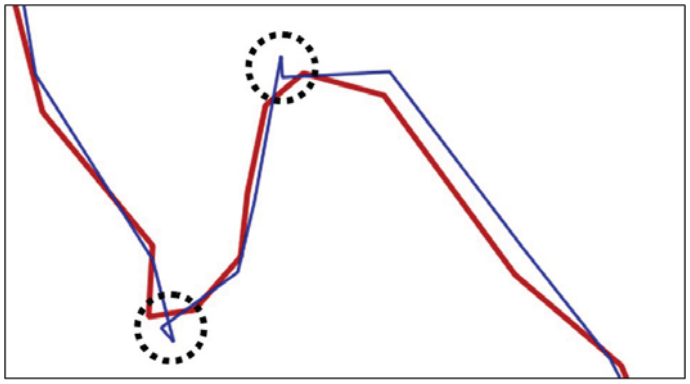

Figure 3: Topological inconsistencies on a simulated geometry (in blue)

The second type of problem deals with the non-respect of the original shape of the object after a simulation. For instance, as exposed in figure 4, the simulation of random digitizing errors on an original polyline with a straight shape (in red) can generate a distorted simulated geometry (in blue). As a consequence, the simulated polyline loses its original straight shape, and its length automatically increases (because a straight line minimizes the length). Thus, this simulation can be considered as unrealistic, because a human operator would have preserve the straight shape of the polyline.

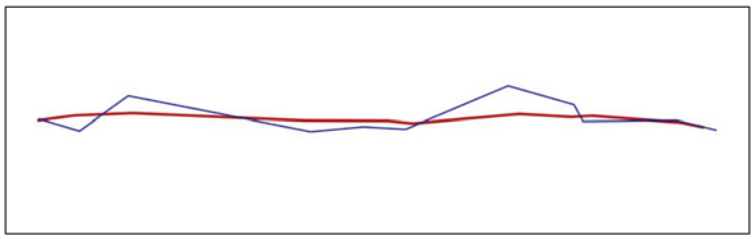

Figure 4: Non-respect of the original shape of a straight line using random simulations

These two examples illustrate a simple fact : the use of simple random models is not adapted to simulate human input processes. Indeed, the operator input process involves mobilizing cognitive mechanisms that are complex to model. So, in order to simulate digitizing errors with a higher degree of realism (and assess their impact on measurements), some operator's behaviors need to be understood and modeled in the simulation process. The next section of this paper will present propositions in order to model the operator input process.

II SIMULATION CONSTRAINTS TO MODEL THE OPERATOR INPUT PROCESS

In order to simulate digitizing errors with more realism, this section presents a set of constraints that try to reproduce the capture mechanisms of the operator. Based on basic assumptions about the data entry process, three types of constraints are integrated in the digitizing error simulation model in order to preserve the realism of the generated objects : the weighting of errors on vertices according to their angularity (1), the weighting of errors on nodes according to their degree (2) and the correlation of errors on vertices using progressive translation from the extremities (3).

2.1 Weighting of errors on vertices according to their angularity

The first type of constraint on the simulation process deals with the weighting of errors on vertices according to their angularity. This constraint is integrated in order to avoid the distortion of straight lines, as exposed in figure 4, that a human operator would have preserved.

The basic underlying assumption is that the angle between successive vertices of a geometry is a representation of the reality determined with the consciousness of the operator. So, if successive vertices of a geometry are aligned, a realistic error simulation should preserve this alignment. On the other hand, if brutal turns in the direction between successive vertices are observed, the risk of digitizing error might be more important. This basic assumption involves to take into account the angularity between successive vertices as a constraint on the simulation of digitizing error.

To model this constraint, a weighting q (between 0 and 1) of the distance d of the error on each vertex P of the geometry is determined according to the angle between successive vertices $P\text{-}1$, P and $P+1$. For instance, A, B and C are three successive vertices. To determine the weighting q_B of the error on the vertex B, its angularity α_B (in radian) is computed using equation 1.

$$\alpha_B = \frac{\left|\widehat{ABC} - \pi\right|}{\pi} \qquad (1)$$

Once the angularity of each vertex is determined, a function can be used in order to determine the weighting q according to the angularity α. Indeed, based on our assumption, the more the angle is close to π , the more the error should be small. But it does not mean that the error is null (which would be the case with $\alpha = 0$). Moreover, it is extremely rare to get angles between successive vertices where the angularity α_B is close to 0 or 2π.

As a consequence a function need to be defined in order to determine the weighting q of the error according to the angularity α. This function can take basic forms (e.g. $q=0.75\alpha+0.25$; $q=\alpha^{0.5}$) or more complex formulations, but the goal of this paper is not to define the best function to determine the appropriate weighting according to a given angularity. In the experiments (section 3), a basic form will be used.

Once the function is defined, the weighting of digitizing errors can be applied on each vertex of the geometry using a normal law $N(0, \sigma)$. As exposed on figure 4, which represents the simulation of errors using a weighting according to the angularity, the more the angularity is important, the more the the error on the vertex is important.

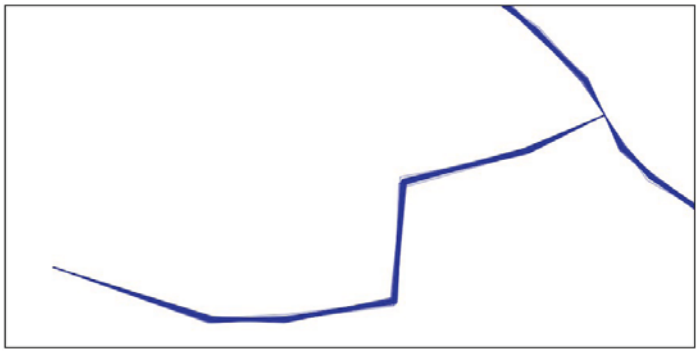

Figure 4: Simulation of errors weighted by the angularity between successive vertices

The example of simulation exposed on figure 4 shows that no errors is applied on the extremities of the polyline, where it is not possible to determine any angularity value. To determine the error applied on the extremities of a polyline, a second method, based on the degrees of the extremity nodes, is exposed in the following subsection.

2.2 Weighting of errors on extremity nodes according to their degree

Because it is not possible to apply a weighting based on the angularity between successive vertices on the extremities of a polyline, a method based on the degree of extremity nodes is proposed.

The underlying assumption formulated to determine this constraint is that the more a node is connected with other edges, the easier its localization should be for the operator, and the more precise its positioning should be. On the other hand, if an extremity is not connected with any other edge, we can expect that it is more difficult for the operator to locate the node precisely. As a consequence, the positioning of unconnected extremity nodes should be more imprecise, and the digitizing error should be more important.

Following this assumption and in order to simulate digitizing errors with more realism, we consider that the error applied on the extremities of a geometry have to be weighted according to the degree of the edges. Thus, for a node N with a degree d located at the extremity of a polyline, we propose to apply a weighting q_N as defined in equation 2, using a constant a.

$$q_N = \frac{a}{d} \qquad (2)$$

As mentioned previously, the goal of this paper is not to define the most appropriate value of the constant *a* used to compute the weighting of errors on extremities. In this paper, we will use a value of *a=1*. Nevertheless, some other parameterizations of this constant could be experimented in further researches.

The weighting q_N is then applied on the simulated errors on the initial and final nodes of the geometry using a normal law *N(0, σ)*. Figure 5 illustrates the application of the weighting of errors on extremity nodes according to their degrees, associated with the weighting of errors on vertices according to their angularity. It shows that the error applied on unconnected nodes is more important than on connected nodes.

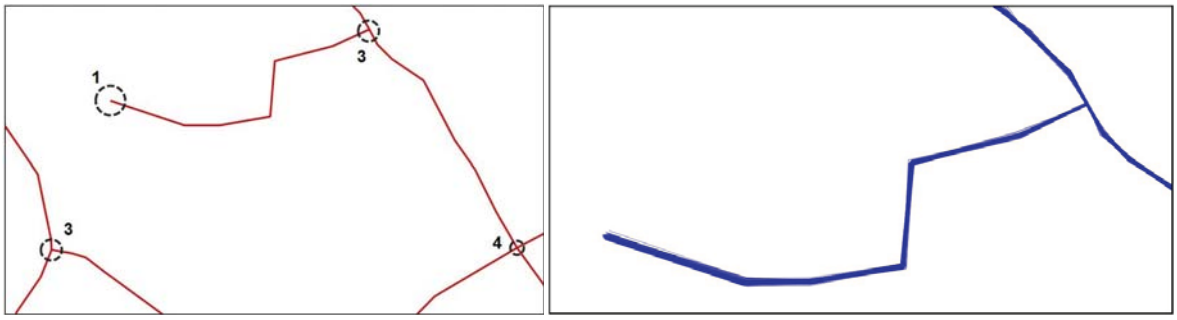

Figure 5: Determination of the degrees of nodes (left) for the weighting of errors on extremities (right)

This second constraint shows how to apply a weighting for the simulation of digitizing error on the extremities of a geometry according to its configuration. Finally, in order to take into account the dynamic dimension of the data capture process between the initial and the final node, a last constraint is proposed : the correlation of errors between successive geometries.

2.3 Correlation of errors between successive vertices

Several contributions (Heuvelink et al., 2007; De Bruin, 2008) consider that the simulation of errors on a geometry should integrate a correlation between successive vertices. It is admitted that an operator can generate a systematic error (i.e. a bias) during the input process of vector data. Nevertheless, it is difficult to simulate this error, because it depends of the operator and the way he works. But some other reasons (e.g. the dynamic dimension of the data entry process) can justify to integrate a correlation of errors between successive vertices.

To explain the integration of the correlation of errors between successive vertices, we can use the example of an operator capturing a road. If an error occurs at the extremity of the road (i.e. at the crossroad), we can assume that the operator will try to cushion progressively this error on the entire polyline, instead of correcting it brutally on a single vertex, what would generate a non-respect of the shape of the road. Indeed, the input process of vector data is a dynamic (and active) process. Thus, if the operator generates an error, he will try to correct the error without reducing his productivity. Following this example, we can consider that the error generated on the extremities (proposed in the previous subsection) is propagated progressively on the entire geometry, which supposes a correlation of errors between successive vertices.

To model the correlation of errors between successive vertices, we propose a method that generates a translation of each vertex, based on the errors on the extremities.

Using the method proposed in the precedent subsection, errors a generated at the extremities of a geometry according to the degree of the nodes, and following a normal law *N(0, σ)*. As a consequence a translation using vectors $\vec{v}$ and $\vec{u}$ is performed on the extremities *A* and *C*, and a translation is applied on an intermediate vertex *B* with a vector $\vec{w}$, computed with a weighted sum of vectors $\vec{v}$ and $\vec{u}$.

The weighting coefficients a and b are defined as the ratio between the length from the intermediary vertex to each extremity node with the total length of the polyline, as exposed on equation 3.

$$\vec{w} = a * \vec{v} + b * \vec{u} \quad (3)$$

where a is the ratio between the length of [AB] and the length of [AC] and b is the ratio between the length of [BC] and the length of [AC]

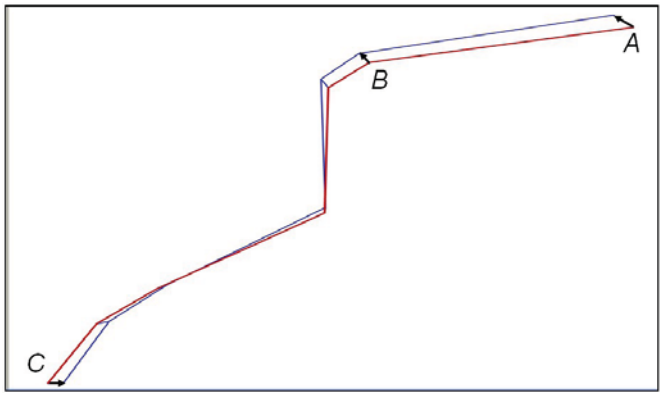

Figure 6: Translation of a vertex *B* according to the errors on the extremities *A* and *C*

It is important to notice that in figure 6, the error on intermediate vertices (using the angularity between successive vertices, as proposed on subsection 2.1) is not applied.

2.4 Combination of constraints for digitizing error simulation

The combination of the three constraints varies according to the type of geometries and their characteristics. For polylines, the three different constraints proposed to generate realistic digitizing error simulations are combined for each realization of the simulation as follows: (1) error simulation on the extremity nodes according to their degree, (2) translation of intermediary vertices according to extremity nodes errors (i.e. correlation of errors), (3) error simulation on translated vertices according to their angularity.

For polygons, it is necessary to take into account the characteristics of the represented objects. We propose to differentiate the objects representing a partition of the real world (e.g. administrative units) and objects that don't represent a partition of the real world (e.g. water surfaces). For polygon objects representing a partition of the real world, the same methodology as the one proposed for polylines is applied. Indeed, these objects suppose the existence of common borders. We can then suppose that a correlation between successive vertices exist, because the operator will start to capture a border from an initial node (the border between three units) to a final node. As a consequence, he will cushion the error from the extremities on the entire border. Thus, these polygon objects can be simplified as a graph.

For polygon objects which don't represent a partition of the real world, only the method of error simulation on vertices according to their angularity is applied. Indeed, because these geometries are closed and isolated, the notion of initial and final nodes doesn't exist. As a consequence, the method proposed to generate errors on extremities (according to the degree of the nodes) and translation between successive vertices (according to errors on extremities) don't need to be applied.

The different methods proposed in this section are experimented in the following section on real objects.

III EXPERIMENTS

To illustrate the functioning of the methods proposed to simulate digitizing errors, an experiment is proposed on a dataset of administrative units, extracted from the French BDCARTO database in the area of Handaye and Saint-Jean-de-Luz (south-west of France). The total area of these administrative units is about 133.43 sq. km.

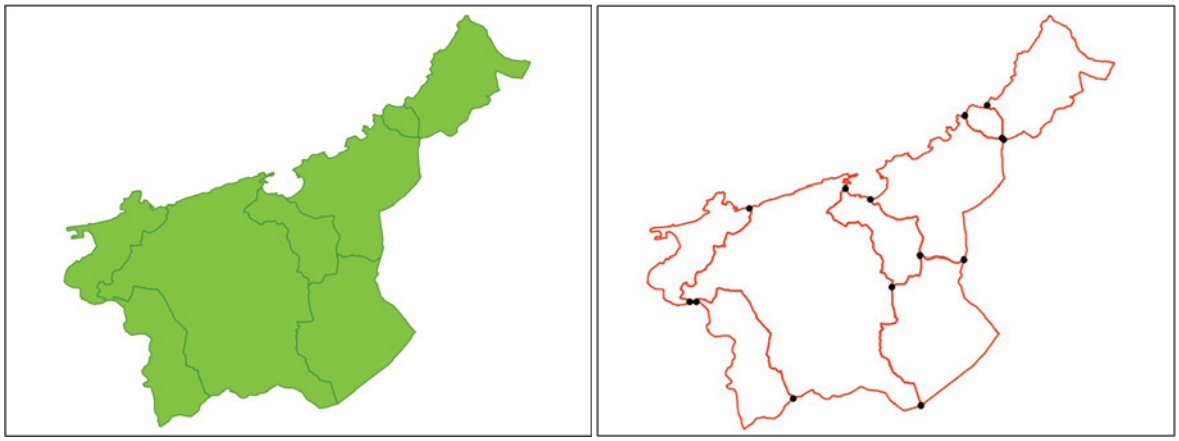

Figure 7: The administrative units used for the experiment (left) and their conversion in a topological graph (right, with edges in red and nodes in black)

To simulate digitizing errors, a normal law $N(0, \sigma)$ is used, with a value σ = 4.97 m. (determined using the automatic capture scale estimation method, see Girres, 2015 and Girres, 2012) and a function $q=0.75\alpha+0.25$ for the weighting of errors according to the angularity between successive vertices. Finally, 1000 realizations of the simulation are realized. To perform the simulation, administrative units, which are considered as a partition of the real world, are converted in a topological graph (figure 7). This method allows to extract nodes and to process each border between units as a polyline during the simulation. Finally, after the simulation, polygon units are re-built using the graph faces.

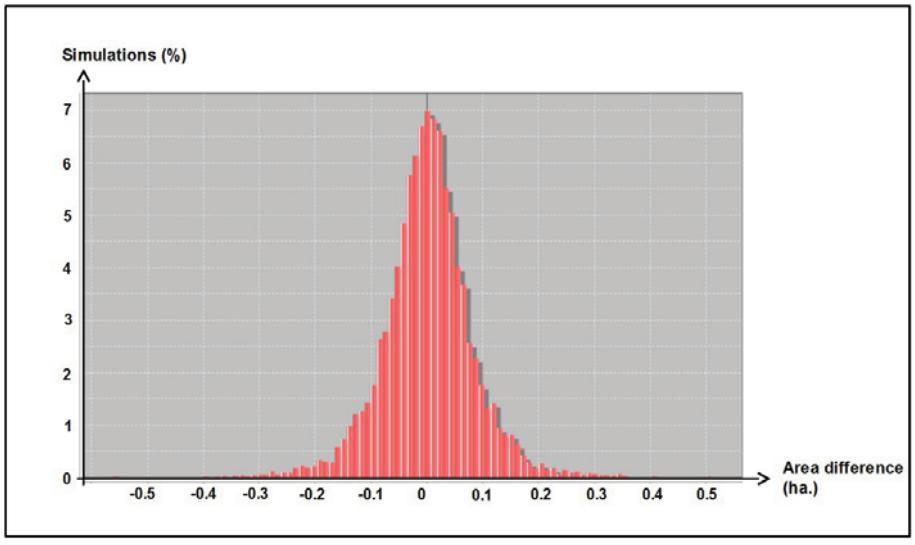

Figure 8: Distribution of area differences between original and simulated administrative units

Results of the experiments show that the distribution of area differences between the original dataset and the simulated datasets presents a value of σ_A = 0.008 sq. km. (= 0.8 ha). Assuming that this distribution follows a normal law, we can consider that the area imprecision is about +/- $3\sigma_A$ (with a confidence of 99%) which means an area imprecision of +/- 2.4 ha (0.018% of the original area).

As a comparison, the same experiment performed with no constraints on the simulation of digitizing errors would have generated an imprecision of about +/- 4.8 ha (or 0.036% of the original area), which means an area imprecision multiplied per two. These results show that the use of a simple random digitizing error simulation process can generate important over-estimations of the measurement imprecision, and that the integration of constraints in the simulation process allows to limit such exaggerations.

CONCLUSION

As a conclusion, this paper presents original methods in order to simulate digitizing errors with a higher degree of realism, by integrating constraints modeling the operator input process. Observations show that classical random processes used to simulate digitizing errors can generate topological inconsistencies and non-respect of the original shape of objects, and these problems would have been avoided by a human operator. Based on a set of three assumptions about the operator input process, three methods are then proposed to integrate constraints in the simulation process. Results of the experiments show that the use of these constraints reduce the measurement imprecision generated by a classical random process, in addition with the suppression of potential topological inconstancies. However, several improvements need to be realized on this method, especially the question of the parameterization of error weightings (according to the angularity of successive vertices or the nodes' degrees). Further researches should be investigated in order to provide the most appropriate ways to parametrize the different constraints. In spite of these implementation problems, this approach proposes new materials in order to model digitizing error with more realism, and assess its impact on the positioning of geographical objects or on geometric measurements.

References

Bolstad, P. V., Gessler, P. and Lillesand, T. M. (1990). Positional uncertainty in manually digitized map data. *International Journal of Geographical Information Systems 4*(4), 399-412.

Bonin, O. (2002). *Modèles d'erreurs dans une base de données géographiques et grandes déviations pour des sommes pondérées; application à l'estimation d'erreurs sur un temps de parcours*. PhD Thesis, Paris 6 University, France.

De Bruin, S. (2008). Modelling positional uncertainty of line features by accounting for stochastic deviations from straight line segments. *Transactions in GIS 12*(2), 165-177

Girres, J.-F. (2012). *Modèle d'estimation de l'imprécision des mesures géométriques de données géographiques. Application aux mesures de longueur et de surface*, PhD Thesis, Paris-Est University, France.

Girres, J.-F. (2015). Estimation of geographical databases capture scale based on inter-vertices distances exploration, *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences 2(3)*, 305-310.

Goodchild, M., Shortridge, A. and Fohl P. (1999). Encapsulating simulation models with geospatial data sets. In Lowell, K. and Jaton, A. (eds) : *Spatial Accuracy Assessment : Land Information Uncertainty in Natural Resources*, Ann Arbor Press, 123-129.

Heuvelink, G. B. M., Brown, J. D. et van Loon, E. E. (2007). A probabilistic framework for representing and simulating uncertain environmental variables. *International Journal of Geographical Systems 21*, 497-513

Hunter, G. J. et Goodchild, M. F. (1996). A new model for handling vector data uncertainty in geographic information systems. *URISA Journal 8*(1), 51-57.

Keefer, B., Smith, J. and Gregoire, T. (1988). Simulating manual digitizing error with statistical models. In *Proceedings of GIS/LIS'88 Conference*, 475-483.

Vauglin, F. (1997). *Modèles statistiques des imprécisions géométriques des objets géographiques linéaires*. PhD Thesis, Marne-La-Vallée University, France.