DELINEATION AND CONSTRUCTION OF 2D GEOMETRIES BY FREEHAND DRAWING AND GEOMETRIC REASONING

J. Meidow*, H. Hammer, L. Lucks

Fraunhofer IOSB, Ettlingen, Germany - firstname.lastname@iosb.fraunhofer.de

KEY WORDS: education technology, interaction, constraints, geometric reasoning, adjustment

ABSTRACT:

The creation of accurate and consistent line drawings is the subject of various applications. Prominent examples are the delineation of human-made objects in aerial images and the construction of technical line drawings, flow-charts, or diagrams. Interactive solutions usually restrict the user's interaction during the design process to enforce geometric relations such as orthogonality or incidence. To avoid the time-consuming selection of operational modes, a freehand approach is desirable using strokes as the only user input. In this case, the construction principles have to be inferred automatically by geometric reasoning with uncertain observations. We present and discuss the corresponding methods in the context of educational technology. By introducing and utilizing a user-friendly software tool, we offer a hands-on approach to explore the feasibility and usability of the procedure. The experiments comprise the polygonal approximation of 2D shapes, theorem proving, and the construction of human-made figures.

1. INTRODUCTION

1.1 Motivation

Human-made objects and their models predominantly feature geometric relationships such as parallelism or orthogonality. Examples are residential buildings, diagrams, or technical line drawings. For many technical applications, these constructions must feature topological consistency, completeness, and integrity (Mäntylä, 1987). Thus, these properties are usually enforced during the design process by restricting the user's interactions.

As an example, Figure 1 shows on the left side the construction of an orthocenter. The orthocenter is the intersection of the three altitudes in a triangle, which in turn are given by the perpendiculars to the bases passing through the vertices. Thus, in total, we have seven geometric constraints: three pairs of straight line segments forming right angles, and four triplets of segments intersecting in a point. Given six of these constraints, the seventh constraint can be deduced.

During ruler-and-compass construction, the user has to select the modes 'add a line,' 'add a perpendicular,' and 'add a point' followed by the typical press-drag-release sequence for computer mice. The interactive selection of geometric entities is realized by snapping to nearby entities, i.e., by checking if the mouse is over an existing geometric element. Thus, time-consuming interaction is required to capture the user's intention: The user has to select appropriate construction tools to enforce desired constraints, e.g., incidence or orthogonality. A similar rationale applies to parametric modeling with CAD programs: The user has to select a tool, e.g., 'add a rectangle,' followed by interactive model instantiation.

Alternatively, the construction principles can be automatically inferred from the input by stochastic and geometric reasoning, i.e., the automatic recognition and enforcement of constraints. The tracked positions of the input device shown in Figure 1 on the right side are the strokes constituting the user's input. Here,

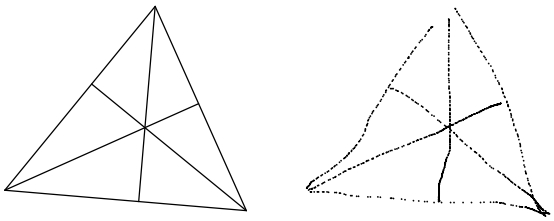

Figure 1. Orthocenter of a triangle. Left: construction, right: intended freehand strokes as user input.

the consideration of uncertainty is essential since the precision and the accuracy of the data depend on many factors, especially on

- the user's skill to draw accurately given an input device such as computer mouse, stylus, or finger in combination with a touchpad or screen,
- the user's willingness to draw accurately according to the predetermined model, e.g., straight line segments or circular arcs,
- the present environmental conditions such as the quantity of light or the view angles, and
- the resolution of the used input device.

For the realization of successful applications, the uncertainty caused by these effects must be captured, represented, and considered in the entire reasoning process. Thus, we argue that a stringent and rigorous stochastic approach is decisive to recognize the construction principles automatically. Following this paradigm, the user needs not be explicitly aware of the sequential construction process and there is less interaction required since the selection of appropriate construction tools or modes by the user is dispensable.

*Corresponding author

1.2 Contributions

Following the approach presented in (Meidow and Lucks, 2019), we discuss geometric reasoning in the context of educational technology. This comprises the approximation of strokes by straight line segments and the enforcement of automatically recognized constraints by adjustment. The interactive study of geometric problems is supported by a modeless software tool that allows a hands-on approach for geometric reasoning. The corresponding learning-by-doing process is accompanied by various visualizations, e.g., confidence regions, types and relevance of constraints, etc.

We consider three prominent applications with strokes as user input and line drawings as output:

- Theorem proving for geometric problems by reasoning, e.g., closing theorems such as Desargues's theorem.
- The polygonal approximation of building outlines featuring geometric constraints for the edges, such as parallelism or orthogonality.
- Computer-aided constructions of two-dimensional human-made figures, e.g., flow charts, diagrams or technical line drawings.

1.3 Related Work

To the best of our knowledge, there are no contributions which cover all aspects of the task at hand. Probably closest to our approach is the work of Johnson et al. called "Sketch it, make it" in the context of modeling for laser cutting (Johnson et al., 2014). The pen-based interaction proposed in this paper does not require the user to enter persistent modes but exploits drafting conventions: For instance, the designer has to draw tick marks and right-angle braces to indicate line segments of identical length and enclosed right angles. Disambiguities are resolved by a not otherwise explicated "constraint manager."

In contrast to (Johnson et al., 2014), we utilize the uncertainty of the data. Early discussions on the uncertainty of a straight line can be found in (Wolf, 1938). Representations for uncertain straight line segments can be found in (Beder, 2004) and (Meidow et al., 2009) as well as in up-to-date textbooks, e.g., (Förstner and Wrobel, 2016). The hyperbolic error band has been used for the visualization of the uncertainty of epipolar lines (Faugeras, 1993, p. 351).

The solution of equation systems with the help of Groebner bases has been studied in the context of cartographic generalization in (Brenner and Sester, 2005). In (Loch-Dehbi and Plümer, 2011, Loch-Dehbi and Plümer, 2009) independent constraints are found by automatic theorem proving using Wu's method (Wu, 1986). The feasibility is demonstrated, but no real data sets have been analyzed.

2. METHODS

The implementation of the proposed approach is based on various concepts to get a modeless automatic procedure that features real-time capability. They are reflected by the following workflow, which has to be applied after adding a stroke to the drawing:

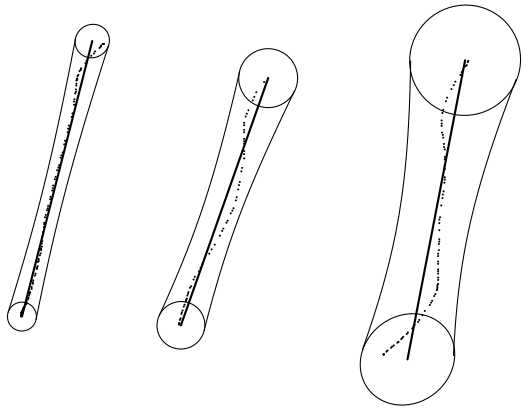

Figure 2. Strokes with different precisions depicited by the confidence regions of the approximating straight line segments. The confidence regions are most narrow close to the centroid of the corresponding point sets.

1. the approximation of the user's strokes by straight line segments (feature extraction)
2. the recognition of geometric relations (hypothesis generation testing)
3. the identification of subtasks (connected components), and
4. the selection of consistent and independent constraints for
5. the enforcement of the deduced constraints (adjustment).

The first step is essential to obtain a reasonable estimate for the precision of the input data. This information is used in the following for hypotheses testing and adjustment.

2.1 Recognition of Geometric Relations

Feature extraction and adjacencies. Given a stroke as a set of uncertain point coordinates, we estimate a statistically best-fitting straight line and the positions of the two end-points delimiting the straight line segment. Besides the estimate of the line parameters $\mathbf{l}$, the procedure provides an estimate of the covariance matrix $\mathbf{\Sigma}_{ll}$ of the parameters and an estimate of the a priori unknown variance factor which reflects the precision of the observed point coordinates (Förstner and Wrobel, 2016, pp. 397–400).

The estimates are only representative if no outliers are present and the model is valid, i.e., a straight line segment reasonably approximates a stroke. Thus, we assume the user to draw as accurately as possible straight line segments and do not apply a robust estimation. Figure 2 shows examples of different strokes. The variations might result from the user's skills and willingness to draw accurately, from varying environmental conditions, and from different resolutions of the input devices. Because of a line's uncertainty in direction (orientation) and lateral position, the confidence region is given by the area between two hyperbola branches. As to be expected, the confidence region is most narrow close to the centroid of the corresponding point set.

For the generation of hypotheses to be checked, we consider only pairs and triples of spatially adjacent segments. Two segments are considered to be adjacent if they intersect, if one

constraint	distance	ν
orthogonal	$d_{\perp}(\ell, m) = \mathbf{l}^{\mathsf{T}}\boldsymbol{D}\mathbf{m}$	1
parallel	$d_{\parallel}(\ell, m) = -\mathbf{l}^{\mathsf{T}}\boldsymbol{S}\mathbf{m}$	1
concurrent	$d_{\circ}(\ell, m, n) = \det([\mathbf{l}, \mathbf{m}, \mathbf{n}])$	1
identical	$\boldsymbol{d}_{\equiv}(\ell, m) = \boldsymbol{J}^{\mathsf{T}}(\mathbf{l} - \mathbf{m})$	2

Table 1. Distance measures according to (Meidow and Lucks, 2019) for three straight lines ℓ, m, and n with corresponding degrees of freedom ν. See the text for explanation of the variables.

segment touches the other, or if both segments touch each other, i.e., form a chain. These relations can be recognized by checking the incidences of end-points and straight lines for all pairs of segments (Beder, 2004). Here, the overlap of axis-aligned bounding boxes can be utilized to cull candidates.

Geometric Constraints. Adopting the approach presented in (Meidow and Lucks, 2019), we introduce the relations 'parallel', 'orthogonal', 'identical', and 'concurrent', i.e., 'copunctual', as geometric relations for pairs and triplets of straight lines. Further applicable constraints are discussed in (Heuel, 2004) and (Brenner and Sester, 2005). After recognizing an identity constraint, the involved segments can be merged immediately by averaging. This approach avoids usually unnecessary identity constraints and allows for easy prolongations of segments.

Table 1 summarizes for three straight lines ℓ, m, and n the used constraints and the corresponding distance measures. We utilize the homogeneous representations $\mathbf{l}$, $\mathbf{m}$, and $\mathbf{n}$ for the three lines to formulate the constraints as polynomials with the diagonal matrix $\boldsymbol{D} = \mathrm{Diag}([1, 1, 0])$ and the skew-symmetric matrix $\boldsymbol{S} = [0, 0, 1]_{\times}$. The columns of the 3×2 matrix $\boldsymbol{J}$ span the nullspace of the spherically normalized vector $\mathbf{l}^{\mathsf{T}}$ and the projection leads to two independent equations for the identity constraint (Förstner and Wrobel, 2016).

For the distance measures listed in Table 1 we compute variances and covariance matrices, respectively, by variance-covariance propagation. Theses entities serve for the formulation of test statistics within the hypothesis testing.

Hypotheses Generation and Testing. We formulate the null hypothesis that an assumed geometric relation is correct. In this case, we have the expectation values $\mathrm{E}[\boldsymbol{d}] = \mathbf{0}$ for the means of the distances summarized in Table 1. Since we have no prior probabilities for alternative hypotheses, we follow the standard testing procedure according to (Neyman and Pearson, 1933). The corresponding test statistics

$$T = \boldsymbol{d}^{\mathsf{T}}\boldsymbol{\Sigma}_{dd}^{-1}\boldsymbol{d} \quad \sim \quad \chi^2_{\nu} \tag{1}$$

is χ^2-distributed with $\nu = 1$ or $\nu = 2$ degrees of freedom in our case.

The probability α of rejecting a null hypothesis although it actually holds, should be low. After choosing a significance number, e.g., $\alpha = 0,05$, the critical value $\chi^2_{1-\alpha;\nu}$ of the corresponding χ^2_{ν}-distribution is computed. The hypothesis will not be rejected if $T < \chi^2_{1-\alpha;\nu}$ holds for the value of the test statistics.

The result of these tests is a set of constraints which represent the identified geometric relations and are likely to hold.

Input: Set $\mathcal{C}$ of consistent and non-redundant constraints required up to now, set $\overline{\mathcal{C}}$ of redundant constraints recognized up to now, set $\mathcal{A}$ of A additional constraints a_i
Output: Set $\mathcal{C}'$ of functionally independent constraints, set $\overline{\mathcal{C}}'$ of redundant or contradictory constraints

function GREEDYSEARCH($\mathcal{C}$, $\overline{\mathcal{C}}$, $\mathcal{A}$)
  $\mathcal{C}' \leftarrow \mathcal{C}$
  $\overline{\mathcal{C}}' \leftarrow \overline{\mathcal{C}}$
  for $i = 1$ to A **do**
    $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{a_i\}$ ▷ to be checked
    Compute $M \times N$ matrix $\boldsymbol{H}$ iteratively for set $\mathcal{C}'$.
    if $\mathrm{rank}(\boldsymbol{H}) < \min(M, N)$ **then**
      $\mathcal{C}' \leftarrow \mathcal{C}' \setminus \{a_i\}$ ▷ remove again
      $\overline{\mathcal{C}}' \leftarrow \overline{\mathcal{C}}' \cup \{a_i\}$
    end if
  end for
end function

Table 2. Greedy search for functionally independent constraints. $\boldsymbol{H}$ is the Jacobian matrix of the constraints.

2.2 Adjustment

The interactive construction is a step-by-step procedure. During this process, the human operator is geared to the results obtained so far. Thus, the formulation of an incremental and recursive adjustment seems to be natural and adequate. We consider new, additional straight lines and the results of the preceding adjustment to be the unconstrained input for the next adjustment. Then, a greedy search is applied to select consistent and independent constraints.

Identification of Subtasks. For large-scale problems, the decomposition into sub-tasks is mandatory. By considering the segments and constraints of a construction as vertices of a bipartite graph, we relate the segments to the recognized constraints and vice versa (Meidow and Lucks, 2019). The connected components of this graph constitute sub-tasks that can be solved separately. After adding a new straight line segment to the scene, or after deleting segments and constraints, the connected components have to be determined anew. This determination can fast and be done efficiently by a depth-first search (Cormen et al., 2009).

Selection of Independent Constraints. The adjustment requires a set of consistent and functionally independent constraints. For an incremental construction process, a greedy search appears to be the natural choice to determine such sets automatically.

Figure 2 lists a pseudo-code for this approach: Given a set $\mathcal{C}$ of already established constraints, the algorithm checks the newly recognized constraints $\mathcal{A} = \{a_i\}$ consecutively whether they are redundant or actually contradictory. A constraint is consistent if the Jacobian matrix $\boldsymbol{H}$ of the constraints has full rank. Constraints which have been discarded might become subsequently necessary after the deletion of constraints by the user. Therefore, we keep track of the rejected constraints, too, and the greedy search can be used after incremental and decremental operations.

For large problems with many constraints, the procedure described above can become numerically unstable, since it depends on the rank of the Jacobian matrix. Since all of the constraints

can be formulated as polynomials, the greedy search can also be performed using algebraic methods, i.e., Groebner basis calculations (Brenner and Sester, 2005). These methods do not rely on numerical properties and can in principle handle problems of arbitrary size. For algebraic reasoning it is advantageous to formulate the constraints differently and to process parallelism and identity first, then orthogonality and concurrence last. Especially in the presence of concurrence constraints, the Groebner bases can become quite large leading to memory problems. For details of our approach to algebraic reasoning, we refer to (Meidow and Hammer, 2016).

3. IMPLEMENTATION

A successful realization of the methods explained in Section 2 implies proper representations of the geometric entities and constraints and appealing human engineering. Both aspects are crucial for the efficiency and the acceptance of software solutions.

3.1 Representations

We formulate geometric constraints for straight lines. These correspond to straight line segments, which result from the approximation of strokes as user input. By representing the straight lines with homogeneous coordinates, we obtain constraints which are simple and bilinear w.r.t. the observations. However, the utilization of homogeneous coordinates constitutes an over-parametrization and requires normalization constraints to obtain unique results within the adjustment process. Thus, compared to a conventional Euclidean parametrization, we initially get larger equation systems to be solved (Meidow et al., 2009).

To avoid this overload, we exploit the concept of reduced coordinates to come up with a minimal representation of spherically normalized homogeneous coordinate vectors representing a straight line ℓ

$$\ell : \{\mathbf{l}, \boldsymbol{\Sigma}_{l_r l_r}\}\,, \tag{2}$$

cf. (Förstner and Wrobel, 2016, Chap. 10.2.2, eq. 10.31). This representation contains the spherically normalized 3-vector $\mathbf{l}$ representing the straight line ℓ and the 2×2 covariance matrix $\boldsymbol{\Sigma}_{l_r l_r}$ representing the uncertainty of the straight line in terms of the so-called reduced coordinates $\boldsymbol{l}_r$ in the tangent plane of the sphere at point $\mathbf{l}$.

Note that this representation allows for easy testing of geometric relations and estimation of geometric entities. See (Förstner and Wrobel, 2016) for a comprehensive presentation.

3.2 Human Engineering

An ergonomic and user-friendly graphical user interface is required to increase the acceptance of the proposed approach. This interface comprises (1) the interactive input in the form of strokes, (2) the visualization of approximating straight line segments and deduced constraints, and (3) an editing possibility including automatic updates by adjustment.

User input and real-time feedback. In the approach, the selection of operational modes is replaced by the automatic recognition of constraints and their subsequent enforcement. Thus, strokes are the only input for the construction or delineation. However, an additional editing functionality is desirable since we cannot expect all automatically drawn decisions to be correct in terms of 'intended.' To support the learning-by-doing approach, the user should be able to revise the reasoning results, e.g., by undo/redo operations, deletion of unintended segments or constraints, etc. Furthermore, immediate feedback is desirable during the step-by-step construction of line drawings. To achieve real-time capability for problems of moderate size, the identification of sub-tasks is mandatory. Sets of independent constraints have to be determined, which can be solved independently.

Visualizations. The independent sets of constraints and straight line segments form connected components which can be presented color-coded to the user to indicate the sub-tasks. The tracked positions of the input device form ordered sequences of points (strokes), which are approximated by straight line segments. The uncertainty of these segments is depicted by drawn confidence regions, i.e., ellipses for end-points and hyperbolas for straight lines. By varying the precision of the strokes, the human operator can study the effects on the recognition and enforcement of constraints. Last but not least, color-coded symbols can illustrate the types (geometric relations) and the relevance (required or not required) of the recognized constraints at appropriate positions.

4. EXPERIMENTS

Several experiments can be conducted by using a software implementation that exploits the design principles explained in the previous section. We discuss three use cases: Theorem proving for geometric constructions, the polygonal approximation of building outlines and roofscapes, and the creation of plain technical drawings or diagrams.

4.1 Performance Evaluation

If an evaluation of results is possible or not, depends on the specific application:

The investigation of geometric configurations implies the identification of independent constraints, e.g., for the study of closing theorems. We use numerical criteria within a greedy selection process, which might lead to numerical problems for large-scale problems. Since the constraints are formulated as polynomials, we can apply algebraic methods for theorem proving and thus for the verification of results. An interactive polygonal acquisition of an object's shape is usually subjective. The result depends on human interpretation and the required degree of generalization and approximation, respectively. Actually, interactive acquisitions often serve as reference data, e.g., for the evaluation of results obtained by automatic methods. However, given a second, independent acquisition, the results can be compared w.r.t. completeness, e.g., by the determination of the overlap (Meidow and Lucks, 2019). No evaluation is possible for the constructions of human-made figures, e.g., flowcharts, diagrams, or technical drawings.

4.2 Theorem Proving

We consider the construction of Pappus's hexagon to illustrate the corresponding theorem in its affine form:

Theorem 1 (Pappus's hexagon theorem) *If six points x_1, y_2, x_3, y_3, x_2 and y_1 of an affine plane are alternating incident with two straight lines ℓ and m, and if the straight line $\overline{x_1\, y_2}$ is parallel to $\overline{x_2\, y_3}$ and the straight line $\overline{y_1\, x_2}$ is parallel to $\overline{y_2\, x_3}$, then the straight lines $\overline{y_1\, x_1}$ and $\overline{x_3\, y_3}$ are also parallel.*

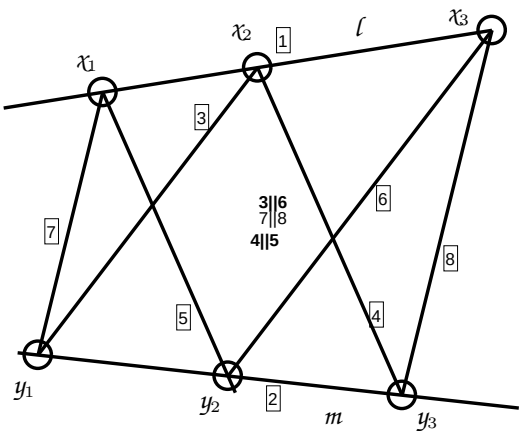

Figure 3. The construction of Pappus's hexagon featuring nine constraints for the straight lines: Six times concurrence ('$\odot$') and three times parallelism ('$\|$'). The constraint '$7\|8$' has been recognized to be redundant.

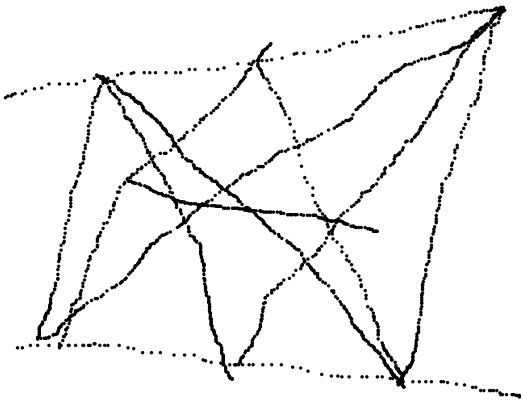

Figure 4. Strokes used as input for the construction depicted in Figure 5.

Figure 3 shows a corresponding construction featuring nine constraints: Six times concurrence ('$\odot$') and three times parallelism ('$\|$') of straight lines. According to the theorem, we have eight independent constraints. Please note, that the redundant constraint identified by the greedy search (Algorithm 2) might be parallelism or concurrence—depending on the order of the recognized constraints. Furthermore, bear in mind that in contrast to ruler-and-compass constructions, the line ℓ or the line m or both lines can be drawn last with the proposed approach. The construction of the hexagon shown in Figure 3 is based on strokes and fitted straight line segments.

If we add the two straight lines $\overline{y_1\, x_3}$ and $\overline{x_1\, y_3}$ to the construction, we obtain the three intersection points $\overline{x_1\, y_2} \cap \overline{y_1\, x_2}$, $\overline{x_1\, y_1} \cap \overline{y_1\, x_3}$, and $\overline{y_2\, x_2} \cap \overline{x_2\, y_3}$, which are collinear, see Figure 5. Figure 4 shows the strokes used as input for the construction.

In total, 24 geometric constraints have been recognized: 21 times concurrence and three times parallelism. Note, that if four straight lines meet in a point, $\binom{4}{3} = 4$ concurrence constraints can be formulated, but just two of them are functionally independent. Thus, a set of 14 constraints remains containing 12 times concurrence and two times parallelism or 11 times concurrence and three times parallelism.

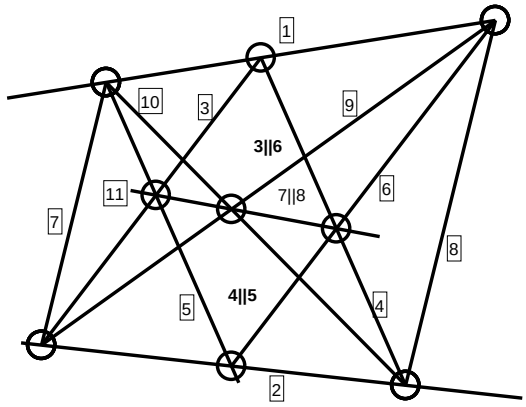

Figure 5. Pappus's hexagon with three collinear intersection points in the middle. In particular, the constraints '$7\|8$' and '$\odot(9, 10, 11)$' are redundant.

4.3 Polygonal Approximation of Outlines

As a prominent application, we demonstrate the acquisition of building outlines and roofscapes based on orthophotos. Figure 6 shows interactively outlined roof areas in a true orthophoto (Cramer, 2010). Redundant constraints are shown in light gray. Many of the parallelism constraints have been identified to be redundant since orthogonality is presented first to the greedy search. The set of constraints and segments splits into five connected components: The building's roof and four dormers. Each component constitutes a subtask to be solved.

Table 3 summarizes the recognized constraints and their relevance for each subtask. In general, different orders of consideration during the greedy search will lead to different sets of independent constraints. This becomes clear just considering a rectangle. Here, six constraints can be found, four times orthogonality and two times parallelism. Now, if the orthogonality constraints are processed first, as is the case with our numerical formulation, three of them are functionally independent, and the parallelism constraints are a result of these. If, as is the case with our algebraic formulation, parallelism is considered first, the set of independent constraints consists of two times parallelism and one orthogonality constraint, the other orthogonalities following from these. However, the results of the adjustments will be identical due to the geometric equivalence.

4.4 Constructions

To illustrate the design of a simple technical drawing, we consider the construction of a building with a gable roof in parallel projection with hidden line removal. Figure 8 depicts a construction obtained by geometric reasoning and subsequent adjustment. The corresponding strokes are shown in Figure 9. The dashed lines have been introduced as auxiliary lines to enforce eaves at the same height and the symmetry of the roof. These lines can later be removed to obtain a proper model of the building.

The construction in Figure 8 features 13 straight line segments and 45 constraints—13 times orthogonality ('$\square$'), 12 times parallelism ('$\|$'), one time identity ('$\bowtie$'), and 19 times the concurrence of three straight lines each ('$\odot$'). Actually 14 constraints are required to enforce all recognized geometric relations.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-5-2020-77-2020 | © Authors 2020. CC BY 4.0 License.

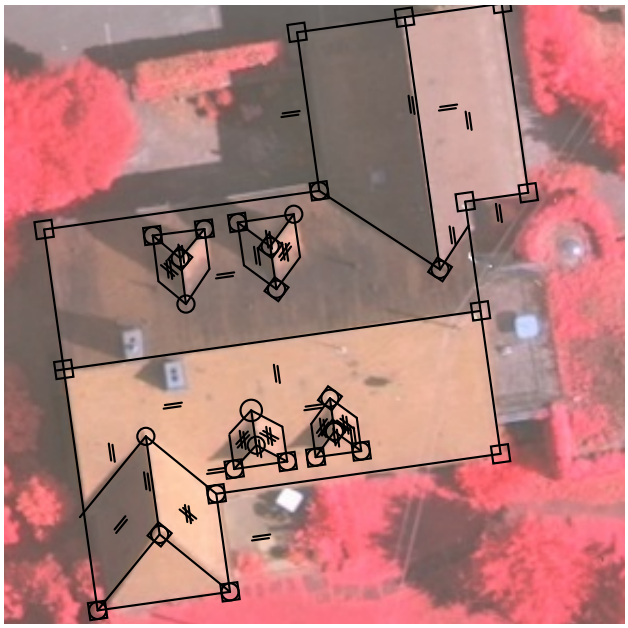

Figure 6. Brightened true orthophoto of an urban scene with superimposed outlines obtained by the recognized constraints (□, ||, ⊙) and subsequent adjustment.

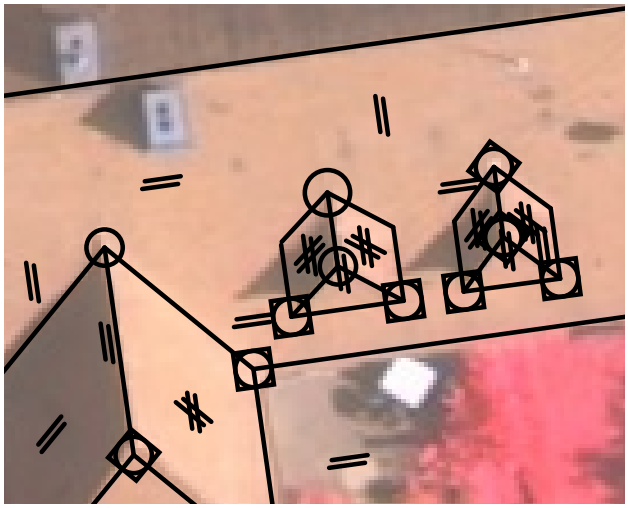

Figure 7. Display detail of Figure 6 showing the building's dormers which constitute subtasks to be solved.

connected component	constraint type	recognized	required
#1	orthogonality	13	11
19 segments	parallelism	21	3
	concurrence	7	7
#2	orthogonality	2	2
8 segments	parallelism	3	2
	concurrence	4	4
#3	orthogonality	3	3
8 segments	parallelism	2	1
	concurrence	3	3
#4	orthogonality	1	1
8 segments	parallelism	5	4
	concurrence	4	4
#5	orthogonality	2	2
8 segments	parallelism	5	3
	concurrence	4	4

Table 3. Recognized and required constraints for the scene shown in Figures 6 and 7.

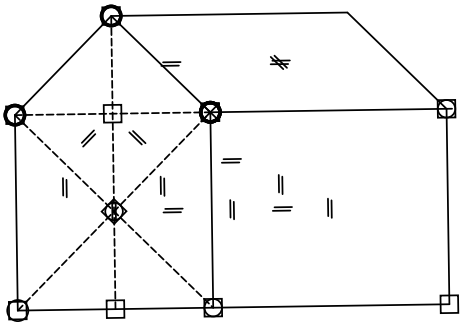

Figure 8. Drawn wireframe model of a building with gable roof in parallel projection. The construction with 14 segments features 45 constraints of which 21 are required. The dashed auxiliary lines enforces eaves at the same height and a symmetric roof.

After removing the auxiliary lines, we yield the final result depicted in Figure 10. The construction features nine segments and 15 constraints of which nine are actually required, i.e., non-redundant.

5. SUMMARY AND OUTLOOK

5.1 Summary

The automatic recognition of constraints and their subsequent enforcement offer a new paradigm for the construction of technical drawings by computer-aided design. Compared to conventional approaches, the selection of operation or construction modes becomes dispensable. The common interactive snapping of geometric entities is replaced by the automatic detection of incidences and identities, respectively, by hypothesis testing. As a consequence, the user's interaction is limited to perform strokes by a press-drag-release or touch-draw-lift-off sequence. These motions provide tracked positions as point sequences, which are the bases for statistical analysis: Given a specific input device and environmental conditions, the user's willingness and capability to draw can be captured by the estimation of straight line segments.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-5-2020-77-2020 | © Authors 2020. CC BY 4.0 License.

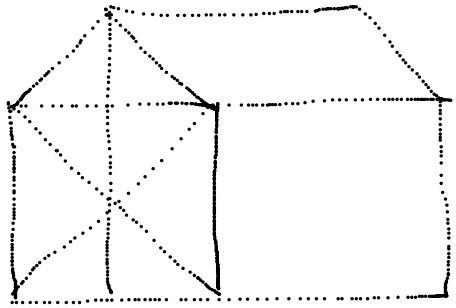

Figure 9. Strokes used for the construction shown in Figure 8. .

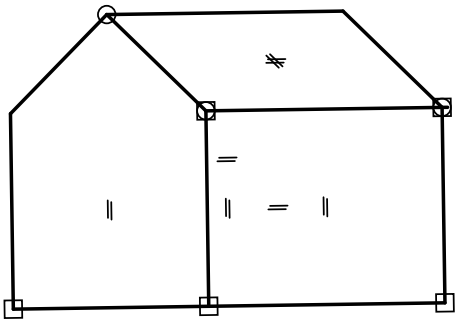

Figure 10. Line drawing after removing the auxiliary lines shown in Figure 8. The construction features 9 segments with 15 geometric constraints of which 9 are required.

The reasoning is then performed with these segments and the corresponding straight lines. Thus, the only relevant parameter is the significance level, which might be adjusted by the user. By zooming into the scene, the user can draw locally more accurately. The identification of connected components for the sets of related segments and constraints paves the way for large-scale applications. In contrast to ruler-and-compass constructions, the user must not explicitly be aware of a feasible order of construction steps. For example, to construct the orthocenter of a triangle as depicted in Figure 1, the user could start by drawing the altitudes first provided that he or she has a good awareness of geometric dimensions and construction principles.

For the demonstration of technology and educational purposes, we implemented a software tool featuring a user-friendly interface to increase the method's acceptance. This interface covers the interactive input, visualizations, and an editing possibility. The latter is decisive since we cannot expect all automatically drawn decisions to be correct in terms of 'intended.' Thus, the user is able to revise the reasoning results by undo-operations, deletion of unintended segments and constraints, etc. By adjusting the significance level, the user can control the sensitivity of the recognition. Visual feedback is provided by showing confidence regions, indicating recognized constraints, and marking sub-tasks. This allows for the interactive exploration and study of geometric problems and constructions.

5.2 Outlook

For the future, the incorporation of additional constraints is envisaged, e.g., equidistant parallels or identical angles, which enables free-hand drawing of work-flows or diagrams, for instance. If also points are considered explicitly, metric constraints can be envisioned, too.

For large-scale applications, numerical problems are likely to occur due to decisions based on estimated ranks and condition numbers. Exact solutions could be obtained by applying algebraic methods, for instance, automatic theorem proving using Wu's method (Wu, 1986). Unfortunately, the number of resulting polynomials in the bases and the orders of the polynomials can be quite large, leading to long computations. This hampers real-time applications. However, the algebraic complexity depends on the formulation of the constraints and the parametrization of the geometric entities.

ACKNOWLEDGMENT

The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (Cramer, 2010): `http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html`

REFERENCES

Beder, C., 2004. Fast statistical geometric reasoning about uncertain line segments in 2D- and 3D-space. *Pattern Recognition* 3175, pp. 375–382.

Brenner, C. and Sester, M., 2005. Cartographic generalization using primitives and constraints. In: *XXII International Cartographic Conference*, La Coruna, Spain.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., 2009. *Introduction to Algorithms*. 3rd edn, The MIT Press.

Cramer, M., 2010. The DGPF test on digital aerial camera evaluation – Overview and test design. *Photogrammetrie – Fernerkundung – Geoinformation* 2, pp. 73–82.

Faugeras, O., 1993. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. 1st edn, The MIT Press, Cambridge, MA, USA.

Förstner, W. and Wrobel, B. P., 2016. *Photogrammetric Computer Vision*. Springer.

Heuel, S., 2004. *Uncertain Projective Geometry. Statistical Reasoning in Polyhedral Object Reconstruction*. Lecture Notes in Computer Science, Vol. 3008, Springer.

Johnson, G. G., Do, E. Y., Gross, M. D. and Hong, J. I., 2014. Sketch it, make it: Freehand drawing for precise rapid fabrication.

Loch-Dehbi, S. and Plümer, L., 2009. Geometric reasoning in 3D building models using multivariate polynomials and characteristic sets. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Loch-Dehbi, S. and Plümer, L., 2011. Automatic reasoning for geometric constraints in 3D city models with uncertain observations. *ISPRS Journal of Photogrammetry and Remote Sensing* 66(2), pp. 177–187.

Mäntylä, M., 1987. *An Introduction to Solid Modeling*. Computer Science Press, Inc., New York.

Meidow, J. and Hammer, H., 2016. Algebraic reasoning for the enhancement of data-driven building reconstructions. *ISPRS Journal of Photogrammetry and Remote Sensing* 114, pp. 179–190.

Meidow, J. and Lucks, L., 2019. Draw and Order – Modeless interactive acquisition of outlines. *ISPRS – International Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* V-2/W7, pp. 103–110.

Meidow, J., Beder, C. and Förstner, W., 2009. Reasoning with Uncertain Points, Straight Lines, and Straight Line Segments in 2D. *ISPRS Journal of Photogrammetry and Remote Sensing* 64(2), pp. 125–139.

Neyman, J. and Pearson, E. S., 1933. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of The Royal Society* 231, pp. 289–337.

Wolf, H., 1938. Über die Eigenschaft der plausibelsten Gerade einer fehlerzeigenden Punktreihe. *Zeitschrift für Instrumentenkunde* 11, pp. 429–442.

Wu, W., 1986. Basic principles of mechanical theorem proving in elementary geometries. *Journal of Automated Reasoning* 2, pp. 221–252.