(Replaces part 3.4.1) Channel boundaries simulation method

Monte Carlo (MC) simulations as statistical methods are generally used in cases where processes are random or when assumptions in the theoretical mathematics are badly known (Brown and Duh, 2004; Openshaw et al., 1991). Applying MC simulations in this research context is the main novelty of this study. This approach has two main advantages. Firstly, MC simulations are particularly well suited to our problem because of the difficulty of distinguishing between inherent and processing errors in the measured RMSE over the whole area. Secondly, MC simulations assume a spatial continuity and a relative spatial homogeneity of the error, which is consistent with resulting spatial patterns of errors observed after the coregistration or digitising process. MC simulations are also relatively easy to perform and applicable in very different cases. This approach could thus improve the generalisation of methods for calculating planform changes and spatially variable uncertainty in a fluvial context, as suggested by Donovan et al. (2019).

The approach used in this study followed the rules of boundary simulations (Burrough et al., 2015). As described in the previous section, SV-error has been interpolated on the whole study area. Then for each node, all pixels in a 5m buffer around them were selected. These local distribution of error have then been checked and followed a normal distribution for a large part of them. A normal distribution of error was then calculated by averaging the mean local error and by calculating the standard deviation for each node, in each sub-reach. Hence, for each run (1000 runs in total), a specific value of error in x (e_x[i= 1, ...,1000]) and y (e_y[i= 1, ...,1000]) was randomly extracted from the respective normal distribution in order to shift each node from its original position.

Furthermore, in accordance with results from Podobnikar (2008), the shape of a particular channel is assumed to remain coherent after simulation. In this study, as the distance between nodes is significantly higher than the local registration error, it might be possible to move nodes of each sub-reach in any x and y directions without significantly impacting the shape. However, when that latter condition is not respected (in historical maps for instance, Herrault et al., 2013), the operation can potentially lead to strong geometrical errors such as “butterfly polygon” or excessive geometric distortions. These errors might be partially corrected (moving average algorithm, Douglas Peucker filtering) but could wrongly modify the original shape of channels. Thus, we proposed an hybrid solution to simulate shifting of nodes in space : 1) nodes from one sub-reach can move in any Y directions (i.e. positive or negative) for each run; (2) nodes from one sub-reach can only move in only one X direction for each run. That latter rule allows to avoid topological errors while simulating the most probable displacements of polygon channels. We also believe this choice is preferable to allow transferability of our method to other fluvial contexts. The direction of errors in x and y were randomly selected at each MC simulation with equal probability weights (i.e. 50 % each). A graphical illustration of the whole methodological process is available in Appendix B.

Last, as mentioned by Donovan et al. (2019), it is quite hard to distinguish between errors inherent to the coregistrating and digitising processes. For this reason, a digitising error (e_d) equal to 1 pixel was added as a reasonable constraint within the simulation process, considering the resolution of the orthophotos. This digitising error is assumed to be uniform over the entire area and does not fluctuate in different simulation runs. Only the direction in x and y has been randomly defined for each node from one sub-reach at each MC simulation. These directions may vary from one node to another for one sub-reach.

The overall mathematical expression of the simulation process can be expressed as follows:

$$x_{changed} = x_{original} + (|e_x| \times [-1;1]) + (|e_d| \times [-1;1])$$

$$y_{changed} = y_{original} + (|e_y| \times [-1;1]) + (|e_d| \times [-1;1])$$