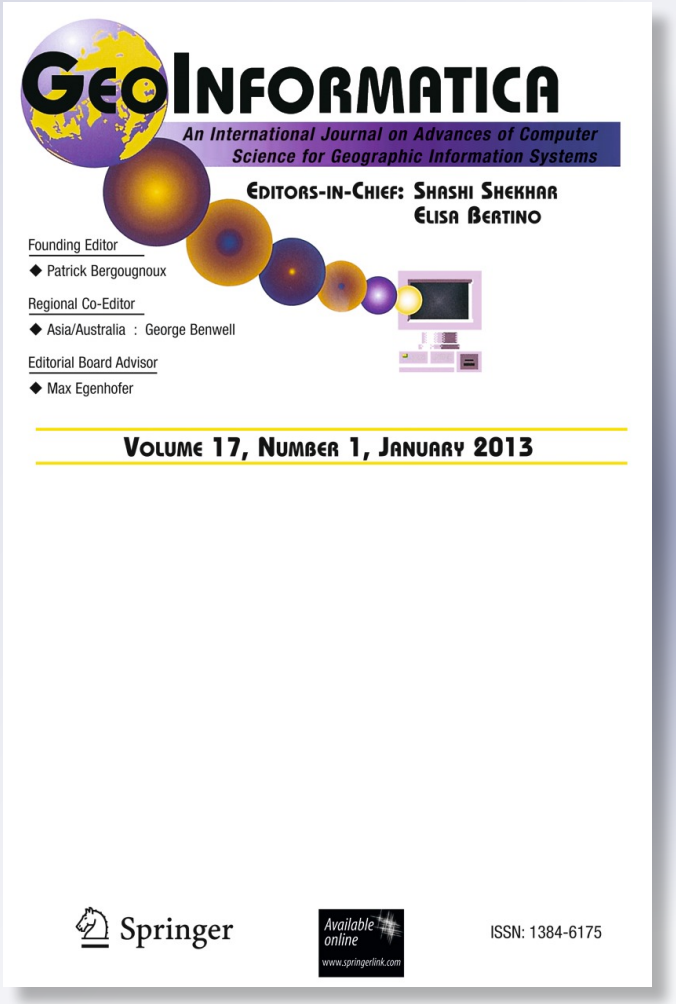
Building pattern recognition in topographic data: examples on collinear and curvilinear alignments

Xiang Zhang, Tinghua Ai, Jantien Stoter, Menno-Jan Kraak & Martien Molenaar

GeoInformatica
An International Journal on Advances of Computer Science for Geographic Information Systems

ISSN 1384-6175
Volume 17
Number 1

Geoinformatica (2013) 17:1-33
DOI 10.1007/s10707-011-0146-3



Springer

Your article is published under the Creative Commons Attribution license which allows users to read, copy, distribute and make derivative works, as long as the author of the original work is cited. You may self-archive this article on your own website, an institutional repository or funder's repository and make it publicly available immediately.

Geoinformatica (2013) 17:1–33
DOI 10.1007/s10707-011-0146-3

Building pattern recognition in topographic data: examples on collinear and curvilinear alignments

Xiang Zhang · Tinghua Ai · Jantien Stoter · Menno-Jan Kraak · Martien Molenaar

Received: 6 November 2010 / Revised: 3 October 2011 /
Accepted: 4 October 2011 / Published online: 28 October 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract Building patterns are important features that should be preserved in the map generalization process. However, the patterns are not explicitly accessible to automated systems. This paper proposes a framework and several algorithms that automatically recognize building patterns from topographic data, with a focus on collinear and curvilinear alignments. For both patterns two algorithms are developed, which are able to recognize alignment-of-center and alignment-of-side patterns. The presented approach integrates aspects of computational geometry, graph-theoretic concepts and theories of visual perception. Although the individual algorithms for collinear and curvilinear patterns show great potential for each type of the patterns, the recognized patterns are neither complete nor of enough good

X. Zhang (✉) · T. Ai
School of Resource and Environmental Science, Wuhan University, 430079 Wuhan, China
e-mail: xiangzhangchina@gmail.com

T. Ai
e-mail: tinghuaai@gmail.com

X. Zhang · M.-J. Kraak · M. Molenaar
Faculty of Institution for Geo-Information Science and Earth Observation,
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

X. Zhang
e-mail: xzhang@itc.nl

M.-J. Kraak
e-mail: kraak@itc.nl

M. Molenaar
e-mail: molenaar@itc.nl

J. Stoter
GIS Technology, OTB, Delft University of Technology, Delft, The Netherlands
e-mail: j.e.stoter@tudelft.nl

quality. We therefore advocate the use of a multi-algorithm paradigm, where a mechanism is proposed to combine results from different algorithms to improve the recognition quality. The potential of our method is demonstrated by an application of the framework to several real topographic datasets. The quality of the recognition results are validated in an expert survey.

Keywords Building patterns · Building alignments · Pattern recognition · Data enrichment · Computational geometry · Visual perception theory · Map generalization

1 Introduction

Building patterns are important local structures that characterize urban and other built-up spaces [7]. Other characteristic urban structures are neighborhood level street patterns (e.g. radial and grid networks) that describe the layout of streets, roads and other thoroughfares [20], as well as city-wide patterns that describe the relationship between cities and/or sub-cities (e.g. satellite and network cities). These hierarchically interconnected structures always have historical, social, cultural and functional implications of built environments [7], which is important information in many disciplines. A traditional way to comprehend and communicate such information is topographic maps, a very effective visualization tool to convey patterns generated by underlying geographic processes [19].

To better understand and interpret patterns and their deeper implications at different scales, topographic maps should be carefully designed and generalized (both graphically and syntactically) so as to preserve the patterns. Current research on automated map generalization [11, 13, 28, 29] shows the needs for a better understanding of the generalization of important characteristics and patterns (i.e. syntactic content of a map), and for evaluation techniques that can assess how well such patterns are kept during the generalization. After all, map generalization is essentially a process of representing and discerning important geographic characteristics and patterns, by suppressing unnecessary details and noises for certain purposes and map scales.

Currently, topographic data have rich geometries but lack explicit knowledge on relationships, structures, patterns and higher semantics of spatial objects. Stoter et al. [30] showed that a cartographer adds this knowledge in the interactive generalization process. In an automated process this knowledge need to be available at computer level. Fortunately, such knowledge can be inferred from geometries of such objects and their spatial relationships [24], much the same way as an experienced reader interprets a map.

Therefore, our objective is to enrich topographic datasets with structural information on building patterns in an automated manner, which is particularly useful for the automated generalization and evaluation in urban and rural contexts. In particular, the enriched information can be used to facilitate map object grouping in support of meaningful aggregation and other operators. For example, it can be used to improve building typification [6] and displacement algorithms [4], by increasing the weights of buildings belonging to certain patterns. In addition, building patterns are useful

in much broader fields, such as spatial analysis, database enrichment for semantic interoperation, information retrieval, etc. For instance, Lüscher et al. [17] used the patterns to infer higher semantics such as terraced houses, and Gersmehl [12] suggests that functional aspects (e.g. residential houses, resorts or military bases) of a built-up space can be implied from the alignments of buildings and their relationships to nearby roads.

Basically, building patterns can be divided into two types: linear alignments and nonlinear clusters. Linear alignments are common features that can be found in less dense part of urban areas (e.g. residential areas), suburban and rural areas, whereas nonlinear clusters almost mainly occur in densely populated regions such as inner city (i.e. city centers) and commercial/industrial areas. When it comes to map generalization, nonlinear clusters are simply generalized into blocks or built-up areas (i.e. polygons enclosed by surrounding roads) due to their high densities [27]. Linear alignments, on the other hand, are usually preserved or even enhanced during the generalization, in which their forms and characteristics have to be considered [27]. Therefore, we will focus on two linear alignments (i.e. collinear and curvilinear) in this paper.

We propose a framework that automatically recognizes building patterns from topographic data. This framework integrates aspects of computational geometry, graph-theoretic concepts and theories of visual perception. It advocates the use of multiple algorithms to detect different types of building patterns followed by a combination process. We claim that in this way the quality and completeness of the recognized building patterns can be improved.

The remainder of this paper is structured as follows. Section 2 briefly reviews previous attempts to detect specific building patterns for automated map generalization. Section 3 clarifies the context and definition of building patterns and discusses some abstract attributes of building patterns that help to better understand our work. After presenting the recognition framework and algorithms in Section 4, Section 5 carefully designs experiments, in which an expert survey is carried out to validate the recognition quality. Section 6 then discusses the experimental results. This paper ends with conclusions (Section 7).

2 Related work

Automated map generalization has reported many approaches to the detection of building patterns in topographic data. Regnauld [22, 23], for instance, adopted a graph-theoretic approach based on [32] to detect clusters for building typification. His approach, however, leaves large space for further refinements. The algorithms presented in this paper partly follow this graph-theoretic approach, coupled with visual perception rules to recognize more refined patterns. Christophe and Ruas [8] focused on the detection of collinear alignments using a straight line based technique. Their results contained many false positives which were rejected by a subsequent quality assessment. It is worth mentioning that this technique is similar to Hough transform [14] in computer vision, in which only patterns with analytic forms can be detected (e.g. straight lines and circles). To accommodate better integration of multiple generalization operators, Li et al. [16] detected building groups by firstly

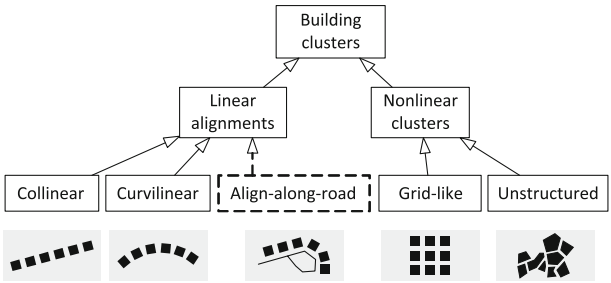


Fig. 1 Typology and representative symbols of building patterns in topographic data

identifying building pairs (i.e. two proximate buildings that look similar) and then connecting adjacent pairs that share a same building. This approach could lead to some foreseeable false positives, and the assumed elongated building form (i.e. having major axes) further limits its applicability.

Motivated by automatic interpretation and generalization of spatial data, Anders [2] proposed a hierarchical parameter-free clustering algorithm, which uses nearest neighbor graph, minimum spanning tree, relative neighborhood graph, Gabriel graph and Delaunay triangulation. This algorithm can find natural groups of objects without having to specify the number of clusters to be identified. Based on the identified clusters, Anders [3] detected grid-like building patterns on top of relative neighborhood graph in support of building typification.

To get more insight into the local settlement structures available on topographic maps, we have proposed a hierarchical typology of building patterns (i.e. clusters) [33], distinguishing between linear alignments (collinear, curvilinear and align-along-road) and nonlinear clusters (grid-like and unstructured), see Fig. 1. In this previous work, we have developed two individual algorithms for recognizing align-along-road and unstructured building patterns (i.e. arbitrary clusters). Because we have focused on building clusters, arbitrary arrangement of buildings (i.e. isolated buildings with random spatial distributions) was not covered in this typology. However, it is an important arrangement and thus detection algorithms also need to be devised.

To summarize, all the reviewed work focused on individual algorithms for detecting certain types of building patterns (e.g. collinear alignment). Since the boundary between different pattern types can be ambiguous [8], it is not clear yet whether those techniques can work together and how the results of different pattern types can be combined together automatically. This is important because otherwise substantial human effort is required to decide on which approach to apply in which circumstance.

Our approach consists of three major components. First, we put forward an alternative to the detection of collinear alignments. Second, we propose an approach to recognizing curvilinear alignments—important but previously less discussed alignments. For both pattern types two algorithms are developed, which are able to recognize alignment-of-center and alignment-of-side patterns. Finally, as patterns detected by different algorithms may conflict with each other, we propose a framework to combine the patterns from respective algorithms in order to find the complete and best possible results. Again, this issue has not been tackled previously. This proposed framework should give better results than the single algorithm approach.

3 Building patterns in topographic data

This section explains the use of the term 'pattern' in different geo-spatial contexts and in which context the concept of 'building pattern' is meaningful. After that, a semi-formal definition of building pattern is given, followed by a discussion of its abstract attributes.

3.1 Meanings of geo-spatial pattern

The geo-spatial domain distinguishes between generic pattern and domain-specific pattern. The generic pattern is the output of a spatial data mining process which aims at discovering interesting and previously unknown, but potentially useful knowledge from large datasets (e.g. determine the "hot spots" in crime analysis) [25]. Because the discovery and description method for such patterns are generic (e.g. those for spatial outliers, co-locations and clustering), we term them generic patterns. The domain-specific (geo-spatial) patterns, on the other hand, have four fundamental features:

1. The detection of such patterns does not yield any new knowledge. In fact, most of them are results of geographic processes that are already known;
2. Most of such patterns are recognized and described in ad-hoc manners (e.g. those for building, river, and street patterns);
3. Such patterns have their typical structures and may quite differ from others;
4. Underlying geographic processes which generate different patterns can be inferred from the structures of the patterns.

Building patterns are domain-specific and have their distinctive spatial structures or arrangements (cf. Fig. 1). In fact, they are the result of urban design and development processes where morphological, perceptual and visual principles are employed [7, 18].

3.2 Definition and abstract attributes of building patterns

Semi-formally, building patterns can be defined as building clusters whose elements, i.e., buildings, are homogeneous in terms of specific properties (e.g. spacing, size, orientation, shape and density), and which have typical spatial arrangements. This section discusses some abstract attributes of building patterns which improve the understanding of building patterns and our recognition approach in a broader context.

Visual significance — The repetition and recurrence of some selected visual elements (e.g. shape, size and color) or their combinations will lead readers to recognize significant building patterns. To emulate this process, our detection algorithms use Gestalt principles of perceptual organization [31]. These principles include *proximity*, *similarity*, and *good continuity* (i.e. smooth connection among compositional parts). Many geo-spatial processing assume implicitly a *common region* principle, as for instance buildings are usually partitioned by road networks before processing. In addition, *gradual variation* or *rhythm* is of great importance

for the detection of curvilinear alignments (see Section 4.4.3). Although this principle contradicts somewhat the concept of similarity, it remains that the pattern elements are in general similar, with gradual variations that are perceptually tolerable. It is noticeable that the repetition, good continuity and gradual variation properties are also in line with Alexander's "The Nature of Order" theory about built spaces [1].

Symbolization — It is the ability to represent something (or a group) by a simple, familiar symbol through association, resemblance, or convention, and symbolization may differentiate one group from another. Topographic patterns are usually symbolized by developing templates or representative symbols that resemble their typical arrangements. Examples include radial and trellis patterns in drainage system, collinear, curvilinear, and grid-like patterns (Fig. 1) in building structure. The algorithms of in this paper are formalized templates for recognizing building patterns and their variations.

Ambiguity — The definition and recognition of building patterns are inherently ambiguous, and the ambiguities in general take two forms. First, in the grouping process it is sometimes tricky to decide whether a building belongs to this or that group. Second, recognition approaches generally make crisp distinctions between different pattern types, though the difference may be ambiguous [8]. This means that a group of buildings could be recognized as a collinear, a curvilinear or an align-along-road alignment, depending on the person or approach involved. It may also be the case that some patterns will not be found by one algorithm and will be recognized by another (see Section 5.2.2). Therefore, combining different results for a more refined and complete result (Section 4.6) becomes necessary. Section 6.3 will further discuss the involved ambiguities.

4 Method: recognition framework and algorithms

This section firstly introduces the recognition framework in Section 4.1. Sections 4.2–4.6 explain in detail the major steps of the proposed framework.

4.1 Framework overview

The proposed framework consists of six steps (Fig. 2). It uses multiple algorithms to recognize building patterns from topographic data:

1. Constructing proximity graph—ProxG$<V, E>$ on input data: buildings are vertices and proximity relationships among objects are edges (see Section 4.2);
2. Performing pre-clustering based on ProxG after Step 1. Minimum Spanning Tree (MST) is firstly derived from ProxG. Then significant inter-cluster edges are identified and pruned from the MST (Section 4.2). After this step, some general clusters are separated;

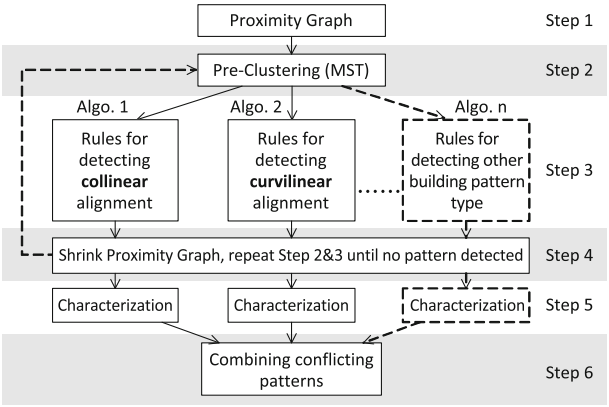


Fig. 2 Multi-algorithm framework for building pattern recognition

3. Applying different algorithms for detecting different building patterns, based on the pruned MST (see e.g. the tracing algorithm in Section 4.3). For each algorithm, different rules are employed to detect a specific pattern type (e.g. Sections 4.4.1–4.4.3);
4. Shrinking ProxG and repeating Steps 2–3 to find more patterns: after patterns have been detected, they are added to the result set and their element buildings are removed from ProxG, and then Steps 2 and 3 are re-executed. This process is repeated until there is no pattern can be recognized by the algorithms from the remaining ProxG. For each loop, detected patterns are added to the result set (see detailed explanation in Section 6.1);
5. Characterizing the detected patterns using a homogeneity measure described in Section 4.5. The homogeneity measure is implemented with subtle differences that are known to specific detection algorithms (also see Section 4.5);
6. Evaluating the conflicting patterns recognized by different algorithms against each other, and combining them into a final result (Section 4.6).

The following sections explain the framework step by step. These steps focus on algorithms and rules for the recognition of collinear and curvilinear alignments, and how the results can be combined.

4.2 Step 1 & 2—deriving building clusters based on proximity graph and MST

Delaunay triangulation (DT) is an ideal tool to model contextual and proximity relationship between spatial objects in a natural way [2, 15]. In our approach, we adopt constrained Delaunay triangulation (CDT), where line segments of buildings and roads are constrained lines of CDT. To derive proximity graph from CDT, buildings are modeled as vertices and any two buildings (v_i and v_p) that are connected directly by at least one triangle form an $edge(v_i, v_p)$ of the graph.

More specifically, we refine the implementation of CDT by inserting extra points to constrained lines. The interval between inserted points is based on the minimal distance among the data points. We choose to construct the refined CDT based on object outlines rather than on object centers. This is because this refined CDT provides a better way of modeling the proximity relationship between buildings and between buildings and roads (see e.g. Fig. 11a) than basing CDT on centers.

In addition, the computation of nearest distance between buildings and between buildings and roads on this structure becomes more efficient [33]. This step results in initial proximity graph.

Then, we derive Minimum Spanning Tree (MST) from the proximity graph. In graph theory, a spanning tree of an undirected graph G is a tree that contains all vertices of G. The weight of a tree is defined as the sum of the weights of all its constituent edges. A minimum spanning tree of G is a spanning tree whose weight is the minimum among all spanning trees of G. In our approach, each MST edge is weighted by the nearest distance between building outlines connected by this edge. To group building clusters that are perceptually significant, we adopt the notion of inconsistent edge developed by Zahn [32]. It is formally defined as follows:

$$edge_i = \begin{cases} \text{inconsistent}, & \text{if } w_i > I_l \cap w_i > I_r \\ \text{consistent}, & \text{otherwise} \end{cases}, \tag{1}$$

where w_i is the weight of $edge_i$; I is a measure of significance that can be defined on both left (I_l) and right (I_r) side of $edge_i$:

$$I = \max\left\{ f \cdot \text{mean}_{\text{weight}},\ \text{mean}_{\text{weight}} + n \cdot \text{std}_{\text{weight}} \right\} . \tag{2}$$

In Eq. 1, inconsistency means that an edge's weight is significantly larger than the mean weight of its nearby edges on both sides (if not applicable, one side) of this edge. Consider $I = \text{mean}_{\text{weight}} + n \cdot \text{std}_{\text{weight}}$, I is larger than the mean weight by n units of standard deviation. If a normal distribution is assumed, then I, with $n = 3$, is larger than 99.7% weights in the distribution. It is in this sense that '*significantly larger than*' is defined. Practically, $\text{std}_{\text{weight}}$ may sometimes equal to zero. In this case $f \cdot \text{mean}_{\text{weight}}$ is used to define '*significantly larger than*', where f is a magnifying factor. Eq. 2 integrates both aspects based on their maximum. We used $f = 2$, $n = 3$ in our experiments. Details of the parametrization can be found in [32, 33].

The perceptual clusters are formed by pruning all identified inconsistent edges (Fig. 3). The use of inconsistent edges enables reasonable inter-cluster edges to be identified, which is more adaptive than using a fixed threshold. For example, all edges in group O are identified as intra-cluster edges even if all of them are longer than the inconsistent edge $e1$ in Fig. 3b. If we used a fixed threshold instead, e.g., $T =$

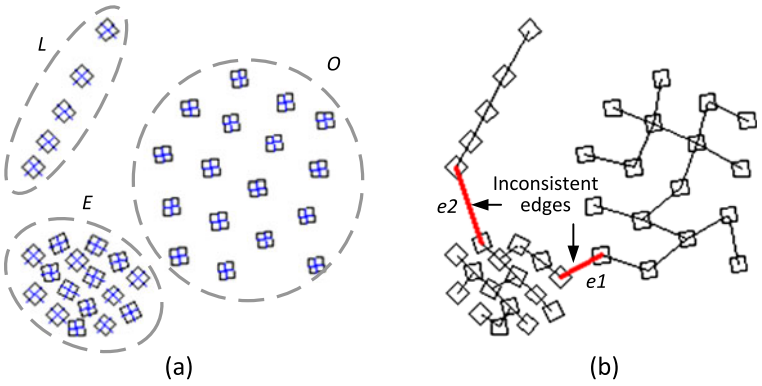


Fig. 3 Perceptual groups E, L and O (**a**) and their MST and inconsistent edges (**b**)

Algorithm 1 A sketch algorithm to trace linear paths from pruned MST

Input: pruned MST; *rules* for recognizing specific alignments
Output: a collection of linear paths ($LPColl$)
for all vertex v_0 of degree 1 or of degree 3 in MST **do**
  $v_1 \leftarrow$ adjacent vertex of v_0
  initialize a linear path (LP): $LP \leftarrow edge(v_0, v_1)$
  repeat {add a new vertex each loop to LP if possible}
    find a vertex $v_2 \leftarrow$ adjacent vertex of v_1 and $v_2 \neq v_0$, so that $edge(v_1, v_2)$ satisfies *rules* w.r.t. LP
    if v_2 exists **then**
      add $edge(v_1, v_2)$ to LP
      $v_0 \leftarrow v_1$; $v_1 \leftarrow v_2$; $v_2 \leftarrow NULL$
    else
      add LP to $LPColl$ if LP contains more than 3 vertices
      reset LP: $LP \leftarrow edge(v_1, v_2)$
      $v_0 \leftarrow v_1$; $v_1 \leftarrow v_2$; $v_2 \leftarrow NULL$
    end if
  until v_1 is of degree 1
  add LP to $LPColl$ if LP contains more than 3 vertices
end for
Post-processing: search-for-connection and search-for-extension

$Length(e1)$, all edges in group O would have been identified as inter-cluster edges, which would lead to an unacceptable clustering for this case.

4.3 Step 3—tracing algorithm

There is no guarantee that collinear and curvilinear alignments will be found after pruning inconsistent edges. In fact, the result is only a mixture of nonlinear and linear clusters that are neither significant nor regular. Therefore, the pruned MST serves as a starting point for all of our recognition algorithms. This section sketches out an algorithm template (Algorithm 1), designed for tracing linear paths (i.e. a sequence of vertices that are consecutively connected by MST edges) from pruned MST. Different rules for recognizing collinear and curvilinear alignments are deployed as parameters (Sections 4.4.1–4.4.3).

Algorithm 1 starts from vertices of degree 1 or 3 and traverses the pruned MST in the following way. For each new edge encountered in the tracing iteration, the algorithm determines if the edge is coherent with the existing alignment concerning the recognition rules described in Section 4.4. The algorithm may find a number of edges connected with current vertex v_1, and the edge that best satisfies the rules is added to the alignment. If no edge satisfies the rules, the algorithm goes to the next edge in the searching direction (if possible) and restarts a new tracing.

After the tracing procedure, the initially detected alignments may not be perfect (e.g. Fig. 4). Search-for-connection and search-for-extension are required to post-process and refine the initial result. First, search-for-connection is needed in the case of a short distance between end vertices of two detected alignments (e.g. the two curvilinear alignments in Fig. 4). This search can be done by connecting the two

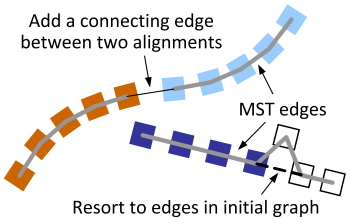


Fig. 4 Post-processing of initially detected alignments

end vertices of both alignments, if the connecting edge does not violate any of the recognition rules used. Second, search-for-extension is required because MST only connects locally shortest edges rather than visually desired edges. In Fig. 4, the two white buildings on the bottom right should have been grouped into the alignment of the four dark blue buildings. However, MST does not connect the two white buildings to the four because of the intermediate non-aligned building. Consequently, the tracing procedure stops at the sharp turn. To solve this problem, we then extend each traced alignment at its end vertices and search for adjacent edges in proximity graph instead of MST. If an edge is found that best satisfies the required rules, the edge (and its connecting buildings) is added to the alignment.

4.4 Step 4—rules for the recognition of linear alignments

Section 4.4.1 addresses the recognition rules for collinear alignment, after which a technique for detecting one-side alignments is presented (Section 4.4.2). Section 4.4.3 formalizes a perceptual model (a set of different rules) to detect curvilinear alignments.

4.4.1 Collinear alignments

A salient feature of collinear patterns is that buildings are aligned as if they form a straight line. The straightness is controlled by path angle rule in the tracing process described in Section 4.3. Path angle at vertex v_i is the angle formed by $vector(v_{i-1}, v_i)$ and $vector(v_i, v_{i+1})$ in the path (Fig. 5b).

The detection of the collinear pattern requires proximity, path angle, orientation and size similarity rules. Shape similarity is ignored because after studying topographic maps, we found that shape is not a dominant feature in determining a collinear alignment. First, most buildings have similar man-made shapes. Second,

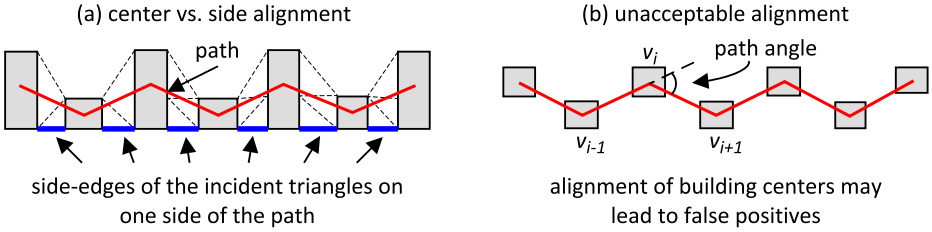


Fig. 5 Alignment-of-center vs. alignment-of-side technique

buildings with complex shapes are usually much bigger than the ordinary ones, so complex buildings can always be identified by size similarity rule. Nevertheless, shape property is still used for characterizing the detected patterns (see Section 4.5).

Proximity rule is again based on the notion of inconsistent edge introduced in Section 4.2. That is, if the weight of a new edge encountered in Algorithm 1 is inconsistent with respect to the existing alignment, the edge is regarded to be not part of this alignment. **Path angle rule** requires that angular deviations between the new edge and all edges in the existing alignment should be within certain threshold ($PathAngleDev$). This rule suppresses accumulative angular errors and ensures a straight-line-like alignment. **Orientation similarity rule** ($OrientationDev$) ensures that the angular deviation between any two adjacent buildings does not exceed 45°, as according to Duchêne et al. [9]. **Size similarity rule**, then, prevents the size contrast ($SizeContrast$) between any two adjacent buildings in the alignment from being too large (i.e. bigger building/smaller building $<$ $SizeContrast$).

4.4.2 One-side alignments

Some collinear alignments (see Fig. 5a), which can be widely found in real topographic data and are also perceptually significant, cannot be detected by the above model without an adaptation of parameters. This is mainly due to the varying size of adjacent buildings (Fig. 5a) that increases the path angles, which leads to a violation of the path angle rule.

To detect such patterns, we use alignment-of-side instead of alignment-of-center technique to measure the alignment of buildings. The alignment-of-side technique is facilitated by the incident triangles stored after the construction of CDT (e.g. the dashed triangles in Fig. 5a). To implement this technique, new path angle is calculated as the vector angle between adjacent side-edges of the incident triangles (e.g. the bold blue lines in Fig. 5a). Note that the new path angle can be measured on two sides of an alignment. If this new path angle rule, proximity and orientation rules are satisfied on at least one side, the group is recognized as a one-side alignment.

This alternative better resembles the way how human eyes perceive one-side alignments. In particular, it is more robust in detecting one-side alignments than arbitrarily adjusting $PathAngleDev$, as the latter will increase the chance of false positives. For instance, using alignment-of-center technique, the one-side alignment in Fig. 5a can be detected by increasing $PathAngleDev$. However, this will identify the group in Fig. 5b as a collinear alignment, which is unacceptable. Note that the alignment-of-side technique can also be applied for curvilinear alignments. This technique does not address size similarity rule as it is not required for one-side alignments (e.g. Fig. 5a). A mechanism for combining alignment-of-center and alignment-of-side techniques is presented in Section 4.6.

4.4.3 Curvilinear alignments

Curvilinear building alignments are patterns with gradual variations. Empirically, paths formed by such alignments gradually alter their heading angles and orientations of composing buildings, while keeping other properties such as spacing and size as similar as possible. Because most of the meandering features represented in geo-spatial data are too complex to be described by mathematical functions, we use the theory of 'association field' [10] to formulate rules for recognizing curvilinear

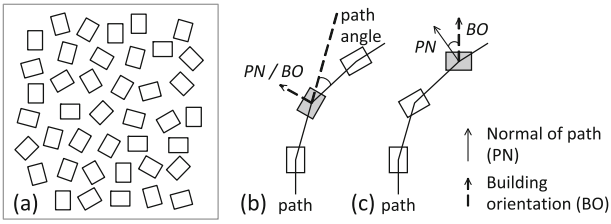


Fig. 6 Rules for human eyes to recognize curvilinear patterns - adopted from [5]

alignments. This theory of visual perception suggests a set of localized linking rules that explains why and how curved paths can be detected by human observers (Fig. 6). Contemporary researchers in visual perception also acknowledged that these rules are more concrete and realizable than Gestalt principles [5, p. 159].

In general, the psychological experiment by Field et al. [10] suggests that the human perception of curved or undulated paths formed by adjacent visual stimulus (e.g. Fig. 6a) is reflected by good continuity (i.e. smooth connection). More specifically, it depends on two criteria. These are the path angle and the degree to which the path elements (e.g. composing buildings in our case) are aligned or misaligned along the path. Take Fig. 6 for example, if the normal vector of the path (PN) coincides with the building orientation (BO), the building is perfectly aligned (the shaded building in Fig. 6b). If PN and BO are far apart, the building is misaligned (the shaded building in Fig. 6c). Field et al. [10] show further that the allowed maximum path angle (*MaxPathAngle*) is 40°–60°, provided that the elements are aligned along the path; misalignment by ±15° reduces the performance of detecting such paths (see also [5]). These rules explain why human visual system can see a 'snake' in Fig. 6a.

We formalize the above rules as follows. First, $MaxPathAngle \in [40°, 60°]$ was selected for path angle rule. It means that the path can be regarded as smooth if its path angle is smaller than *MaxPathAngle*. Second, the misalignment angle (*MisalignAngle*) for a building is the allowed deviation between its BO and the PN at this building. *MisalignAngle* is a dependent variable of the path angle at the building. The relationship between *MisalignAngle* and the path angle is that *MisalignAngle* is relatively relaxed (=15°) when the path angle is relatively small (≤15°), and *MisalignAngle* then decreases with the increase of the path angle, and it ends with 0 when the path angle reaches *MaxPathAngle* (Fig. 7). If the calculated misalignment angle of a building exceeds its *MisalignAngle*, the building is regarded as a misaligned building.

The detection of curvilinear building patterns is based on Algorithm 1 specialized by the above formalized rules, along with proximity and size similarity rules. The

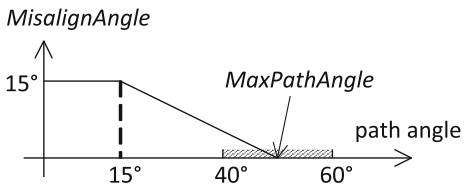


Fig. 7 The function describing the relationship between *MisalignAngle* and calculated path angle

alignment-of-side technique as presented in Section 4.4.1 is also applicable for curvilinear patterns.

4.5 Step 5—characterization of recognized building patterns

According to the proposed definition of building patterns in Section 3.2, the detected patterns can be generally characterized using the notion of homogeneity. Homogeneity describes how much regular the interested characteristics of the patterns are. Here, we consider homogeneities on four characteristics (Adding new ones is also possible): spacing, calculated between two proximate buildings using nearest distance; size, computed by area metric; orientation, implemented by wall statistical weighting method [9] which is well fitted for building orientations (blue crosses in Fig. 3a); and shape, measured by the complexity of building shapes: $ShapeIndex = Perimeter/(2 \cdot (\pi \cdot Area)^{1/2})$ [21]. Equation 3 formally defines the homogeneities of these properties:

$$Homo(C_i) = \begin{cases} STD(C_i)/Mean(C_i), & C_i \in \{spacing, size, shape\} \\ STD(C_i)^*/NFactor, & C_i \in \{orientation\} \end{cases}, \quad (3)$$

where $C_i = \{c_1^i, \ldots, c_p^i\}$ is a vector of characteristic values measured from a group of buildings $\{b_1, \ldots, b_p\}$. The homogeneities of spacing, size, and shape are based on the notion of *coefficient of variance* (i.e. STD/Mean). The denominator serves not only as a normalization factor, but also it takes the relative regularity into account. For example, if two groups have the same STD, the group with a larger mean is regarded more regular, and vise verse. However, this is not a suitable concept for the homogeneity of orientation because orientation is a cyclic variable. To normalize the homogeneity of orientation, we divide STD of orientations by a factor (*NFactor*). Here we choose $NFactor = 45°$, because the max angular deviation between two regular buildings is 45° [9].

Note that, STD of building orientations (BO) in Eq. 3 has different interpretations depending on the type of patterns to be recognized. For collinear alignments, it has a standard calculation: $STD = ((BO_i - Mean(BO))^2/N)^{1/2}$. However, STD requires a different interpretation for curvilinear alignments. In essence, STD describes the degree of deviation from mean. To better describe the homogeneity of orientation for a curvilinear alignment, normal of path (PN) is more appropriate as the local mean for each BO, instead of the real mean of BO (Fig. 8). In particular, when all BO align themselves fully with their local PN, the buildings are regarded to be perfectly aligned in terms of orientation. As a result, STD of orientation (starred in Eq. 3) for curvilinear alignments is better calculated by the formula: $STD = ((BO_i - PN_i)^2/N)^{1/2}$.

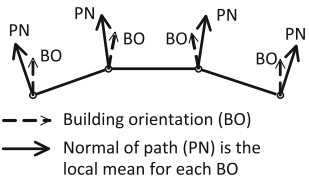


Fig. 8 Using local PN instead of the mean of BO to compute the homogeneity of orientation for curvilinear alignments

Finally, component homogeneities are integrated into a single value using Eq. 4:

$$InteHomo = \sum w_i \cdot Homo(C_i), \quad C_i \in \{spacing, size, shape, orientation\} \ , \quad (4)$$

where w_i is the weight of component homogeneities. A lower homogeneity value indicates a visually stronger pattern. In addition, an alignment consisting of more buildings is also regarded stronger.

Homogeneity is a generic way to describe the regularity of all pattern types and it is useful in the subsequent step (Section 4.6). Next we will introduce an extended characteristic - curvature descriptor. With this descriptor, it is possible for the system to infer more specific forms of the alignments, e.g., arc-like, circle-like.

A suitable estimation of local curvatures for an alignment would be: $LocalCurvature(v_i) = PathAngle(v_i)/Distance(v_{i-1}, v_{i+1})$. The denominator is more or less the same for a given alignment and can be considered as a constant, therefore we use $LocalCurvature(v_i) = PathAngle(v_i)$ to estimate the curvatures. In addition, this new estimation is easier to interpret. The forms of alignments can be analyzed based on the distribution of local curvatures estimated for them (Fig. 9).

Figure 9 illustrates three typical forms. They are alignments with constant curvature (c1, c2), with monotone curvature (m) and with wavy curvature (w). Specifically, if the curvatures of an alignment almost constantly distribute around the zero (c2 in Fig. 9), it has a straight-line-like shape. If the curvatures constantly distribute with a considerable distance to the horizontal axis (i.e. zero), the alignment has an arc-like shape. In this case if the head and tail of the alignment are close to each other, it has a circle-like shape (c1 in Fig. 9). Wavy alignments have the property that its curvatures go from positive to negative part of the curvature space or in an opposite direction; crossing points at horizontal axis identify inflection points in the alignments (e.g. w in Fig. 9). Alignments with Monotone increase or decrease in curvatures without changing signs are only theoretical (e.g. the spiral in Fig. 9). Because the curvature descriptor is not directly useful in the detection of building patterns, we will discuss some results in Section 6.5.

4.6 Step 6—combining conflicting patterns

As shown in Section 5.2, the algorithms (and their alignment-of-side variations) described above can produce reasonable results, but the results from these algorithms are not in itself complete. For instance, although the curvilinear algorithm is able to recognize some collinear alignments, it does not recognize all of them successfully. In other cases, a group of buildings may at the same time be recognized as e.g. a collinear and a curvilinear alignment. As a result, those potentially conflicting

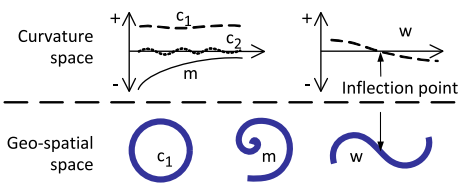


Fig. 9 Hypothetical distributions of different forms of alignments

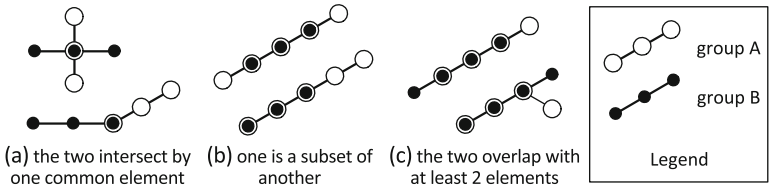


Fig. 10 Relationships between two potentially conflicting building patterns

patterns have to be combined and the most probable one should be selected for a better and more complete result.

The basic criteria for combining conflicting alignments are the number of elements (*NumOfElem*) contained in patterns and their homogeneity values (i.e. *InteHomo* in Eq. 4). Specifically, the more elements a pattern contains the better it is, and the lower the homogeneity value the stronger the pattern is. According to these two principles, we formulate a more detailed scheme to combine two potentially conflicting patterns (suppose groups A and B in Fig. 10 are patterns recognized by different algorithms, e.g., collinear and curvilinear ones):

1. If two groups A and B intersect by one common element (e.g. Fig. 10a), rather than treating them as combined patterns, we keep both patterns as individual ones; otherwise the combination can yield rather complex and irregular groups;
2. If one group B is the subset of another group A (e.g. Fig. 10b), group A is selected as the final result. In a special case where A and B are exactly the same but they are assigned to different pattern types (by different algorithms), the one with lower *InteHomo* is selected;
3. If two groups A and B have at least 2 elements in common (e.g. Fig. 10c), we consider the following criterion: $InteHomo/NumOfElem$. The group with a lower value for this criterion is selected;
4. For cases where the patterns are detected by only one of the algorithms, they are added to the final result set.

The combining process should be applied in two stages in the framework. The first stage stays within each algorithm, by evaluating the results detected by the alignment-of-center technique against those by the alignment-of-side one. This stage produces a harmonized result for each pattern type. The second stage is to evaluate and combine the harmonized results between different pattern types (e.g. collinear and curvilinear), which leads to a final result. Examples are presented in Section 5.2 and discussed in Section 6.2.

5 Experiments

5.1 Implementation

The proposed framework and algorithms were implemented as extensions to *GenTool*—an interactive generalization platform that has been implemented and maintained in Visual C++ 6.0 by our colleagues at Wuhan University since 1998.

(a) Refined CDT

(b) Proximity graph

(c) Pruned MST

Fig. 11 Computation of refined CDT, proximity graph and pruned MST on buildings and roads

The platform was applied in a national wide project in China to generalize and update topographic databases (from 1:10k to 1:50k). Among others, *GenTool* implemented an incremental Delaunay triangulation algorithm, based on which we implemented refined CDT, proximity graph, MST and inconsistent edge concepts (e.g. Fig. 11). Additionally, it implemented four recognition algorithms, including two algorithms for collinear alignments using alignment-of-center (*CollAlgo1*) and alignment-of-side (*CollAlgo2*), as well as two algorithms for curvilinear alignments using alignment-of-center (*CurvAlgo1*) and alignment-of-side (*CurvAlgo2*).

5.2 Experiments and results

We tested our approach against four topographic datasets of different geographic areas and scales. Dataset1 (scale 1:10k), Dataset2 and Dataset3 (scale 1:50k) are part of Dutch topographic datasets (Kadaster, the Netherlands). Dataset4 is part of ICC (Catalonia, Spain) topographic datasets at 1:25k. Figure 12 demonstrates

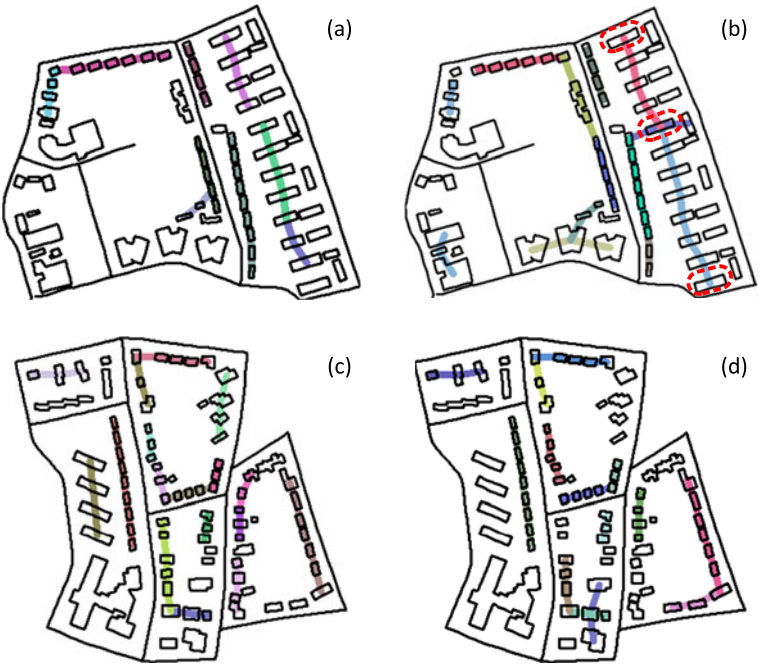


Fig. 12 Results obtained by *CollAlgo1* (**a**), *CollAlgo2* (**b**), *CollAlgo1* (**c**) and *CurvAlgo1* (**d**) (Dataset1: Kadaster, NL)

a general impression of some algorithms. Four experiments were carried out to show the result of parametrization, application of the framework, recognition quality and computational efficiency. Note that in the resulting figures black thin lines (if applicable) denote MST edges and (randomly) colored bold lines denote recognized patterns.

5.2.1 Adaptation of parameters

To better demonstrate the algorithms, different parameter values were tested to show their influence on the recognition results. We found that the most dominant parameters are $PathAngleDev$ for collinear pattern and $MaxPathAngle$ for curvilinear pattern. First, $PathAngleDev = \{5°, 10°, 15°, 20°, 25°, 30°\}$ were used to detect collinear alignments in Dataset2 (Fig. 13).

Figure 13 shows that the best working range of $PathAngleDev$ for collinear algorithms was from 15° to 20°. When using a value lower than 10°, some alignments were missed (false negatives, blue circles in Fig. 13). When using a threshold higher than 25°, some false positives were created (red circles in Fig. 13). The tests on the other datasets confirmed this working range of $PathAngleDev$.

Likewise, the $MaxPathAngle = \{40°, 60°\}$ was used for curvilinear patterns. The two values for our test cases did not show any difference, except for the alignment highlighted by the red circle on the bottom of Fig. 15b which can only be detected

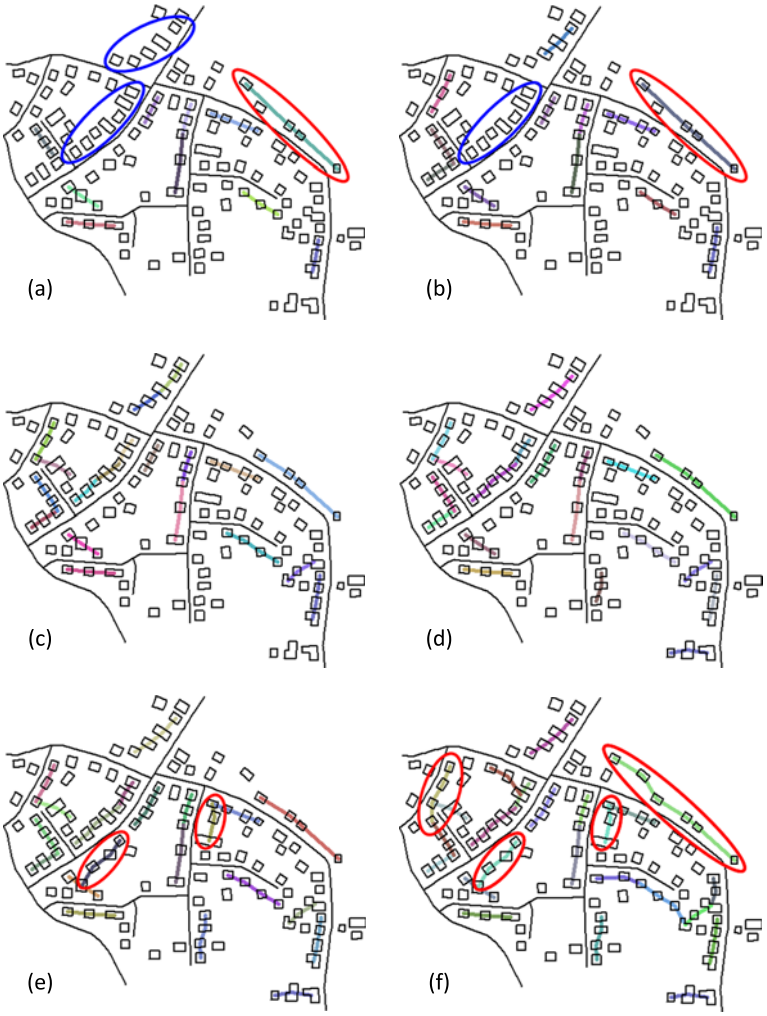


Fig. 13 Result of *CollAlgo1* with different path angle values: **a** $PathAngleDev = 5°$, **b** 10°, **c** 15°, **d** 20°, **e** 25°, **f** 30°; obvious false positives (*red circles*) and false negatives (*blue circles*) are exemplified (Dataset2: Kadaster, NL)

with the value of 60°. Consequently $MaxPathAngle = 60°$ is preferred since it is more tolerable to alignments with bigger curvatures.

5.2.2 Application of the framework

To demonstrate the framework, including the combination step (Section 4.6), the following parameters were used for all the datasets: $PathAngleDev = 18°$, $OrientationDev = 45°$ (for collinear), $MaxPathAngle = 60°$ (for curvilinear), and $SizeContrast = 3.2$ (used for alignment-of-center technique only). The component homogeneities were equally weighted to calculate the integrated homogeneity (Eq. 4). Figures 14, 15 and 16 show intermediate and final combined results for our datasets.

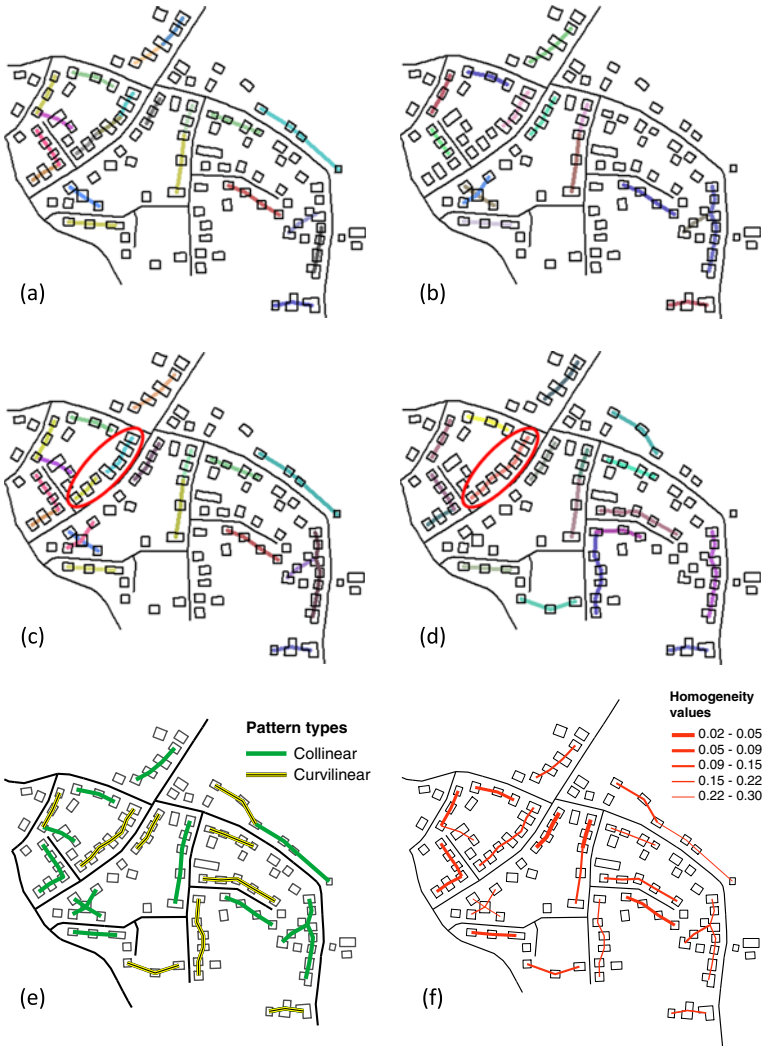


Fig. 14 Combination of different results: **a** *CollAlgo1*; **b** *CollAlgo2*; **c** harmonized result for **a** and **b**; **d** harmonized result for *CurvAlgo1* and *CurvAlgo2*; the final combined result visualized by pattern types (**e**) and homogeneity values (**f**); (Dataset2: Kadaster, NL)

The first observation in these figures is that, in most situations, all individual algorithms were able to detect alignments that are visually significant. Still, some alignments were not fully recognized by one and other algorithms. It was obvious that collinear algorithms segmented long curves (with apparent variations in curvature) into small pieces of straight lines. Not surprisingly, such long curves were successfully recognized by curvilinear algorithms. For example, the long curves (highlighted by red circles in Figs. 14d and 15b) were detected by curvilinear algorithms, but they were segmented by collinear algorithms (see their counterparts in Figs. 14c and 15a). An extreme case is demonstrated in Figs. 16b–d, where most curves were segmented

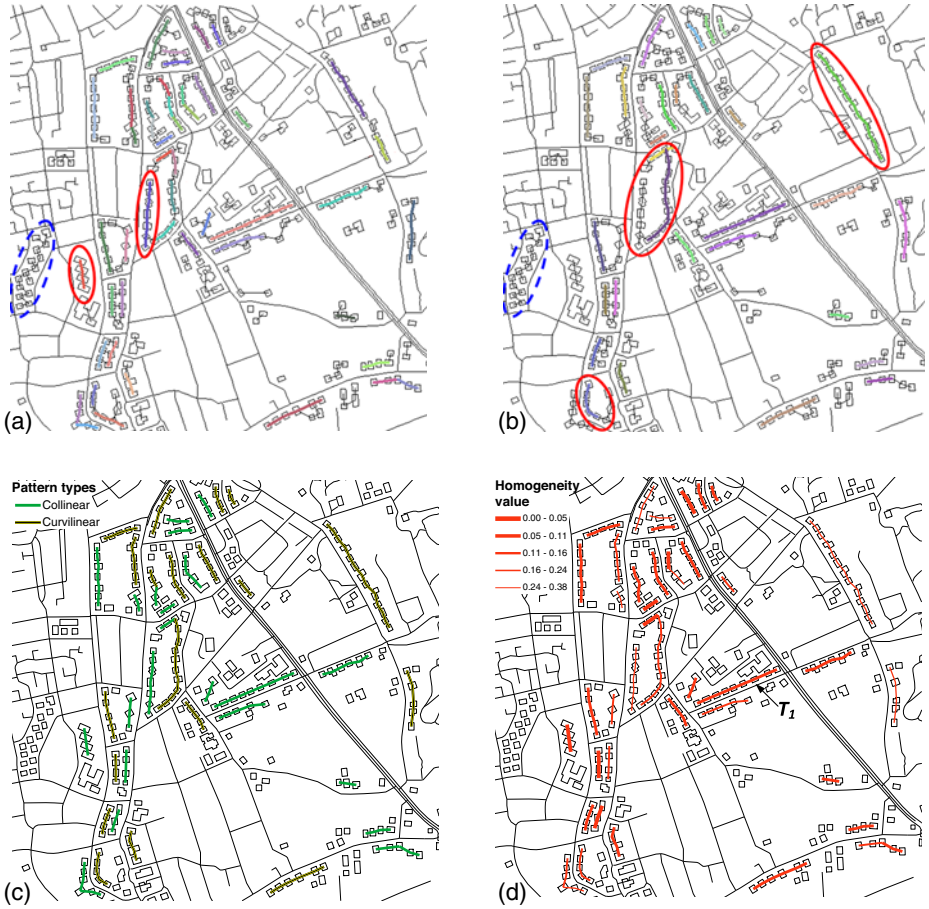


Fig. 15 Combination of different results: **a** harmonized result for *CollAlgo1* and *CollAlgo2*; **b** harmonized result for *CurvAlgo1* and *CurvAlgo2*; the final combined result visualized by pattern types (**c**) and homogeneity values (**d**); (Dataset3: Kadaster, NL)

by collinear algorithms. On the other hand, although curvilinear algorithms were also able to detect part of the collinear alignments, they did not recognize all of them. For instance, collinear algorithms recognized the alignments (highlighted by red circles in Figs. 15a and 16d) while curvilinear algorithms did not, as can be seen from their counterparts in Figs. 15b and 16b–c.

The difference between algorithms working with alignment-of-center and alignment-of-side techniques is obvious. Examples refer to the differences between e.g. Fig. 12a and b, and between Fig. 14a and b. In particular, the areas highlighted in Fig. 16b identify potential longer curves (curvilinear alignments). These alignments were not fully recognized using alignment-of-center, but they were recognized using alignment-of-side technique (cf. Fig. 16c).

Last but not least, the experiment shows that the final results combined from different algorithms were more complete than the individual results. In fact, they are unions of the individual results. Further, Figs. 14e, 15c and 16e show that the

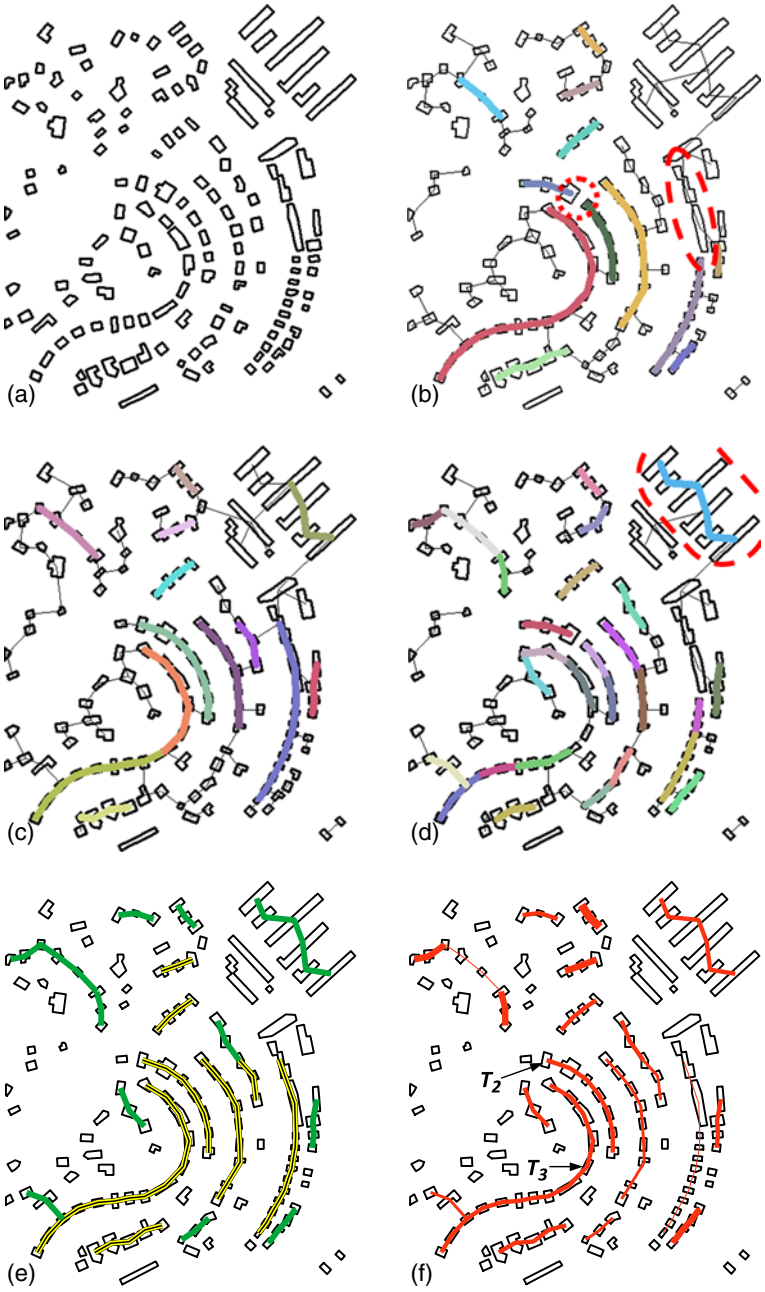


Fig. 16 Combination of different results: **a** Dataset4 (ICC, Catalonia, Spain); **b** *CurvAlgo1*; **c** *CurvAlgo2*; **d** harmonized result for *CollAlgo1* and *CollAlgo2*; the final combined result visualized by pattern types (**e**) and homogeneity values (**f**)

datasets are mixtures of different types of alignments. Figures 14f, 15d and 16f show the homogeneities calculated for the final patterns (the wider the symbol the stronger the pattern is in terms of homogeneity).

5.2.3 Validation of the recognition results

To validate the quality of the recognition results, an expert evaluation was carried out. In this survey, an independent cartographer (from Kadaster, the Netherlands) visually identified the patterns in representative excerpts of the datasets used in this research (Fig. 17).

The survey was presented in a Word document. The invited expert used the drawing tools available in MS Word to 'draw' the patterns he regards to be significant. Note that the expert only used a straight line tool to mark all pattern types, with colors indicating different types of patterns.

We validate the recognition results by comparing the automatically detected alignments (ADAs) with the visually identified alignments (VIAs) and considering the following three side notes. First, for all the three datasets, the expert identified isolated and pairs of buildings as meaningful patterns, but by definition the building patterns in our approach contain at least three buildings. Consequently, we ignored all the VIAs that only contain one or two buildings. Second, since this research focuses on linear alignments, we also ignored grid-like and unstructured patterns. Finally, in the expert's view, curvilinear and align-along-road alignments are variations of collinear alignments. As a result, linear alignments are directly compared regardless of their respective types.

The comparison approach works as follows: if an ADA partly coincide with a VIA (i.e. the two have common features), we count it as a true positive (tp); if an ADA occurs without a (partly) coincident VIA, we count it as a false positive (fp); if a VIA occurs without a (partly) coincident ADA, we count it as a false negative (fn).

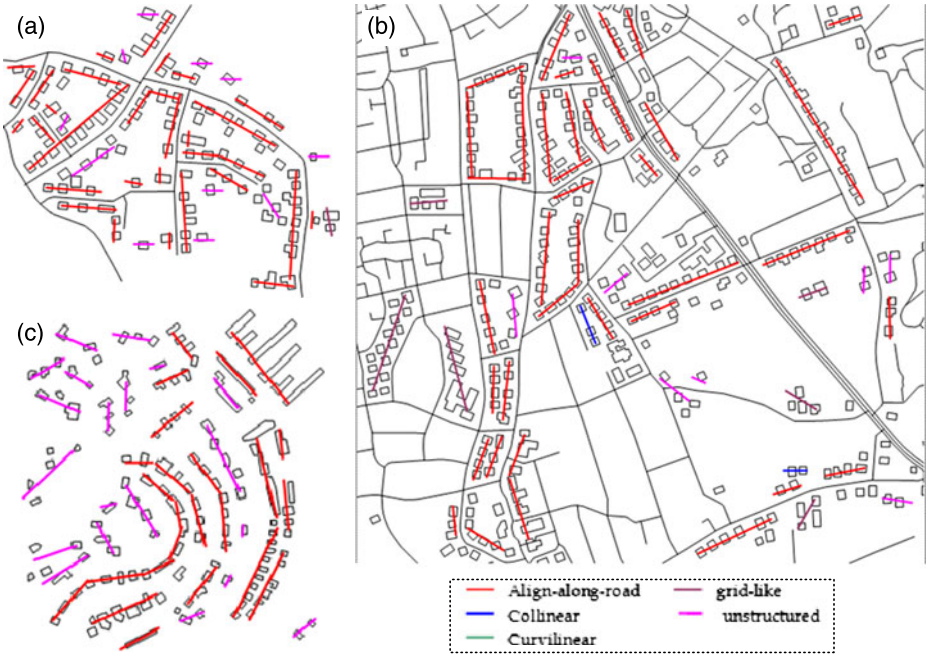


Fig. 17 Building patterns visually identified by a cartographer (Dataset2-4)

Table 1 Precision and recall of automatically recognized alignments with respect to visually identified alignments (Dataset2-4)

	Precision	Recall
Dataset2	73.9%	85.9%
Dataset3	77.3%	91.9%
Dataset4	60.0%	100.0%

Then we use $precision = tp/(tp + fp)$ and $recall = tp/(tp + fn)$ to characterize the recognition quality (Table 1).

From the comparison result we can conclude that the recognition quality is in general satisfactory. The recall for the three datasets is high (approx. 90% and higher). This indicates that the automatic approach recognized almost all the VIAs. The relatively low precision (lower than 80%) indicates that the automatic approach recognized more alignments than the expert. That the expert chose more rigorous rules in determining the alignments could be one possible reason. The ambiguities in defining and distinguishing the patterns could be another, which we will discuss in Section 6.3.

5.2.4 Computational efficiency

To give an idea of the computational time the framework takes, we tested Dataset3 (consisting of 391 roads and 333 buildings; 44 linear alignments were detected) on two strategies. The first strategy processed the data set as a whole and the second strategy applied a divide-and-conquer approach (buildings divided into groups by roads). The test was carried out on a HP laptop (CPU 2.00GHz). The time taken (s) is shown in Table 2.

Generally, the performance is rather practical: it took 2.674 s to recognize alignments from Dataset3 without using a divide-and-conquer strategy. It shows further that Step 1—triangulation and deriving of proximity graph took the majority of the running time (2.063 s). The four recognition algorithms (Steps 2–5) were fairly efficient: together they took 0.249 s. Step 6 took 0.362 s to combine conflicting alignments from different algorithms.

The whole process greatly accelerated (in total 1.012 s) when applying a divide-and-conquer optimization. In particular, Steps 1 and 6 benefit most from this: the two steps took 0.672 s and 0.096 s, respectively. Note that this optimization did not

Table 2 Performance (time taken (s)) of our framework and its sub-steps on Dataset3 using two strategies: processed as a whole and divide-and-conquer

		Processed as a whole				Divid-and-conquer			
Time taken (s)	Step 1	2.063				0.672			
	Steps 2–5	*Coll-Algo1*	*Coll-Algo2*	*Curv-Algo1*	*Curv-Algo2*	*Coll-Algo1*	*Coll-Algo2*	*Curv-Algo1*	*Curv-Algo2*
		0.015	0.031	0.062	0.141	0.016	0.023	0.078	0.125
		0.249 (in total)				0.242 (in total)			
	Step 6	0.362				0.096			
	Total	2.674				1.012			

improve the performance of the individual algorithms, because the algorithms work on the MST structure that is segmented into small groups by roads no matter the optimization is used or not (cf. Fig. 11: no partitioning before triangulation).

6 Discussion

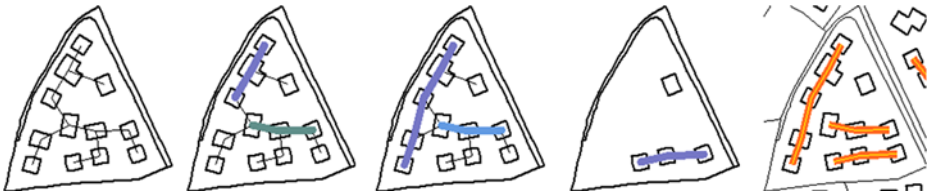
This discussion consists of two parts. Sections 6.1–6.3 discuss the results presented in Section 5. Sections 6.4 and 6.5 present current limitations and suggestions for further improvement.

6.1 Intermediate steps that improve the recognition quality

This section explains several intermediate steps in the framework (Section 4) that aim to improve the recognition quality. Take *CollAlgo1* for example, Fig. 18b shows the detected collinear alignments without post-processing (Section 4.3) after the algorithm has been executed once. By applying the post-processing (e.g. search-for-extension) the result was improved (a longer alignment was detected in Fig. 18c). Visually, the alignment on the bottom should have been recognized, but it was not detected due to the original form of MST structure (Fig. 18a). As described in Section 4.1, the framework automatically records the detected alignments and then removes them from the proximity graph, after which MST is recalculated and the above-mentioned steps are re-executed. This iteration proceeds until all possible alignments are successfully recognized (e.g. Fig. 18d and e).

Usually, building blocks contain limited number of buildings and thus it reduces the times of the above-mentioned iteration. In our test datasets, the iteration stopped at second round only occasionally, and in most cases all the potential alignments were detected at first round of the iteration. Consequently, this iteration substantially improved the recognition quality without reducing too much computational efficiency (see Section 5.2.4).

Note also that some single objects in the removed alignments, usually vertices of degrees 1 and 3, may also be part of other patterns (see e.g. Fig. 12b). We decided not to keep those shared objects when removing detected patterns for the following reasons. First, the removal of patterns only occurs between iterations, while most alignments with shared objects were detected in the first iteration. Second, false positives would have been created if the shared objects were kept. For example, the three buildings marked in Fig. 12b would consequently have been recognized



(a) Pruned MST (b) Initial result (c) Extension (d) 2nd round (e) Final result

Fig. 18 Intermediate steps to improve the recognition quality (part of Dataset3)

as a new alignment which does not really exist. Still, it is true that some potential patterns may be missed if we do not keep those shared objects. To better address this problem without causing new problems, a possible improvement is discussed in Section 6.5. Similarly, a group of objects can be part of another group. Nevertheless, this does not contradict our approach. For instance, a collinear alignment detected using *CollAlgo1* may be part of a curvilinear alignment using *CurvAlgo2*. This part-of relation can be used to form building pattern hierarchies. Also, the two alignments can be combined into a harmonized result using the method described in Section 4.6.

6.2 Single algorithm versus multi-algorithm approach

The results show that the collinear and curvilinear algorithms and their variations (e.g. alignment-of-side technique) are able to recognize building patterns that are also contented with human perception. However, these individual algorithms are not competent to fully recognize building alignments. The results show merits and limitations of each algorithm and that different algorithms are partly complementary (Section 6.2.1). Therefore a framework that combines different algorithms will yield better results for building patterns recognition than the single algorithm approach, which can be concluded from several observations (Section 6.2.2). Finally, Section 6.2.3 discusses the efficiency of the two approaches.

6.2.1 Potentials and problems of the individual algorithms

The algorithms for collinear and curvilinear patterns are appropriate to recognize straight lines respectively curves. These individual algorithms refine the general clusters (clusters linked by black thin lines in Fig. 15a) detected by an approach similar to [23]. In the design phase, we made a sharp distinction between collinear and curvilinear patterns. It is thus reasonable that the collinear algorithms fail to recognize curvilinear alignments. On the other hand, straight lines can be regarded as a special case of curves, so the curvilinear algorithms are expected to be able to find collinear alignments. The experiments confirm that, this is true in many cases but not in all cases. For example, when the buildings align themselves to the same direction, and when this direction is quite different from the normal direction of the path (e.g. the red circles in Fig. 15a), curvilinear algorithms will identify them as misaligned buildings. However such alignments can be successfully detected by collinear algorithms, because they use a much looser rule on building orientations. Given this deficiency, this study shows that still the visual perception theory [10] used in the curvilinear algorithms is effective and yields satisfactory results, especially in recognizing long curvilinear patterns.

On the other hand, the results demonstrate that it is very difficult to use a single algorithm to detect all patterns. First of all, universal rules appropriate for detecting different pattern types are not available. Usually different ad-hoc models are used. Technically, it is also difficult to accommodate competing recognition rules (e.g. for collinear, curvilinear, alignment-of-center and alignment-of-side) in one algorithm. Considering the fact that the recognition results from different algorithms are partly complementary, using multiple algorithms followed by a combination process should yield better results. The experiments that combined the results of individual algorithms confirmed this (see Figs. 14, 15 and 16).

6.2.2 Advantages of using a multi-algorithm paradigm

The use of multiple algorithms followed by a combination process also improves the quality (in terms of homogeneity) of the detected patterns. This is because, in the combination process, two conflicting patterns are evaluated against each other, and the one with better quality (defined in terms of number of elements and homogeneity, see Section 4.6) is selected as the final result.

Admittedly, each specific algorithm may be well suited for a certain type of data. But real data are mostly mixtures of various characteristics as shown in the experiments, i.e., only in rare cases a dominant pattern type may character a dataset. Even if a dataset is rather homogeneous, with the single algorithm approach one still has to identify the main characteristics of the dataset and manually choose the most appropriate algorithm. The proposed multi-algorithm approach can deal better with these problems due to the evaluation and combination mechanism. Therefore, it is more adaptive to different types of data.

6.2.3 Efficiency considerations

The experiments show that combining the results of multiple algorithms reduces the performance compared with the single algorithm approach. The total running time of the multi-algorithm approach, however, is still acceptable. This is because the most time-consuming computation—DT (more than 60% of the whole processing for Dataset3) is in both cases required. Applying multiple algorithms and combining the results in a final procedure are computationally less expensive (less than 30%). It is therefore worthwhile to sacrifice a little performance for improved quality and completeness of the recognition. For time-crucial applications, a divide-and-conquer strategy that greatly improves the performance could be applied.

6.3 Ambiguities in defining and distinguishing building patterns

Several observations from the expert survey (Section 5.2.3) demonstrate further ambiguities outlined in Section 3.2. An example of the ambiguity in defining building patterns is how many buildings counts as a pattern, as described in Section 5.2.3. Besides, the groups that were identified by both the expert and the proposed approach differ slightly. The difference lies in whether a building belongs to a group or not (e.g. the red circles in Fig. 15a and their counterparts in Fig. 17b). The reason may be the ambiguity in the definition, which causes the use of different recognition rules and parameters.

The survey result also demonstrates ambiguities in distinguishing pattern types. Typically, the expert identified linear alignments as align-along-road patterns. The reason, also inferred from the expert survey, was the presence of either real or virtual roads (e.g. white spaces between alignments in Fig. 17c). The expert may therefore characterize linear alignments as unstructured clusters where he cannot perceive virtual roads. For instance, he identified obvious linear alignments as unstructured clusters in Fig. 17c, which explains why the precision for dataset4 is the lowest (60%). The above observation shows a different understanding of how to distinguish types of patterns. Additionally, the curvilinear alignments in Fig. 17c were segmented into consecutive straight lines and identified as align-along-road patterns by the expert.

This may be due to the limited drawing tools available, but the reason could also be that the boundary between linear types is ambiguous and subjective.

6.4 Limitations of current methodology

One of the drawbacks of our current approach is that it sometimes fails to recognize linear alignments in densely populated areas, such as city centers, commercial and industrial areas. In such areas the MST usually structures as trees with short branches, where linear paths are not clearly visible (see e.g. the group highlighted by the blue dashed circle in Fig. 15). Ideally, we would prefer to classify such building groups as nonlinear patterns, because linear characteristics are no longer significant there. As a result, future work is needed to complete our current framework by allowing for nonlinear clusters. A simple solution could be firstly removing the recognized linear alignments from the MST, and then applying e.g. the unstructured pattern detection algorithm described in [33] to recognize nonlinear clusters.

To perfect the recognition of building patterns, contextual knowledge should also be incorporated. Actually, different types of building patterns may appear in different places depending on spatial contexts. For instance, linear alignments can be found in urban (e.g. residential areas and partly town centers) and suburban areas; nonlinear clusters are dominant in city centers and industrial areas; random arrangements can be found in scattered and mountain villages [26, 27]. The ratio between the detected alignments and the general clusters may be used to partly imply the context of the built environment, although the determination of the context is a more complex inference, which depends also on size, shape and density of buildings [27].

The ambiguities and subjectivity involved in the expert survey lead to several problems in the evaluation of recognition quality. One problem is the low but problematic precision obtained in Section 5.2.3. Apart from the reason of ambiguities explained in Section 6.3, another reason is the way that the expert dealt with the buildings shared among patterns. In the survey, the expert sometimes precluded a building from an alignment if this building is already part of another alignment (cf. the two horizontal alignments in Fig. 18e and their counterparts in Fig. 17b). Because of this, the expert may end up with alignments consisting of only two buildings, which also contributes to the decrease in precision. To address this issue, the survey should be further improved by clearly explaining the concepts and terms used, providing more appropriate ways for experts to identify the patterns. In addition, inviting more experts from different organizations would enable to draw more persistent conclusions.

6.5 Further options and improvement

- User interaction: Intended users can interact and tune the framework at several stages. First, parameters such as $PathAngleDev$ and $MaxPathAngle$ can be tuned before the algorithms starts, to better control the results. Second, after the final result (combined from different algorithms) has been obtained, the users can select more homogeneous patterns by specifying a homogeneity threshold.
- Further enrichment: as described in Section 4.5, the curvature descriptor can be used to enrich the description of building alignments. This enrichment is

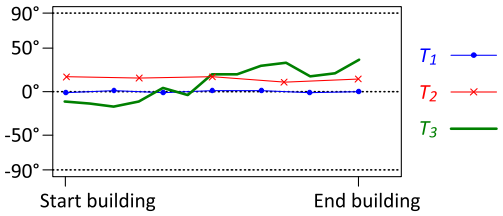


Fig. 19 Experimentally derived local curvatures for sample alignments

useful in automated generalization of building alignments for keeping their specific shapes. For illustration, the descriptor was applied to some detected linear alignments (*T1*, *T2* and *T3* marked in Figs. 15d and 16f). According to the curvature descriptor, we can identify alignments with straight-line shape (*T1*), constant curvatures (*T2*) and wavy curvatures (*T3*) in Fig. 19, which agrees with their respective spatial arrangements.

– Further improvement: As can be seen from Fig. 18, the MST structure may disconnect desired proximity relationship. Hence at times the tracing procedure has to resort to the edges of the proximity graph. In extreme cases, the tracing procedure starting from MST may fail to recognize building patterns (e.g. the stream of buildings next to the road on the right side of Fig. 12a). An improvement could be obtained by starting the tracing procedure on the proximity graph instead of MST. This has at least three advantages. First, the tracing procedure is able to detect those extreme case alignments as more options are available to it. Second, the search-for-connection and search-for-extension described in Section 4.3 are no longer needed. Third, the iteration explained in Section 6.1 is not necessary as all alignments can be detected at the first round of tracing. This last point can avoid the problem caused by removing those shared buildings. The potential downside of this improvement is that it would increase the searching space and hence the computational time. Besides, this improved approach may not be deterministic, because there are no fixed vertices for the tracing procedure to start from, and the tracing has to be adapted for two directional searching. In current development, the tracing starts from vertices of degree 1 and 3 and searches only in one direction. Therefore, further research is required to confirm the above merits in practice.

7 Conclusion

This paper presents a framework and four algorithms for recognizing collinear and curvilinear building alignments from topographic data. The algorithm for each type has two variations: alignment-of-center and alignment-of-side. As shown by the experiments, these individual algorithms have their own merits and limitations. Furthermore, this study shows that the proposed multiple algorithms approach improves the results detected by the single algorithm approach. Also the combination process is effective in finding better patterns between conflicting alternatives and in compiling a more complete result from different recognition results.

This paper also illustrates how the theories of visual perception can be integrated into and combined with different computational models, such as Delaunay

triangulation, MST and other geometric measurements (e.g. shape and orientation), in order to recognize building patterns. The building patterns recognized from topographic datasets proves to be also satisfactory for experts working with visual methods. Additionally, we identify the working ranges of different parameters in which the algorithms may probably yield satisfactory results. Still, one can achieve results that better fit his/her own situations by fine-tuning the parameters, or by adding or dropping recognition rules.

Although our framework is partially implemented (currently we concentrate on linear alignments), the idea underlying the framework, i.e., using multiple algorithms followed by an evaluation and combination process, is applicable to other types of building patterns. As the adopted perception theories are generic to geometric forms, our proposed approach can be adapted to the detection of patterns in other geographic features, such as groups of ponds, lakes and archipelago, as long as related recognition rules are modified according to their own characteristics.

Future work will complete the implementation by covering both linear and nonlinear patterns. In addition, arbitrary (i.e. random) arrangements and isolated buildings should also be identified to support automated generalization of e.g. mountain villages. The approach can be applied in many ways. For instance, the quality measure (i.e. homogeneity value) we derived for building patterns can be used to investigate the generalization of the patterns. These include enhancement of homogeneity, derivation of pattern hierarchies as well as aggregation, typification and simplification of patterns. Besides, the quality measure can be used for automated evaluation of whether and to what extent building patterns are preserved at scale transitions. For the recognized patterns, they may also be used to facilitate different cadastral and urban planning activities.

Acknowledgements We thank the national mapping agencies, Kadaster of the Netherlands and ICC of Spain, for providing data to this work. In addition we acknowledge Jan Bulder for carefully completing the expert survey. The three anonymous reviewers whose comments substantially improve the paper are gratefully acknowledged. This research is supported by the National High-Tech Research and Development Plan of China (Grant number: 2009AA121404). This research is also supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO) and partly funded by the Ministry of Economic Affairs, Agriculture and Innovation (Project number: 11300).

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Alexander C (2002) The phenomenon of life: nature of order, book 1: an essay on the art of building and the nature of the universe. The Center for Environmental Structure, Berkeley
2. Anders KH (2003) A hierarchical graph-clustering approach to find groups of objects. In: The 5th workshop on progress in automated map generalization, Paris. http://www.geo.unizh.ch/ICA/docs/paris2003/papers03.html
3. Anders KH (2006) Grid typification. In: Progress in spatial data handling, pp 633–642
4. Bader M, Barrault M, Weibel R (2005) Building displacement over a ductile truss. Int J Geogr Inf Sci 19(8):915–936
5. Bruce V, Green RP, Geogreson A (2003) Visual perception physiology, psychology, ecology, 4th edn. Psychology Press, New York

6. Burghardt D, Cecconi A (2007) Mesh simplification for building typification. Int J Geogr Inf Sci 21(3):283–298
7. Carmona M, Heath T, Oc T, Tiesdell S (2003) Public places, urban spaces: the dimensions of urban design. Architectural Press, Oxford
8. Christophe S, Ruas A (2002) Detecting building alignments for generalisation purposes. In: Richardson DE, van Oosterom P (eds) Advances in spatial data handling (SDH 2002). Springer, Berlin, pp 419–432
9. Duchêne C, Bard S, Barillot X, Ruas A, Trévisan J, Holzapfel F (2003) Quantitative and qualitative description of building orientation. In: 5th workshop on progress in automated map generalization. Pairs, p 10 p. http://aci.ign.fr/BDpubli/paris2003/papers/duchene_et_al_v1.pdf
10. Field TM, Hayes A, Hess RF (1993) Contour integration by the human visual system: evidence for a local "association field". Vis Res 33(2):173–193
11. Foerster T, Stoter J, Kraak MJ (2010) Challenges for automated generalisation at european mapping agencies: a qualitative and quantitative analysis. Cartogr J 47(1):41–54
12. Gersmehl PJ (1991) The language of maps. Pathway in geography series, title no. 1. National Council for Geographic Education. National Council for Geographic Education, 16-A Leonard Hall, Indiana University of Pennsylvania, Indiana, 207 p. ISBN: 0-9627379-3-3
13. Harrie L, Weibel R (2007) Modelling the overall process of generalisation. In: Mackaness W, Ruas A, Sarjakoski L (eds) Generalisation of geographic information: cartographic modelling and applications. Elsevier, Amsterdam, pp 67–87
14. Hough P (1959) Machine analysis of bubble chamber pictures. In: Proceedings of international conference on high energy accelerators and instrumentation, pp 554–556
15. Jones C, Bundy G, Ware J (1995) Map generalization with a triangulated data structure. Cartogr Geogr Inf Syst 22(4):317–331
16. Li Z, Yan H, Ai T, Chen J (2004) Automated building generalization based on urban morphology and gestalt theory. Int J Geogr Inf Sci 18(5):513–534
17. Lüscher P, Weibel R, Burghardt D (2009) Integrating ontological modelling and bayesian inference for pattern classification in topographic vector data. Comput Environ Urban Syst 33(5):363–374
18. Lynch K (1960) The image of the city. Publications of the Joint Center for Urban Studies. MIT Press, Cambridge
19. Maceachren AM, Ganter JH (1990) A pattern identification approach to cartographic visualization. Cartographica 27(2):64–81
20. Marshall S (2005) Streets & patterns. Spon Press, Taylor & Francis Group, New York
21. Peter B, Weibel R (1999) Using vector and raster-based techniques in categorical map generalization. In: Third ICA workshop on progress in automated map generalization, p 14 p. http://www.geo.unizh.ch/publications/beatp/ICA99_Working_Group_Paper.pdf
22. Regnauld N (1996) Recognition of building clusters for generalization. In: Kraak MJ, Molenaar M (eds) Advances in GIS research II (Proceedings of 6th SDH'96, Delft). Taylor & Francis, London, pp 4B.1–4B.14
23. Regnauld N (2001) Contextual building typification in automated map generalization. Algorithmica 30(2):312–333
24. Sester M (2000) Knowledge acquisition for the automatic interpretation of spatial data. Int J Geogr Inf Sci 14(1):1–24
25. Shekhar S, Zhang P, Huang Y, Vatsavai RR (2004) Trends in spatial data mining. In: Kargupta H, Joshi A, Sivakumar K, Yesha Y (eds) Data mining: next generation challenges and futhure directions, chap 3. AAAI Press, Menlo Park, pp 357–380
26. SSC (2005) Topographic maps—map graphics and generalization, SSC publication series, vol 17. SSC—Swiss Society of Cartography, Wabern, Switzerland. http://www.cartography.ch/publikationen/publications.html
27. Steiniger S, Lange T, Burghardt D, Weibel R (2008) An approach for the classification of urban building structures based on discriminant analysis techniques. Trans GIS 12(1):31–59
28. Stoter J (2010) State-of-the-art of automated generalisation in commercial software. 58, Official Publication—EuroSDR
29. Stoter J, Burghardt D, Duchêne C, Baella B, Bakker N, Blok C, Pla M, Regnauld N, Touya G, Schmid S (2009) Methodology for evaluating automated map generalization in commercial software. Comput Environ Urban Syst 33(5):311–324
30. Stoter J, van Smaalen J, Bakker N, Hardy P (2009) Specifying map requirements for automated generalization of topographic data. Cartogr J 46(3):214–227

31. Wertheimer M (1923) Laws of organization in perceptual forms. In: Ellis WD (ed) A source book of gestalt psychology. Routledge & Kegan Paul, London, pp 71–88
32. Zahn CT (1971) Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Trans Comput C-20(1):68–86
33. Zhang X, Ai T, Stoter J (2010) Characterization and detection of building patterns in cartographic data: two algorithms. In: Joint international conference on theory, data handling and modelling in geospatial information science (SDH' 2010), vol XXXVIII, part 2, pp 261–266

Xiang Zhang received his B.S. and M.Sc. degrees in GIS and cartography from Wuhan University, China, specialized in map generalization and multi-scale representations. Since 2008 he is a Ph.D. candidate in a joint programme between Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, the Netherlands and Wuhan University, China. His Ph.D. topic is automated evaluation of generalization output. His research interest includes multi-scale representation and modeling, knowledge formalization, and application of computational geometry and pattern recognition in automated map generalization and evaluation.

Tinghua Ai is a professor of cartography in School of Resource and Environmental Science, Wuhan University. His research interests include map generalization, spatial cognition and spatial analysis. He published several papers on the application of computational geometry in map generalization. He used to lead a team developing a generalization system which is now widely applied in map production lines in China; the system also played an important role in the engineering project of Chinese 1:50000 map updating.

Jantien Stoter (1971) defended her Ph.D. thesis on 3D Cadastre in 2004, for which she received the Prof. J.M. Tienstra Research-Award. From 2004 till 2009 she worked at the International Institute for Geo-Information Science and Earth Observation, ITC, Enschede, the Netherlands. As associate professor at ITC she led the research group in the field of automatic generalization. She was project leader of an EuroSDR project on generalisation from 2005 till 2009. Since October 2009, she fulfills a dual position: one as associate professor at Section GIS technology at OTB and one as Consultant Product and Process Innovation at the Kadaster. From both employers she is posted to Geonovum. The topics that she works on are 3D, information modeling and multi-scale data integration. Since January 2010 she leads the 3D pilot that aims at establishing a 3D reference model in The Netherlands in a collaboration of 55 partners. In November 2010 she received a VIDI grant, which is a prestigious award given by the Netherlands Organisation for Scientific Research (NWO) for excellent senior researchers.

Menno-Jan Kraak is professor in Geovisualization at the Faculty of Geoinformation Science and Earth Observation (ITC), University of Twente. He holds a PhD in Cartography of Delft Technical University. Since 2007 Menno-Jan is one of the vice-presidents of the International Cartographic Association, responsible for the science portfolio. He wrote many publications on cartography and GIS. His most visible publication is the book Cartography, visualization of geospatial data (with Ormeling) and published by Prentice Hall, (translated in five languages). He is a member of the editorial board of several international journals in the field of Cartography and GIScience.

Martien Molenaar has a doctors degree in geodesy of Delft Technical University. He was a senior lecturer at ITC (1973–1983) and a professor of GIS and Remote Sensing at Wageningen University (1983–1996). In 1996 he returned to the ITC as a full professor in Geoinformatics and Spatial Data Acquisition. From 2001 thru 2009 he was Rector of the ITC. He is president of the Netherlands Geodetic Commission. For the term 2008–2012 he is president of ISPRS technical Commission VI on Education and Outreach. Through his work he has been involved in international projects, consulting missions and he has been lecturing in many countries in Europe, Africa, Asia and Latin America. He wrote more than 200 scientific publications on geodesy, photogrammetry, spatial data modeling, remote sensing and GIS.