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, 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

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

- The enforcement of the deduced constraints (adjustment).
- The identification of subtasks (connected components), and
- Feature extraction and adjacencies.

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 \( l \), the procedure provides an estimate of the covariance matrix \( \Sigma_l \) 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 sets.

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

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Distance (d)</th>
<th>(\nu)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orthogonal</td>
<td>(d_1(l, m) = 1^T Dm)</td>
<td>1</td>
</tr>
<tr>
<td>Parallel</td>
<td>(d_2(l, m) = -1^T Sm)</td>
<td>1</td>
</tr>
<tr>
<td>Concurrent</td>
<td>(d_3(l, m, n) = \det([l, m, n]))</td>
<td>1</td>
</tr>
<tr>
<td>Identical</td>
<td>(d_4(l, m) = J^T(1 - m))</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2. Greedy search for functionally independent constraints. \(H\) is the Jacobian matrix of the constraints.

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

1: function GREEDYSEARCH(\(C, \bar{C}, A\))
2: \(C' \leftarrow C\)
3: \(\bar{C}' \leftarrow \bar{C}\)
4: for \(i = 1\) to \(A\) do
5: \(C' \leftarrow C' \cup \{a_i\}\)  \(\triangleright\) to be checked
6: Compute \(M\times N\) matrix \(H\) iteratively for set \(C'\).
7: if rank(\(H\)) < min(\(M, N\)) then
8: \(C' \leftarrow C' \setminus \{a_i\}\)
9: \(\bar{C}' \leftarrow \bar{C}' \cup \{a_i\}\)
10: end if
11: end for
12: end function

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.

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 \(E[|d|] = 0\) 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

\[
T = d^T \Sigma^{-1} d \sim \chi^2 \nu
\]

is \(\chi^2\)-distributed with \(\nu = 1\) or \(\nu = 2\) degrees of freedom in our case.

The probability \(\alpha\) 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 \(\chi^2_{\nu, \alpha}\) of the corresponding \(\chi^2\)-distribution is computed. The hypothesis will not be rejected if \(T < \chi^2_{\nu, \alpha}\) 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.
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 \( l \)

\[ l : \{1, \Sigma_{\ell}, l\}, \quad (2) \]

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

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 \( x_1, y_1, x_2, y_2, x_3, y_3 \) of an affine plane are alternating incident with two straight lines \( l \) and \( m \), and if the straight line \( \frac{x_1}{x_2} y_2 \) is parallel to \( \frac{x_1}{x_2} y_2 \) and the straight line \( \frac{y_1}{y_2} x_2 \) is parallel to \( \frac{y_1}{y_2} x_2 \), then the straight lines \( \frac{y_1}{y_2} x_2 \) and \( \frac{y_1}{y_2} x_2 \) are also parallel.
If we add the two straight lines and fitted straight line segments. The construction of the hexagon shown in Figure 3 is based on $m$ or both lines can be drawn last with the proposed approach. Thus, a set of 14 constraints remains containing 12 contrast to ruler-and-compass constructions, the line recognized to be redundant.

Figure 3. The construction of Pappus’s hexagon featuring nine constraints for the straight lines: Six times concurrence (‘⊙’) 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 (‘⊙’) 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 $\overrightarrow{y_1 x_3}$ and $\overrightarrow{x_1 y_1}$ to the construction, we obtain the three intersection points $\overrightarrow{x_1 y_1} \cap \overrightarrow{y_1 x_3}$, $\overrightarrow{x_3 y_1} \cap \overrightarrow{y_1 x_3}$, and $\overrightarrow{y_1 x_3} \cap \overrightarrow{x_3 y_1}$, 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, $\binom{4}{2} = 4$ concurrency 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.

### 4.3 Polygonal Approximation of Outlines

As a prominent application, we demonstrate the acquisition of building outlines and roofscape 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 (‘⊙’), 12 times parallelism (‘∥’), one time identity (‘=’), and 19 times the concurrency of three straight lines each (‘⊙’). Actually 14 constraints are required to enforce all recognized geometric relations.
Table 3. Recognized and required constraints for the scene shown in Figures 6 and 7.

<table>
<thead>
<tr>
<th>connected component</th>
<th>constraint</th>
<th>type</th>
<th>recognized</th>
<th>required</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 19 segments</td>
<td>orthogonality</td>
<td>13</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parallelism</td>
<td>21</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>concurrence</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>#2 8 segments</td>
<td>orthogonality</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parallelism</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>concurrence</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>#3 8 segments</td>
<td>orthogonality</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parallelism</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>concurrence</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>#4 8 segments</td>
<td>orthogonality</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parallelism</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>concurrence</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>#5 8 segments</td>
<td>orthogonality</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parallelism</td>
<td>5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>concurrence</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

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.
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


