

# Methods for Fine Registration of Cadastre <sup>1</sup>

## Graphs to Images

Roger Trias-Sanz, *Student Member, IEEE*, Marc Pierrot-Deseilligny,

Jean Louchet, and Georges Stamon, *Senior Member, IEEE*

### Abstract

We propose two algorithms to match edges in a geometrically-imprecise graph to geometrically-precise strong boundaries in an image, where the graph is meant to give an a priori partition of the image into objects. This can be used to partition an image into objects described by imprecise external data, and thus to simplify the segmentation problem. We apply them to the problem of registering cadastre data to georeferenced aerial images, thus correcting the lack of geometrical detail of the cadastre data, and the fact that cadastre data gives information of a different nature than that found in images —fiscal information as opposed to actual land use.

### I. INTRODUCTION

Partitioning an image into its constituent real-world objects is, because of the prior semantic knowledge required, and because of the interaction between the partitioning and the image interpretation, an extremely difficult task which has long been a goal of image analysis.

For some practical applications, a sufficiently good result can be obtained by using external data which indicates the location of significant objects in the image. For example, in a remote sensing context, the cadastre graph —which divides land into plots, and which provides ownership and fiscal information— can be taken as a rough approximation to a partition of land

into fields. The correspondence is, however, not exact because field edges —land use— need not follow cadastre edges —land ownership—, because several adjacent plots may contain the same crop and therefore be a single field, and because a single plot can be divided into several fields, all with the same owner.

The first two problems —geometrical imprecision, and cadastre edges without a corresponding land use edge— can be solved by a *registration* procedure, whereby the cadastre graph and a graph containing the image’s salient edges are matched. From correctly matched elements we can obtain the precise geometry corresponding to geometrically imprecise cadastre elements, thus addressing the first problem. As for the second one, if there is a cadastre edge between two plots containing the same crop, the algorithms will not be able to find in the image any match for this edge; if this is the case, we can infer that the two plots have the same crop.

Registration of cadastres to images has been viewed as a non-rigid registration problem. In non-rigid registration problems, the goal is to find the transformation or deformation within a class that best converts an image to the reference image; matching is then trivial. Transformation classes are usually a subset of  $C^2$  functions  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ . For example, Viglino and Guigues [1] match cadastre graphs to terrain edges by finding the best global transformation from one to the other among the class of polynomial transformations of a certain degree. See Cachier *et al.* [2] and Goshtasby *et al.* [3] for some reviews of nonrigid image registration methods; as they say, most of the work in this domain focuses on medical imaging. Chui and Rangarajan [4] propose a non-rigid registration method for sparse sets of points which could be used in some instances of graph matching problems.

These approaches assume that the initial misregistration is due to sensor-induced deformation, to acquisition errors, or because the relative orientation of the input data is unknown. In our case

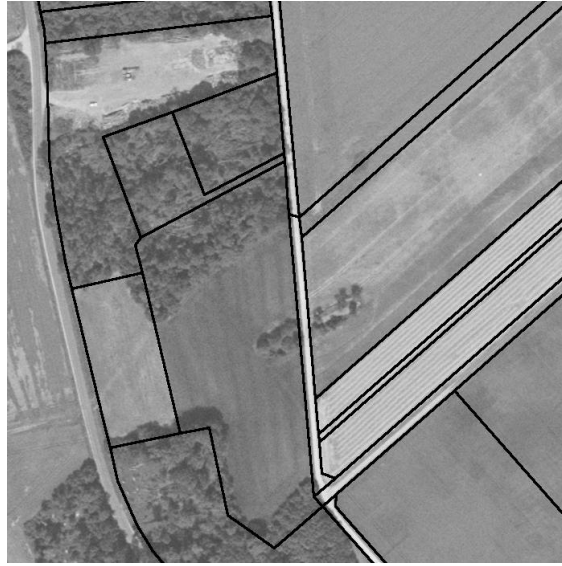


Fig. 1. Typical input data (close-up). Terrain image, with cadastre edges superimposed. Note that some image edges do not have corresponding cadastre edges, and vice versa, and that those that do have slightly different traces.

the image is georeferenced, so the third problem can be ignored, and the initial misregistration is not due to an acquisition error or a sensor-induced deformation, but rather to the two graphs representing data of different nature (see Fig. 1 for a detailed view of typical input data). We also want the cadastre to register onto image *edges* as much as possible. More precisely, we want to locally modify the cadastre graph so that its spatial structure is preserved (that is, we do not modify which faces are adjacent to which in the planar representation) while incorporating the geometric details of corresponding salient edges in the image. To the best of the authors' knowledge, this specific problem has not been previously dealt with.

We approach this as a graph matching problem, like Hivernat and Descombes [5] but with suitable modifications: in this context we have two graphs representing the same physical reality, and the goal is to match the edges and nodes that correspond to the same part of that reality [6], [7]. In our case, we use a multi-scale segmentation of the image to obtain a graph representation, whose edges separate homogeneous image regions and are weighted according

to the dissimilitude between these regions. In doing this, we assume that all significant image edges —at least, all image edges that should be matched with the given cadastre edges— exist in the multi-scale segmentation from which we obtain the graph representation; by making this segmentation sufficiently fine, we can be fairly confident that this is the case. Our experiments confirm this.

There does not exist a scale of analysis for which a single-scale segmentation would yield field edges. By explicitly working with an oversegmentation we avoid this problem. However, this also makes the matching problem asymmetrical, since to each element in the cadastre graph there correspond many elements in the image segmentation (on the order of 1 cadastre element to 20 segmentation elements). Existing graph matching algorithms, such as those described in [8] or [9], expect the graphs to be isomorphic; although provisions are made for non-isomorphisms caused by noise or other defects, usually by adding a null element or label to which non-matchable elements can be associated, these algorithms are not designed to handle such large asymmetries as we have here.

We propose two approaches for solving this matching problem:

The first approach, which was sketched in [10], finds the best match between edges in the cadastre graph and segmentation edges in the image, using simulated annealing. This is described in section III. This *edge-based* method preserves well the face structure of the cadastre graph, but does not always follow salient image edges since auxiliary straight edges must be added to the solution.

The second approach, which was first described in [11], finds a near-optimal match between faces in the cadastre graph and homogeneous regions in the image. This is described in section IV. We propose two variants, one using probabilistic relaxation for the optimization, and another

using simulated annealing. This *region-based* method always follows salient image edges, but does not preserve the face structure of the cadastre graph so well.

The best algorithm, in terms of quantitative evaluation, is the region-based method, using probabilistic relaxation for the optimization. Tests show a 32.2% reduction in the average distance between the cadastre and a ground truth (and up to 43.7% when using a distorted cadastre as input). Improvements are slightly lower with the edge-based method, and even lower with the region-based method with simulated annealing. More important than this quantitative improvement is the fact that the registered portions of the resulting graph follow the edges of the image—exactly, in the case of the region-based method—which will allow us to perform statistical analysis of whole regions, without noise from adjacent regions. Although we apply this method to the registration of a cadastre graph to an aerial image, it can also be used in other contexts. For both algorithms, a score for each cadastre edge is obtained. These registration quality measures, described in sections III-H and IV-D, can be used to determine which cadastre edges do not actually exist in the image, and may need to be removed for further processing.

In this article we present in detail the edge-based method in section III, and the region-based method in section IV, together with a complete quantitative evaluation in section V, and a comparison and conclusion in section VII. Section II describes the procedure used to obtain the initial graphs from the image and cadastre data; this procedure is common to both algorithms.

## II. IMAGE OVER-SEGMENTATION AND INPUT GRAPHS

Both registration methods presented here are set as graph matching problems. That is, they try to match edges, nodes, or faces from a graph representing the image with edges, nodes, or faces from a graph representing the cadastre. For each match, the geometry of the image element is then transferred to the corresponding cadastre element. We need therefore to start with graph

representations for both the cadastre and the image.

The cadastre input is already a graph  $C = (E_C, V_C)$ , a planar one, with graph edges separating adjacent plots, graph *faces* corresponding to plots, and graph nodes where three or more adjacent plots meet. We simplify this graph using mathematical morphology to remove very thin faces (around 1 pixel wide) that might confuse the algorithms.

We also create a graph (the *terrain* or *segmentation graph*) representing the salient edges in the image and their geometry. We need to extract the topology and geometry of the salient edges in an image, and a measure of their saliency. A simple method would be to weight each segmentation edge in a watershed segmentation with the module of the image gradient in that edge. However, this measure of edge saliency would be local and single-scale, and therefore not satisfactory, since meaningful structures may appear at different scales of analysis [12].

Several authors have proposed *multi-scale* algorithms to solve this. We use Guigues' *scale-sets* algorithm [13] because it makes the segmentation criterion and the scale parameter  $\lambda$  explicit: For each  $\lambda$  this algorithm gives a *partition* of the image  $p(\lambda)$ ; the set of values of  $\lambda$  for which a region exists in  $p(\lambda)$  turns out to be an interval,  $[\lambda_{\min}, \lambda_{\max}[$ .

We *flatten* these results to obtain the terrain graph; that is, we build a weighted graph, the *terrain graph*  $T = (V_t, E_t, w)$  ( $w$  is the weight function), whose edges follow the boundaries of the regions given by  $p(\cdot)$  (these graphs have a geometrical component: their edges and vertices exist in a space such as  $\mathbb{R}^2$  or  $\mathbb{Z}^2$ , and not only their topology but also their geometry is considered). The weight of an edge is calculated as follows: We find for each edge  $e \in E_t$  the highest  $\lambda_{\min}$  of all the regions whose boundary contains  $e$ ,  $\lambda_{\min}(e)$ . To improve processing time, we discard edges with small  $\lambda_{\min}$  (in this implementation, the 30% of edges with smallest values of  $\lambda_{\min}$ ). We sort the  $N$  remaining edges according to their values of  $\lambda_{\min}$ , and attribute each

edge with an *apparition weight*  $w$ , which is 1 for the first edge (that with highest  $\lambda_{\min}$ ) and decays exponentially for the remaining edges. The apparition weight is a multi-scale, non-local, saliency measure. Fig. 2 shows the terrain graph corresponding to the image of Fig. 1.

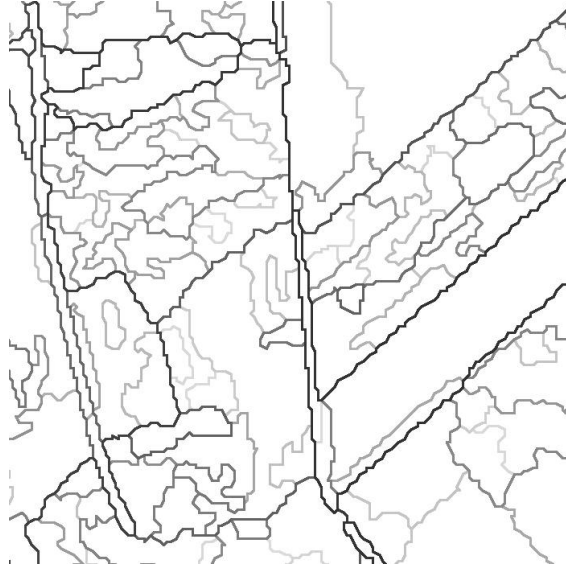


Fig. 2. Terrain graph  $T$  of the image in Fig. 2. Graph edges with higher weights are shown darker.

The region-based registration algorithm works with the duals of the terrain and cadastre graphs  $T$  and  $C$ ,  $\bar{T} = (E_{\bar{T}}, V_{\bar{T}}, w)$  and  $\bar{C} = (E_{\bar{C}}, V_{\bar{C}})$  respectively. The weight of an edge in  $\bar{T}$  that links two vertices  $\bar{v}_1$  and  $\bar{v}_2$  is that of the edge in  $T$  separating the dual faces of  $\bar{v}_1$  and  $\bar{v}_2$  (see Fig. 3).

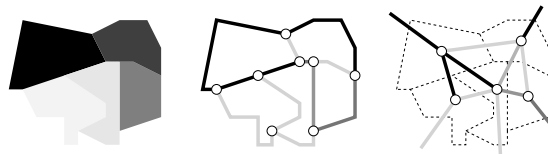


Fig. 3. An image, its terrain graph  $T$ , and its dual terrain graph  $\bar{T}$ . Darker edges have higher weights.

This graph matching is asymmetric: the terrain graph contains many more edges (*terrain edges*) than the cadastre graph, but they are shorter. Typical values are 1000 terrain edges for 50

cadastre edges. Similar ratios hold between faces of the terrain graph and the cadastre graph. For the edge-based algorithm, we may assume that each terrain edge matches at most one cadastre edge, and that each cadastre edge may match several terrain edges. Most terrain edges will not have a match. Note that while most graph-matching algorithms can handle a certain amount of unmatched edges, they are usually not designed with such a degree of asymmetry in mind. Similarly, for the region-based algorithm, we may assume that each terrain face matches at most one cadastre face, and that each cadastre face may match several terrain faces; however, in this case all terrain faces will have a match.

### III. EDGE-BASED REGISTRATION ALGORITHM

The desired result from the edge-based registration algorithm is a match from each edge in the cadastre graph  $C$  to a chain of edges in the terrain graph  $T$ , that is, to an ordered sequence of edges, such that each edge has a common vertex with the next edge in the sequence. However, internally the algorithm manipulates a different kind of solution: A match from each terrain edge to a cadastre edge.

In section III-A we formalize this latter representation of a *solution*. In section III-D we study how to convert from this to the first representation. We can evaluate the quality or *fitness* of a match (section III-E) and therefore use an optimization algorithm to find the optimal one. To avoid the combinatorial explosion associated with this kind of problem, we use simulated annealing [14] to find a near-optimal solution. For that we also need a way of exploring the solution space (section III-B), and an initial solution (section III-C).

However, we have found that, in order to register the cadastre onto image edges and at the same time preserve the spatial structure of the cadastre graph, we have to process separately the areas near cadastre nodes and the rest of the problem. This is because the cadastre and terrain



graphs are not isomorphic near cadastre nodes, in most cases, and therefore forcing registered cadastre edges to strictly follow image edges modifies in most cases the spatial structure of the cadastre graph. To solve this problem, we first apply the simulated annealing optimization to the general case, and then an ad-hoc method that adds straight edges to the terrain graph (section III-G) for the areas near cadastre nodes.

In the description of the edge-based registration algorithm,  $\alpha_1, \dots, \alpha_6$ ,  $N_r$ , and  $p_\perp$  are tuning parameters (see section V-A and table III).

#### A. Solution representation

We represent a *solution* to our registration problem as a relation between cadastre edges and terrain edges. In what we call the *backward* representation, we label each terrain edge with its corresponding cadastre edge, or with  $\perp$  to indicate that it is unmatched (see [5] for a similar representation, although they use it for Markov modeling and in a situation where the  $\perp$ -label is rarely used). We only allow labeling a terrain edge  $e_i$  with cadastre edges that are close to it,  $N(e_i)$ ,

$$N(e_i) = \{e \in E_c : \min_{\substack{z \in \text{trc } e_i \\ z' \in \text{trc } e}} \|z - z'\| < \epsilon\}, \quad (1)$$

(where the trace of a graph edge or node,  $\text{trc } x$ , is the set of pixels that correspond to that edge or node in image space) and do not allow labeling with  $\perp$  for terrain edges with only one near cadastre edge: this disables the optimization process for these edges, leaving only the shortest-path search described in section III-D. A *backward solution*  $S$  then, maps each terrain edge  $e_i$  to an element of  $N_x(e_i)$  where

$$N_x(e_i) = \begin{cases} N(e_i) \cup \{\perp\} & \text{if } |N(e_i)| \neq 1, \\ N(e_i) & \text{if } |N(e_i)| = 1, \end{cases} \quad (2)$$

and  $|X|$  is the notation for the cardinal of a set  $X$ .

### B. Finding a similar solution

In order to solve a problem using methods such as steepest-descent, simulated annealing or genetic algorithms, we need a way to find solutions which are “close” to a given solution. This is the *mutation* or *neighborhood* operator (for genetic algorithms, we also need a *crossover* operator).

In this registration algorithm, we can obtain one neighbor solution  $S'$  from  $S$  as follows: A number  $N_r$  of terrain edges are selected —however, edges which have only one or no cadastre edges nearby are never selected (Eqs. (1)–(2))— and a new random label is given to each selected edge. The new label is  $\perp$  with probability  $p_\perp$ , and the remaining labels are equiprobable:

$$\begin{aligned} p(S'(e_i) = \perp) &= p_\perp \\ p(S'(e_i) = e_k) &= \frac{1 - p_\perp}{|N(e_i)|}, \text{ for all } e_k \in N(e_i). \end{aligned} \tag{3}$$

### C. Finding an initial solution

We also need to find an initial solution  $S_0$ . We label each terrain edge with the cadastre edge that is closest to them in the sense of the average distance  $\bar{d}$  (Eq. (5))

$$S_0(e_i) = \operatorname{argmin}_{e_k \in N(e_i)} \bar{d}(e_i, e_k).$$

Other options include starting with the null solution  $S_0(e_i) = \perp$ , or with a random solution.

### D. From backward to forward representation

Using the backward representation it is easy to find solutions similar to a given one (section III-B) and to create initial solutions (section III-C). But this is not the desired output representation, and it is difficult to calculate its fitness. Therefore, we need to convert from the backward representation to a *forward* representation, in which we map each cadastre edge to a —possibly empty— oriented chain of terrain edges, as follows:

First, for each cadastre edge  $e_k \in E_c$ , to which the terrain edges  $m_k = \{e_i \in E_t : S(e_i) = e_k\}$  are mapped, we create a graph  $T_k$  which contains

- the terrain edges  $m_k$  and the nodes at their ends,
- the cadastre nodes  $n_0$  and  $n_1$  at the ends of  $e_k$ ,
- $N_0$ , the set of terrain nodes close to  $n_0$ , and
- $N_1$ , the set of terrain nodes which are close to  $n_1$ .

After that, we attribute each edge  $e \in T_k$  with a weight  $w_p$  (the *path matching weight*) that depends on its length  $\ell(e)$ , its apparition weight  $w_e$ , and an average “distance”  $\bar{d}(e, e_k)$  between it and  $e_k$ , as

$$w_p = \alpha_1 \ell(e)(1 - w_e) + \alpha_2 \ell(e) \bar{d}(e, e_k), \quad (4)$$

with the average non-symmetric “distance”  $\bar{d}$  between a terrain edge  $e_t \in E_t$  and a cadastre edge  $e_c \in E_c$  defined as

$$\bar{d}(e_t, e_c) := \frac{1}{|\text{trc}(e_t)|} \sum_{z_t \in \text{trc}(e_t)} \min_{z_c \in \text{trc}(e_c)} \|z_t - z_c\|. \quad (5)$$

In that way, we favor registered edges that follow salient image edges and that don’t stray too much from the cadastre edge they correspond to.

Note that  $T_k$  contains several connected components (some of them composed of a single node). We then create a graph  $T'_k$  which contains  $T_k$  and additional straight edges (which we call *connecting edges*), which join every two connected components in  $T_k$  by their closest nodes. We attribute each of these additional straight edges  $e$  with a path matching weight  $w_p = \alpha_3 \ell(e)$ .

Finally, using the path matching weight as the “length” of an edge, we find on  $T'_k$  the “shortest” path  $s_k$  from any node in  $N_0 \cup \{n_0\}$  to any node in  $N_1 \cup \{n_1\}$  using Dijkstra’s algorithm [15], [16]. The forward solution for  $S$  is then the mapping of each cadastre edge  $e_k$  to the shortest path  $s_k$  obtained with this process. This is shown in Fig. 4.

Thanks to the connecting edges, there is always such a path; however, the path matching weight for connecting edges is high, to discourage their use. Note that since we allow these shortest paths to start and end not only on the endpoints of cadastre edges ( $n_0, n_1$ ) but also on nearby terrain nodes ( $N_0, N_1$ ), this will, as intended, not register the cadastre in the areas near cadastre nodes.

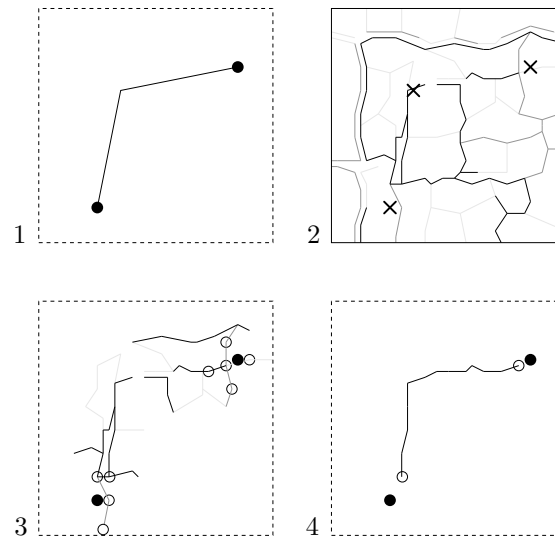


Fig. 4. From backward to forward solution, for a simplified example. 1: a cadastre edge  $e_k$ . 2: terrain graph  $E_t$  (edge weight is shown by line darkness, “x”s show the vertices of  $e_k$ ’s trace). 3: subgraph  $T_k$ , with  $N_0$  and  $N_1$  (hollow dots), and  $n_0$  and  $n_1$  (solid dots). 4: shortest path, including connecting edges. Note that this process is run at each iteration of the annealing algorithm—different backward solutions may give different  $T_k$ s and possibly better shortest paths.

There is some similarity between edge linking—the process by which a set of pixels given by an edge-detection algorithm is converted to a set of curves—and the problem that the transformation described in this section solves. Important differences warrant the development of this new method: First, only a small portion of the original edges are used in the resulting path; the original set of edges is more mesh-like than line-like. Second, edges are weighted, and it is only through their weighting that a path can be found among the many edges (but see [17] for a fuzzy pixel-wise edge linker). Third, there is the need to allow the use of “connecting

edges” in a controlled way.

### E. Calculating a solution’s fitness

The length of the shortest path calculated as in section III-D is already an indication of the quality of a registered cadastre edge. To get a suitable fitness measure for the whole solution  $s_k$ , we sum the shortest path lengths for all edges in  $s_k$  and we add three penalties, to drive down the complexity of solutions and avoid malformed solutions:

First we penalize unused edges, those not mapped to  $\perp$  which are not part of any shortest path. Also, the shortest-path weighting method values registrations for each cadastre edge independently. As a consequence, a cadastre vertex usually corresponds to several terrain vertices. To avoid this we add a second penalty, for each cadastre vertex, which depends on the number of matching terrain vertices —a penalty applies if there is more than one terrain vertex for each cadastre vertex. Finally, we penalize terrain vertices that do not correspond to a cadastre vertex and that have more than two incident terrain edges selected for shortest paths.

In all, the fitness  $f$  of a solution (lower is better) is

$$\sum_{e_k \in E_c} l_{e_k} + \alpha_4 \cdot n_u + \alpha_5 \sum_{v_j \in V_c} (|H_{v_j}| - 1) + \alpha_6 \sum_{v_k \in V_t} (|I_{v_k}| - 2) \quad (6)$$

where  $l_{e_k}$  is the length of the shortest path calculated for the cadastre graph  $e_k$  (the sum of the path matching weights of its edges),  $n_u$  is the number of unused terrain edges,  $H_{v_j}$  is the set of terrain vertices corresponding to the cadastre vertex  $v_j$ , and  $I_{v_k}$  is the set of terrain edges incident to the the terrain vertex  $v_k$ .

### F. Optimization

Once we have a representation (sections III-A and III-D) which allows us to find solutions near a given one (section III-B), to find an initial solution (section III-C), and to calculate a

solution's fitness (section III-E), we can apply standard optimization algorithms such as steepest descent, simulated annealing [14] or genetic algorithms to find the (possibly global) optimum. We have used simulated annealing.

### *G. Registration near cadastre nodes*

The optimization process alone cannot at the same time register the cadastre onto salient image edges *and* preserve the spatial structure of the cadastre graph: To preserve the spatial structure in areas of the image with a low density of image edges we need to allow the use of *connecting edges*, which are straight and do not follow salient image edges. In these same low density areas, because image edges are sparse, strictly following image edges may modify the spatial structure of the cadastre, splitting or deleting faces in the cadastre graph, for example. Additionally, adding connecting edges may make the graph non-planar. Some of the penalties included in the fitness function (section III-E) mitigate the problem, but do not eliminate it completely. The underlying reason is that it is not really possible to register cadastre edges which do not have a corresponding image edge.

These problems tend to occur at the areas near nodes of the cadastre graph. For this reason, the optimization procedure does not attempt to register the cadastre there (section III-D). After the optimization phase, we process these areas as follows to complete the registration (see Fig. 5 for a graphical example):

For a cadastre node  $n$ , let  $E = \{e_1, e_2, \dots, e_k\}$  be the set of cadastre edges incident to it. Each cadastre edge  $e_i$  is registered to a chain of terrain edges which can end at  $n$ , or at a terrain node close to  $n$ . Let's call  $n_i$  the endpoint for  $e_i$  which is close to or equal to  $n$ .

Let  $M$  be a subgraph of the terrain graph containing only edges and nodes close to  $n$ . We can find paths through  $M$  which connect some endpoints  $e_i, e_j$  together. Let's split  $E$  into subsets

$J_1, J_2, \dots, J_m$  grouping those edges whose endpoints can be connected through  $M$ . We then connect these endpoints within each  $J_i$  using these terrain edges in  $M$ . Finally, we join the sets  $J_1, \dots, J_m$ , by adding appropriate straight edges to the registered cadastre graph.

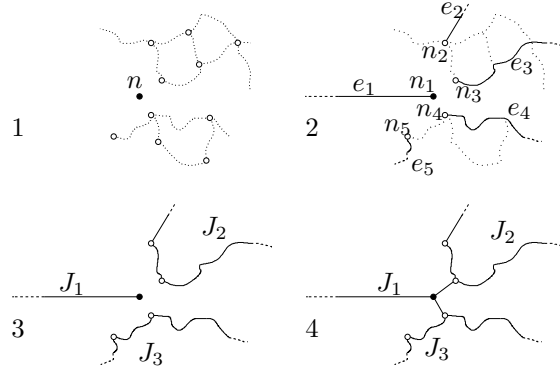


Fig. 5. Registration near cadastre node  $n$ . 1: node  $n$  and terrain subgraph  $M$ . 2: incident edges  $e_1, \dots, e_5$  and endpoints  $n_1, \dots, n_5$  ( $n_1 = n$ ). 3: connection within subsets  $J_1 = \{e_1\}$ ,  $J_2 = \{e_2, e_3\}$ , and  $J_3 = \{e_4, e_5\}$ . 4: connection between subsets  $J_1, J_2,$  and  $J_3$ .

#### H. Registration quality measure

For each registered cadastre edge  $e_k$  we calculate a quality measure, the *registration ratio*. The shortest path for  $e_k$ ,  $s_k$ , is a chain of terrain edges and connecting edges. The registration ratio is the length of the terrain edges in  $s_k$  (that is, not counting connecting edges) divided by the total length of  $s_k$  (we use the Euclidean length, “path matching weight”-induced length of section III-D). A cadastre edge which is fully registered to terrain edges has a registration ratio of 1, while that for a cadastre edge which corresponds to a single connecting edge from end to end is 0.

This appears to be a good indicator of whether or not the cadastre edge follows a true terrain limit. Since path matching weights are higher for connecting edges than for terrain edges, cadastre edges will tend to register themselves onto the terrain unless there is really no terrain edge strong

enough to follow. In section V-B we further explore this.

#### IV. REGION-BASED REGISTRATION ALGORITHM

In contrast to the edge-based registration algorithm, which attempts to match corresponding edges in the terrain and the cadastre graph, the region-based registration algorithm works by trying to match nodes in the dual terrain graph  $\bar{T} = (E_{\bar{T}}, V_{\bar{T}}, w)$ , corresponding to homogeneous terrain regions, to nodes in the dual cadastre graph  $\bar{C} = (E_{\bar{C}}, V_{\bar{C}})$ , corresponding to cadastre plots.

Since  $T$  will be obtained by *over*-segmenting the image, we can assume that each node in  $\bar{T}$  belongs to only one node in  $\bar{C}$ , and formalize a *solution* as a mapping  $s : V_{\bar{T}} \rightarrow V_{\bar{C}}$ , and use various optimization algorithms to find an optimal or near-optimal solution. We propose two variants: In Section IV-A using probabilistic relaxation for the optimization process, and in Section IV-B using simulated annealing.

In the description of the region-based registration algorithm,  $k_0, \dots, k_7, \beta, \pi_0, \pi_1, \pi_2, \pi_r$ , and  $N_r$  are tuning parameters (see section V-A and table III).  $N(a)$  is the set of neighbors of the graph node  $a$ , and  $w_{ij}$  is the weight of the edge in  $\bar{T}$  between nodes  $i$  and  $j$ .

##### A. Registration by probabilistic relaxation

Probabilistic relaxation or relaxation labeling is an optimization method first proposed by Rosenfeld *et al.* [18]. Following Fu and Yan's notation [19], we define a set of objects to be labeled (the terrain nodes  $V_{\bar{T}}$  in our case), a set of possible labels (the cadastre nodes  $V_{\bar{C}}$ ), initial probabilities  $p^{(0)}$ , an influence function  $d$  and a compatibility function  $c$ .

The influence function  $d_{ij}$ , with  $i, j \in V_{\bar{T}}$ , measures the relative influence a face  $j$  has over the face  $i$ . For non-adjacent faces, it is 0. For adjacent faces, we make it dependent on the area



$a_j$  of the face  $j$ , and on the length  $\ell_{ij}$  of the edge  $(i, j)$ . Specifically,

$$d_{ij} = \kappa_i (k_0 + k_1 \ell_{ij}^{k_2} + k_3 a_j^{k_4}), \quad (7)$$

where  $\kappa_i$  is chosen to fulfill the condition  $\sum_j d_{ij} = 1$ .

The compatibility function  $c_{ij}(\lambda_i, \lambda_j)$ , for  $i, j \in V_{\bar{T}}$  so that  $d_{ij} \neq 0$ , and  $\lambda_i, \lambda_j \in V_{\bar{C}}$ , measures the compatibility of the labeling  $i \mapsto \lambda_i$  and  $j \mapsto \lambda_j$ , with  $0 \leq c \leq 1$  and  $\sum_{\lambda_i} c_{ij}(\lambda_i, \lambda_j) = 1$ .

We have chosen

$$c_{ij}(\lambda_i, \lambda_j) = \begin{cases} (1 - w_{ij})^\beta & \text{if } \lambda_i = \lambda_j, \\ \frac{1 - (1 - w_{ij})^\beta}{|V_{\bar{C}}|} & \text{if } \lambda_i \neq \lambda_j, \end{cases} \quad (8)$$

that is, the more salient the edge between two faces, the better it is these faces be labeled differently. This is because two terrain faces are separated by a non-salient terrain edge when they look very similar, that is, when they actually belong to the same field in reality; thus, we want to favor them belonging to the same cadastre region. Conversely, they are separated by a very salient terrain edge when they look very different, probably because they belong to different fields; in that case, we want to favor them belonging to different cadastre regions.

The initial probability function  $p_i^{(0)}(\lambda)$ , with  $i \in V_{\bar{T}}$  and  $\lambda \in V_{\bar{C}}$ , gives the initial state of the system (a sort of fuzzy initial solution). It should reflect the *a priori* probability that  $i$  is mapped to  $\lambda_i$ , but probabilistic relaxation works correctly even if these are not real *a priori* probabilities obtained through statistical analysis. We chose the following initial probabilities: For each node  $i$  in  $\bar{T}$ , we find the barycenter  $b_i$  of its corresponding face in  $T$ . We then find the face in  $C$

whose center of gravity is closest to  $b_i$ , and its corresponding node in  $\bar{C}$ ,  $c_i$ . We define

$$p_i^{(0)}(\lambda) = \begin{cases} \alpha_i \pi_0 & \text{if } \lambda = c_i, \\ \alpha_i \pi_1 & \text{if } \lambda \in N(c_i), \\ \alpha_i \pi_2 & \text{if } \lambda \in \cup_{b \in N(c_i)} N(b) \setminus \{c_i\}, \\ \alpha_i \pi_r & \text{otherwise} \end{cases} \quad (9)$$

with  $\alpha_i$  such that  $\sum_{\lambda \in V_{\bar{C}}} p_i^{(0)}(\lambda) = 1$ .

An iterative process is then run, repeatedly updating the current probabilities with an update function defined in [19],  $p^{(t+1)} = F(p^{(t)}, c, d)$ , until convergence (or for a maximum number of iterations):

$$p_i^{(k+1)}(\lambda) = p_i^{(k)}(\lambda) \cdot (1 + a_i^{(k)}(\lambda)/q_i^{(k)}), \quad (10)$$

$$a_i^{(k)}(\lambda) = s_i^{(k)}(\lambda) - \bar{s}_i^{(k)}, \quad (11)$$

$$\bar{s}_i^{(k)} = \sum_{\lambda \in V_{\bar{C}}} p_i^{(k)}(\lambda) s_i^{(k)}(\lambda), \quad (12)$$

$$s_i^{(k)}(\lambda) = \sum_{j \in V_{\bar{F}}} d_{ij} \sum_{\eta \in V_{\bar{C}}} c_{ij}(\lambda, \eta) p_j^{(k)}(\eta), \quad (13)$$

with different authors giving different choices for  $q_i^{(k)}$ ,

$$q_i^{(k)} = \begin{cases} 1 + \bar{s}_i^{(k)} & \text{(Rosenfeld *et al.* [18]),} \\ \bar{s}_i^{(k)} & \text{(Zucker *et al.* [20]),} \\ 1 & \text{(Chen and Luh [21], [22]).} \end{cases} \quad (14)$$

After convergence to  $p^{(\infty)}$ , the solution is the labeling

$$s^{(\infty)}(i) = \operatorname{argmax}_{\lambda \in V_{\bar{C}}} p_i^{(\infty)}(\lambda). \quad (15)$$

In our implementation, we treat values of  $p_i^{(t)}(\lambda)$  lower than a small threshold as 0, which greatly reduces processing time without changing the results.

## B. Registration by simulated annealing

We can use simulated annealing to find a near-optimal solution. For this we need an initial solution  $s^{(0)}$ , a way of evaluating the *energy* of a solution, and a way of obtaining solutions similar to a given solution. Starting from a solution  $s^{(0)}$  the algorithm iteratively modifies it to obtain a near-optimal solution  $s^{(\infty)}$ .

To obtain  $s^{(0)}$  we use  $p^{(0)}$  defined in Eq. (9):  $s^{(0)}(i)$  is chosen randomly following the probability distribution given by  $p_i^{(0)}$ . To obtain from  $s$  a similar solution  $s'$ , we replace a certain number  $N_r$  of its labelings; each new labeling  $s'(i)$  is chosen randomly following  $p_i^{(0)}$ .

To evaluate the quality of a solution  $s$ , we do the following: For each  $i \in V_{\bar{T}}$ , we find  $e_s(i)$

$$e_s(i) = a_i k_7 \delta_{\text{exterior}}(i) + a_i^{k_5} \cdot \sum_{j \in N(i)} e'_s(i, j) \quad (16)$$

$$e'_s(i, j) = \begin{cases} d_{ij} (1 - (1 - w_{ij})^\beta) & \text{if } s(i) = s(j), \\ d_{ij} (1 - w_{ij})^\beta & \text{if } s(i) \neq s(j) \end{cases} \quad (17)$$

where  $a_i$  is the area of the face in  $T$  corresponding to the node  $i$  in  $\bar{T}$ ,  $d_{ij}$  is as defined in Eq. (7), and  $\delta_{\text{exterior}}(i)$  is 1 if the node  $i$  is mapped to a special node in  $V_{\bar{C}}$  corresponding to the *outside* of the cadastre, and 0 otherwise (this is a penalty to minimize such mappings). The energy of a solution  $s$ ,  $E(s)$ , is the sum of the  $e_s$  of its nodes, plus a penalty for each node that has no neighbor labeled like itself. If there are  $d$  such nodes,

$$E(s) = k_6 d + \sum_{i \in V_{\bar{T}}} e_s(i). \quad (18)$$

The lower the energy, the better the solution.

## C. Post-processing

The output  $s^{(\infty)}$  of any of these two variants is then processed in several ways. First, each node  $n$  which has no neighbor labeled like  $n$ , is merged to the neighbor node  $n' \in N(n)$  for

which the edge  $(n, n') \in E_{\bar{T}}$  has the lowest apparition weight  $w$ . This is to avoid very small isolated regions. Second, following the mapping defined by  $s^{(\infty)}$ , the connected faces in  $T$  which have the same label are merged. The resulting primal graph  $R$  is taken as the registration of the cadastre graph  $C$  onto the image.

#### D. Registration quality measure

Each edge  $e$  in  $R$  is the concatenation of a certain set of edges  $P(e) \subset T$ . We compute  $m(e)$ , the average of the apparition weights of the edges in  $P(e)$ , weighted by their lengths:

$$m(e) = \frac{\sum_{t \in P(e)} w_t \ell(t)}{\sum_{t \in P(e)} \ell(t)}. \quad (19)$$

This gives a measure of how strong the image edge corresponding to a registered cadastre edge is. As in section III-H, low values indicate that there is probably no corresponding image edge, and high values indicate that there is indeed an image edge at that position. In section V-B we further explore the validity of this indicator.

## V. EXPERIMENTS AND EVALUATION

We have done several experiments to investigate the behavior and performance of these registration algorithms.

We first systematically scanned the parameter space in order to determine the optimal parameter set for each of the algorithms and variants. This is described in section V-A.

With these parameter sets the main experiment was run; in this experiment we run the algorithms in the conditions that will be used in the final, production-grade, processing chain: parameters set for fast convergence, images downsampled at 2 m per pixel, and realistic cadastre data. This is described in section V-B.

Finally, other aspects of the algorithms are tested. In section V-C we explore the effect of convergence speed and input cadastre quality on the algorithms' performance.

#### A. *Parameter set selection*

These algorithms use several parameters and weighting factors, and it is therefore necessary to find appropriate values for them. We found these optimal parameter values by scanning over the parameter space (see table III). It is usually desirable to have as few parameters as possible in a system, which is not the case here; however, we found that the precise value of these parameters is not critical, as —except for the simulated-annealing variant of the region-based algorithm— we obtain good results for a wide range of parameter sets and not just for the optimal parameter set.

#### B. *Main experiment*

We have run these algorithms on a test site of 4 km<sup>2</sup>, on which we defined a ground truth, containing the terrain edges in the image (field boundaries and other strong edges). This can be seen as an ideal registration. The segmentation used to obtain the terrain graph  $T$  was computed with Guigues' scale-sets algorithm [13] using the red, green, and blue color components of an image, downsampled to 2 m per pixel to improve speed.

For the edge-based algorithm, we let the optimizer run for 2500 iterations with a fast cooling schedule, taking around 10 minutes to process the whole test site. For the region-based algorithm, we let the optimizers, in both variants, run for about 4 minutes. The best parameter sets were used for this experiment (see section V-A).

The ground truth and the registered cadastre graph were pixelized. In order to obtain a measure of the performance of each algorithm, we computed the distances between each pixel of the

registered cadastre graph and the ground truth, and the distances between each pixel of the *unregistered* cadastre graph and the ground truth. Let  $I_c \subset \mathbb{Z}^2$  be the set of pixels of a cadastre graph (registered or not), and  $I_g \subset \mathbb{Z}^2$  that of the ground truth. The distance between a pixel  $i_c \in I_c$  and the ground truth is

$$d(i_c, I_g) = \min_{i_g \in I_g} \|i_c - i_g\|, \quad (20)$$

and the average distance between the cadastre graph and the ground truth is

$$d(I_c, I_g) = \frac{1}{|I_c|} \sum_{i_c \in I_c} d(i_c, I_g). \quad (21)$$

A histogram of these distances  $d(i_c, I_g)$  is shown in Fig. 6.

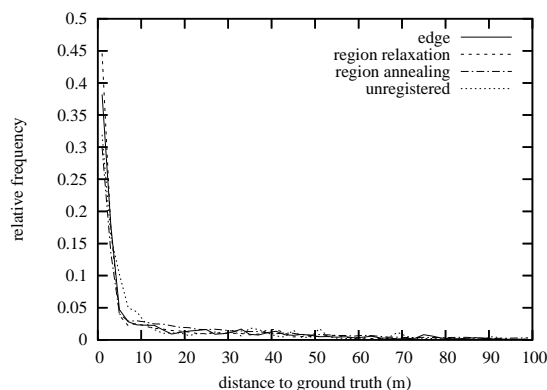


Fig. 6. Histogram of distances from cadastre graph pixels to ground truth, in meters, for the cadastre registered by the edge-based algorithm (*edge*), by the region-based algorithm with probabilistic relaxation (*region relaxation*), by the region-based algorithm with simulated annealing (*region annealing*), and for the unregistered cadastre (*unregistered*).

Since some cadastre edges have no matching ground truth edge (because the cadastre edge does not correspond to any edge in the image), the histogram tails off to large distances. Therefore, in order to obtain a meaningful aggregate quality measure, we excluded cadastre edges for which the average distance to the ground truth was larger than 6 m, or which had a pixel that was farther than 20 m away from the ground truth, and we calculated the average distance to the ground

truth of the pixels in the remaining cadastre edges. These thresholds were manually chosen after inspecting the input data and these histograms, and are meant to exclude *non-matchable* edges (cadastre edges which do not have a corresponding image edge). Since these thresholds are applied to both the unregistered and the registered cadastre, their actual value is of little importance for finding how much the registration reduces the average distance in relative terms. We found experimentally that misregistered edges were never so far away from the ground truth.

A histogram of the distances from the remaining cadastre edges to the ground truth is shown in Fig. 7, and a summary is given in table I. In this table, *length* is the total length of the unregistered or registered cadastre edges, including non-matchable edges, and *length (matchable)* is that *excluding* non-matchable edges. The average distances given in this table are these calculated using Eq. (21). Fig. 8 shows graphical results for a part of the test region. The large differences in table I in length between the unregistered and registered cadastres has a simple explanation: unregistered cadastre edges are often long straight lines, whereas registered edges follow terrain details with pixel accuracy. Registered edges are therefore much longer than unregistered ones.

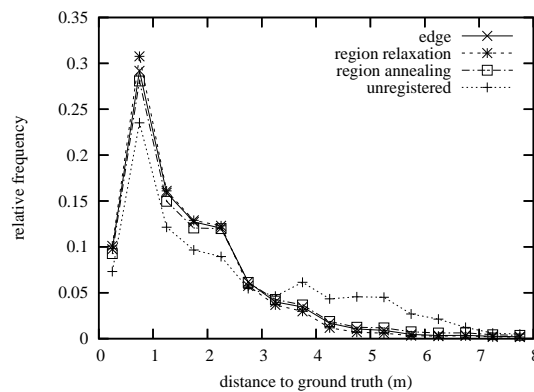


Fig. 7. Histogram of distances from cadastre graph pixels to ground truth, in meters, excluding non-matchable edges (main experiment: 2 m per pixel, fast convergence, best parameter set). Note how the registered histograms shows higher counts than the unregistered histogram for low distances.

TABLE I

EVALUATION RESULTS (MAIN EXPERIMENT).

cadastre	unregistered	registered	registered	registered
algorithm		edge	reg. relax.	reg. anneal.
length	95.9 km	188.0 km	210.6 km	283.5 km
length (matchable)	32.7 km	61.6 km	66.7 km	76.4 km
<b>mean distance</b>	2.351 m	<b>1.640 m</b>	<b>1.594 m</b>	<b>1.844 m</b>
iterations		2500	1000	15000
<b>distance reduction</b>		<b>30.23%</b>	<b>32.19%</b>	<b>21.58%</b>

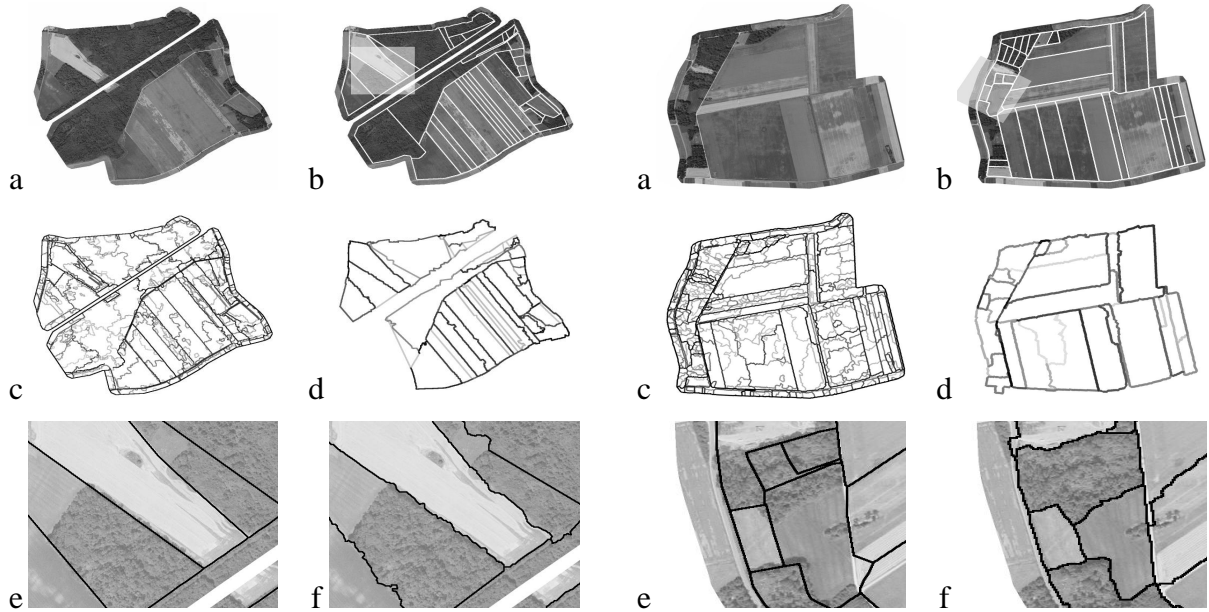


Fig. 8. Results for a  $1.4 \text{ km}^2$  portion of the test site, processed by the edge-based algorithm (left half) and for a  $1 \text{ km}^2$  portion of the test site, processed by the region-based algorithm, with probabilistic relaxation (right half). Source image (a); cadastre graph  $C$  (b, boxed areas are shown enlarged in the bottom row); terrain graph  $T$  (c, darker edges have higher apparition weights); registered cadastre (d, darker edges have higher registration ratios); close-up on the unregistered (e) and registered (f) cadastre.



The presented algorithms successfully register the cadastre onto the image. With the best parameter set, the edge-based algorithm reduces the average distance between the cadastre graph and a ground truth from 2.35 m to 1.64 m, 30.2% less; the region-based algorithm reduces it to 1.59 m, 32.2% less, with the probabilistic relaxation optimization, and to 1.84 m, 21.6% less, with the simulated annealing optimization. More important than this numerical result—which is nonetheless useful for comparison to other algorithms, or for selecting the best parameter set—is the fact that the resulting graph is registered onto terrain edges, hence onto salient edges in the image, and therefore statistical analysis of these regions will be less perturbed by adjacent regions.

Visual inspection (see subfigures 8*d*) seems to show that the registration quality measures presented in sections III-H and IV-D indicate if a cadastre edge actually exists in the image. Using this information to delete cadastre edges that cannot be found in the image should be straightforward. In order to determine if this quality measure actually indicates if a cadastre edge actually exists in the image, we have calculated, for each value of this quality measure (suitably discretized and binned) the average, for all pixels in the registered cadastre which have the given value for the quality measure, of the distance to the nearest pixel in the ground truth. This is shown in Fig. 9. As before, we should consider that for pixels with a distance to ground truth below a certain threshold (for example, 6 m, as before) there is a corresponding image edge, and that for pixels above that threshold there is actually no corresponding image edge. We clearly see, for the edge-based algorithm and for the region-based algorithm with simulated annealing, that cadastre pixels with a corresponding image edge have, on average, a high value of the registration quality measure, whereas those pixels that do not have a corresponding image edge have, on average, a low value of the registration quality measure. More research is needed

for the region-based probabilistic relaxation method.

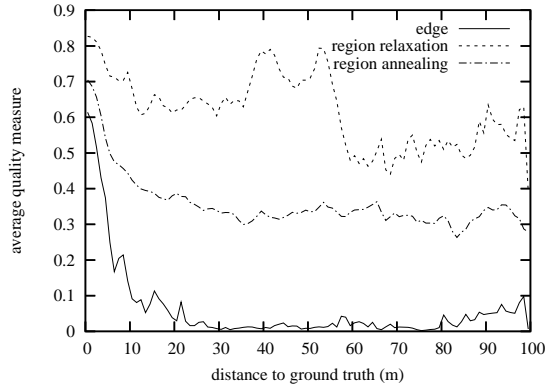


Fig. 9. Average value, for each pixel in the registered cadastre, of the registration quality measure as a function of the distance from that pixel to the nearest ground truth pixel.

### C. Other experiments

In this section we present the results of tests exploring the effect of the cooling schedule and the cadastre quality on the algorithms' performance.

1) *Increased number of iterations:* Execution speed is critical for our application, which is why we worked with downsampled images and we let the optimizer run for a relatively small number of iterations (2500 for the edge-based method, 1000 for the relaxation region-based method, and 15000 for the annealing region-based method), with parameters set for a fast convergence, or a fast cooling schedule. For example, for simulated annealing, we have used a cooling schedule  $T_n = T_0 \cdot k^n$  with  $k$  close to, but less than, 1; a slower cooling schedule is one with a  $k$  closer to 1 than a faster cooling schedule. For the edge-based algorithm, we took  $k = 0.9$ , and for the simulated annealing region-based algorithm, we took  $k = 0.98$ . These give fast execution times of the order of 1 minute per square kilometer.

However, to fully evaluate its performance we also ran it for more iterations (15000 for the edge-based method, 10000 for the relaxation region-based method, and 150000 for the annealing

region-based method), with parameters set for a slower convergence ( $k = 0.97$  for the edge-based algorithm, and  $k = 0.99$  for the simulated annealing region-based algorithm). For all these test we used the optimal parameters found in section V-A.

The results for the experiments with slow convergence with 2 m-per-pixel images are listed in table II under *slow convergence*. It can be seen that there is little difference between forcing a fast convergence and allowing a slower convergence —actually, slower convergence gives slightly worse results, although this could be just bad chance, since there is a random component in these algorithms. Slower convergence, however, comes at the cost of increased computation time. This justifies our choice of a fast convergence as the default setting.

2) *Worse input cadastre*: A related issue is that it may be difficult to fully appreciate the performance of this algorithm with this cadastre data, since the initial mis-registration is not very large in terms of distance to the ground truth (but note that reducing this distance is only one of the goals of this algorithm, the other being to obtain cadastre edges that follow the geometrical details of the image edges). That is, should the best performance of these algorithms be given as “reduces average distance to 1.7 m” or “reduces average distance by 30%”?

To answer that, we manually distorted the cadastre graph and ran the algorithm, with 2 m-per-pixel images, with the fast cooling algorithm and the optimal parameter set. The results are listed in table II under *distorted cadastre*. It can be seen that despite the quality of the distorted unregistered cadastre was much worse (*unreg. dist.* column), the registration of both cadastres reaches similar distances to ground truth (*distance* column). It seems that the obtained quality levels are some sort of upper limit that the algorithms reach independently of the quality of the input data, but beyond which they cannot perform, and that therefore the performance should be described as “reduction to 1.7 m” instead of “reduction by 30%”.

#### D. Summary of experiments

Table II summarizes the results of all the experiments. Using a slower cooling schedule does not improve the registered distance. On the other hand, tests with a manually distorted cadastre graph show that the upper performance limits are to be measured in absolute distances, not as a relative improvement, for both with the small unregistered distance of the original cadastre and the larger unregistered distance of the distorted cadastre the algorithms reach similar registered distances.

TABLE II

EVALUATION RESULTS FOR ALL EXPERIMENTS (“CONV.”: COOLING SCHEDULE, OR SPEED OF CONVERGENCE; “UNREG. DIST.”: DISTANCE FOR UNREGISTERED CADASTRE).

conv.	cadastre	algorithm	unreg. dist.	distance	
fast	original	edge	2.351 m	1.640 m	-30.23%
fast	original	reg. relax.	2.351 m	1.594 m	-32.19%
fast	original	reg. anneal.	2.351 m	1.844 m	-21.58%
fast	distorted	edge	2.831 m	1.665 m	-41.20%
fast	distorted	reg. relax.	2.831 m	1.595 m	-43.65%
fast	distorted	reg. anneal.	2.831 m	2.020 m	-28.62%
slow	original	edge	2.351 m	1.665 m	-29.18%
slow	original	reg. relax.	2.351 m	1.605 m	-31.72%
slow	original	reg. anneal.	2.351 m	2.001 m	-14.90%

## VI. HOMOGENEITY TESTS

The algorithms described above do not always produce homogeneous regions. If a cadastre region is itself not homogeneous —and not because of some stray pixels in the border, which would be corrected by the registration algorithms, but because it actually contains more than one

plot— in most cases the registration algorithms will produce a heterogeneous region. Depending on the application, we may need a way to detect these heterogeneous regions and, eventually, to decompose them into homogeneous sub-regions. We propose two ideas for future research:

We could analyze the terrain edges contained in the registered cadastre regions, and calculate their maximum appearance weight —excluding terrain edges closer than a certain distance to the region edge. High values would indicate that the region is not homogeneous and should be split into several regions.

In the context of land-use classification, another clue to the fact that a region is heterogeneous would be that per-region classification algorithms report a very low confidence to whatever single terrain class is assigned to the region. In that case, we should attempt to partition the region following high-saliency edges, classify each of the sub-regions, and compare their confidences to that of the whole region.

Another problem is that it is possible that a single plot spans several cadastre regions. Although not as important a problem as that of registered regions being heterogeneous, it may be appropriate to merge adjacent cadastre regions containing the same crop. This can be done by using the registration quality measures of both algorithms as indicators of whether or not the edge follows a true terrain limit.

## VII. DISCUSSION AND CONCLUSION

In this article we have presented two graph matching algorithms specifically tailored for asymmetric graph matching problems —where matches are, in most cases, not one-to-one— which can be used to register a cadastre graph onto an aerial image, or in general to partition an image into objects following a corresponding, but geometrically imprecise, external partition.

We use a multi-scale segmentation of the image which is converted to a weighted graph.

In the edge-based algorithm, the edges of this graph are asymmetrically matched to the edges of the cadastre graph—which correspond to edges between land plots—by optimizing, using simulated annealing, the fitness of a solution. We use an ad-hoc method for the registration near cadastre nodes. In the region-based algorithm, it is the faces of the terrain graph which are matched to the faces of the cadastre graph—each such face corresponds to a land plot—, and either probabilistic relaxation or simulated annealing can be used for the optimization.

Extensive tests show that, numerically, the region-based method with probabilistic relaxation performs best, followed by the edge-based method, and, far behind, the region-based method with simulated annealing. The choice between the first two is not so clear-cut, however, because the algorithms exhibit qualitatively different behavior: The edge-based algorithm preserves the spatial distribution of the cadastre graph much better, at the cost of adding auxiliary edges that do not correspond to image edges. The region-based algorithm, on the other hand, strictly follows image edges, at the cost of not always preserving spatial distribution. Although numerical evaluation gives better figures for the region-based algorithm with probabilistic relaxation, for a given application the trade-off between geometrical precision and topology preservation should be taken into account and perhaps the edge-based algorithm may be a better choice.

As a side effect of the registration we obtain a parameter, the registration quality measure, which seems useful as an indicator of which cadastre edges actually exist in the image. Formal tests to determine if this is actually a good indicator for that purpose support this, but show that further work may be necessary.

## APPENDIX I

### IMPLEMENTATION DETAILS

The best parameter sets found by scanning the parameter space, and used in the experiments, are given in table III.

TABLE III

BEST-PERFORMING PARAMETER SETS.

edge	$\alpha_1 = 6.5$	$\alpha_2 = 0.22$	$\alpha_3 = 3$	$\alpha_4 = 18$
	$\alpha_5 = 0$	$\alpha_6 = 8000$	$N_r = 5$	$p_{\perp} = 0.3$
region relax.	$k_0 = 0$	$k_1 = 1$	$k_2 = 1$	$k_3 = 1$
	$k_4 = 1$	$q_i^{(k)} = 1$	$\pi_0 = 1.1$	$\pi_1 = 0.6$
	$\pi_2 = 0.4$	$\pi_r = 0.1$	$\beta = 1$	
region anneal.	$k_0 = 0$	$k_1 = 1$	$k_2 = 1$	$k_3 = 2$
	$k_4 = 0.7$	$k_5 = 0.7$	$k_6 = 0.1$	$k_7 = 0.1$
	$\pi_0 = 0.83$	$\pi_1 = 0.13$	$\pi_2 = 0.10$	$\pi_r = 0.03$
	$\beta = 2$			

### REFERENCES

- [1] J.-M. Viglino and L. Guigues, "Géoréférencement automatique de feuilles cadastrales," in *Proc. 13ème Congrès de Reconnaissance de Formes et Intelligence Artificielle (RFIA 2002)*. Angers, France: AFRIF-AFIA, Jan. 2002, pp. 135–143.
- [2] P. Cachier, E. Bardinet, D. Dormont, X. Pennec, and N. Ayache, "Iconic feature based nonrigid registration: the PASHA algorithm," *Computer Vision and Image Understanding*, vol. 89, no. 2/3, pp. 272–298, Feb./Mar. 2003.
- [3] A. Goshtasby, L. Staib, C. Studholme, and D. Terzopoulos, "Nonrigid image registration: guest editors' introduction," *Computer Vision and Image Understanding*, vol. 89, no. 2/3, pp. 109–113, Feb./Mar. 2003.
- [4] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2/3, pp. 114–141, Feb./Mar. 2003.
- [5] C. Hivernat and X. Descombes, "Mise en correspondance et recalage de graphes: application aux réseaux routiers extraits d'un couple carte/image," INRIA, <http://www.inria.fr>, Tech. Rep. RR-3529, Oct. 1998.

- [6] S. Gautama and A. Borghraef, "Using graph matching to compare VHR satellite images with GIS data," in *Proc. Intl. Geoscience and Remote Sensing Symposium (IGARSS 2003)*. Toulouse, France: IEEE GRSS, July 2003.
- [7] V. Walter, "Automatic classification of remote sensing data for GIS database revision," in *International Archives of Photogrammetry and Remote Sensing (IAPRS)*, vol. 32, Stuttgart, Germany, 1998, pp. 641–648.
- [8] R. C. Wilson and E. R. Hancock, "Structural matching by discrete relaxation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 19, no. 6, pp. 634–648, June 1997.
- [9] M. Gori, M. Maggini, and L. Sarti, "Exact and approximate graph matching using random walks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 27, no. 7, pp. 1100–1111, July 2005.
- [10] R. Trias-Sanz, "An edge-based method for registering a graph onto an image with application to cadastre registration," in *Proc. of the 2004 Conf. on Advanced Concepts for Intelligent Vision Systems (ACIVS 2004)*, Brussels, Belgium, 2004, pp. 333–340.
- [11] R. Trias-Sanz and M. Pierrot-Deseilligny, "A region-based method for graph to image registration with application to cadastre data," in *Proc. IEEE Intl. Conf. on Image Processing (ICIP 2004)*. Singapore: IEEE, Oct. 2004.
- [12] D. Marr, *Vision*. Freeman and Co., 1982.
- [13] L. Guigues, H. Le Men, and J.-P. Cocquerez, "Scale-sets image analysis," in *Proc. IEEE Intl. Conf. on Image Processing (ICIP 2003)*. Barcelona, Spain: IEEE, Sept. 2003.
- [14] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
- [15] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerical Mathematics*, vol. 1, pp. 269–271, 1959.
- [16] D. Jungnickel, *Graphs, Networks and Algorithms*, ser. Algorithms and Computation in Mathematics. Berlin: Springer, 1999, vol. 5.
- [17] T. Law, H. Itoh, and H. Seki, "Image filtering, edge detection, and edge tracing using fuzzy reasoning," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 18, no. 5, pp. 481–491, May 1996.
- [18] A. Rosenfeld, R. Hummel, and S. Zucker, "Scene labeling by relaxation operations," *IEEE Transactions on Systems, Man, and Cybernetics (SMC)*, vol. 6, pp. 320–433, June 1976.
- [19] A. M. N. Fu and H. Yan, "A new probabilistic relaxation method based on probability space partition," *Pattern Recognition*, vol. 30, no. 11, pp. 1905–1917, 1997.
- [20] S. W. Zucker, E. V. Krishnamurthy, and R. L. Haar, "Relaxation processes for scene labeling: convergence, speed and stability," *IEEE Transactions on Systems, Man, and Cybernetics (SMC)*, vol. 8, pp. 41–48, 1978.
- [21] Q. Chen and J. Y. S. Luh, "Ambiguity reduction by relaxation labeling," *Pattern Recognition*, vol. 27, pp. 165–180, 1994.
- [22] —, "Relaxation labeling algorithm for information integration and its convergence," *Pattern Recognition*, vol. 28, pp. 1705–1722, 1995.





**Roger Trias-Sanz** received the M.Sc. degree in telecommunications engineering from the Technical University of Catalonia, Barcelona, in 2001, and the Ph.D. degree in computer sciences from the René Descartes University, Paris, in 2006, after a one-year break to study Artificial Intelligence and Image Analysis in Paris, and work in the French National Mapping Agency in automatic interpretation of high-resolution color aerial images for the automatic production of fine-scale maps.



**Dr. Marc Pierrot-Deseilligny** is a graduate from the Ecole Polytechnique and ‘Habilitation à Diriger des recherches’ from the René Descartes University, Paris. He has been working since 1990 in image analysis applied to automatic information extraction for geographic systems in several companies. Between 2001 and 2003, he has been head of MATIS, the image analysis laboratory of IGN, the French mapping agency). Since 2003 he is head of research at IGN. His current personal research topics focus on image matching

for photogrammetry.



**Jean Louchet** Biography text here.



**Georges Stamon** Biography text here.