

DRAW AND ORDER — MODELESS INTERACTIVE ACQUISITION OF OUTLINES

Jochen Meidow*, Lukas Lucks

Fraunhofer IOSB, Ettlingen, Germany - (jochen.meidow, lukas.lucks)@iosb.fraunhofer.de

KEY WORDS: building outlines, interaction, constraints, geometric reasoning, adjustment

ABSTRACT:

The representation of an object's outline with polygons or more specific shapes is the first choice for interactive outline acquisition. Usually, an operator has to select appropriate tools to capture an object's outline according to his or her notion of the shape with its underlying construction principles. We propose a modeless approach guided by geometric reasoning, i.e., the user does not have to select operation modes according to his or her model. To cope with modern input devices such as touch pens, styluses, or fingers, we consider strokes as input in arbitrary order. By approximating these strokes with straight line segments, we obtain estimated levels of noise accounting for the uncertainty of the acquisition. These straight line segments are the basis for the subsequent hypothesis generation to recognize geometric relations. After identifying sets of consistent and nonredundant constraints, we enforce these constraints in an adjustment process. To demonstrate the usability of the proposed approach, we capture the outlines of buildings and roof areas in true orthophotos of a benchmark data set.

1. INTRODUCTION

1.1 Motivation

The analysis of aerial images—especially true orthophotos—is a widely-used approach to acquire topographic information for large-scale applications. However, although automatic recognition and extraction procedures become more and more powerful, they cannot provide error-free and complete results. Thus, computer-aided interactive acquisition will remain the standard approach to obtain reference data in the absence of cadastral data.

For the acquisition of building outlines and roofscapes, several proven approaches with their corresponding tools are known to obtain boundary representations:

- Generic descriptions are obtained by modeling a building's outline with polygons and the boundaries of adjacent roof areas with polylines. Alternatively, each roof area can be represented by a polygon with shared vertices, and the building's outline is the envelope of all polygons.
- Specific representations can be obtained by the instantiation of elementary shape geometries, primarily rectangles, again with possibly shared vertices.

The latter approach implies that the user is aware of the construction principles with the involved geometric constraints. The mode of operation restricts the user's interaction by imposing constraints on the possible pointer positions.

From a methodological point of view, the interactive acquisition is identical with the computer-aided design of human-made objects:

- Programs for ruler-and-compass constructions, such as CINDERELLA, feature edit functions, e.g., *add a straight line*,

add a perpendicular, or *add a point* (Richter-Gebert and Kortenkamp, 2012). After the activation, the typical press-drag-release sequence for computer mice is expected with possible snapping to already existing geometric entities.

- A commonly used method to capture building outlines with geographic information systems (GIS), e.g., QGIS (QGIS Development Team, 2009), is to create polygons. This is done by the user by successively marking the vertices of the outlines. In this way, it is possible to draw the outlines as free-forms without any geometric conditions between the sides of the polygon. Also, various editing aids are available, such as forcing perpendicularity of the newly recorded line to the previous one or drawing parallel lines.

Methodically similar working steps and recording tools as in GIS and ruler-and-compass programs can be found in programs for Computer-Aided Design (CAD).

In summary, the conventional approaches require the user to select appropriate construction tools or operation modes which enforce desired geometric constraints or instantiate geometric shapes. Thus, the user must explicitly be aware of the underlying construction principles. Polygonal chains, for instance, are constructed by inserting vertices sequentially, i.e., the order is not arbitrary. Right angles are enforced by the selected construction mode and not via automatic model selection.

To get rid of the time-consuming selection process, we propose an automatic inference of the construction principles by geometric reasoning. As input, we do not expect the conventional ordered set of points but strokes in any order representing straight line segments. Figure 1 illustrates the interactive drawing of a straight line segment: The tracked positions of the input device are initially approximated by a best-fitting straight line with a confidence region of hyperbolic shape. The projection of the extremal points onto this straight line yields the endpoints of the segment.

*Corresponding author

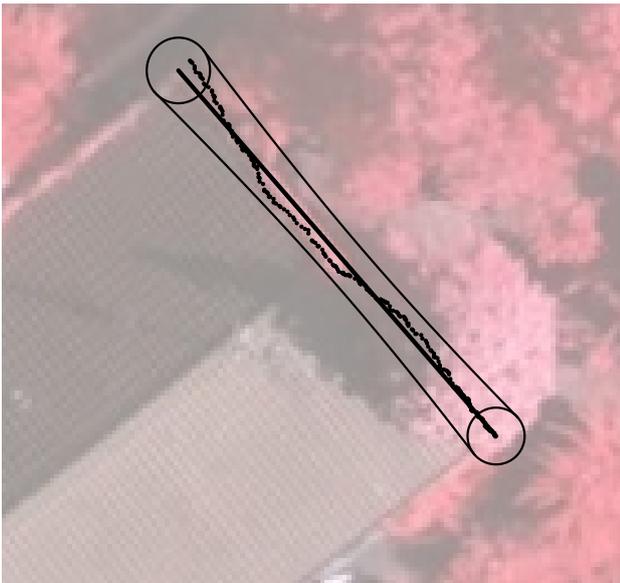


Figure 1. Interactive drawing of a straight line segment: A best-fitting straight line (solid) approximates the tracked positions of the input device (dots). The confidence region of the straight line has a hyperbolic shape, whereas ellipses depict the positional uncertainty of the two endpoints.

1.2 Contributions

We propose a complete workflow to obtain building outlines from given strokes guided by geometric reasoning. Since the user input is provided step-by-step, an incremental approach is pursued with a recursive estimation scheme to enforce recognized geometric constraints. The strokes are tracked point positions of the utilized input device, e.g., stylus, computer mouse, or finger. Each point set is approximated by a best-fitting straight line segment. This estimation also provides a noise level, which accounts for the various sources of uncertainty, e.g., the user’s skill and willingness to draw accurately, environmental conditions, or the resolution of the utilized input device. We assume isotropic uncertainty for the positions of the endpoints. This uncertainty is later used to snap or latch the endpoints to other segments, again utilizing hypothesis testing.

The adjacencies and constraints of the straight line segments are recognized by hypotheses generation and testing. Thus, the only parameter is the used significance level, which can be adapted by the user to change the sensitivity of the reasoning. From the recognized geometric relations, we derive sets of consistent and nonredundant constraints which are used eventually within an adjustment process to enforce the recognized constraints.

1.3 Related Work

In the context of modeling for laser cutting, Johnson et al. proposed a modelless system called *Sketch it, make it* (SIMI) which exploits drafting conventions to recognize the user’s intention (Johnson et al., 2014). For instance, the software recognizes tick marks and right angle braces to capture requested geometric constraints. A not otherwise specified constraint manager assesses the relevance of the constraints. Then uncertainty of the data is not considered.

Early discussions on the uncertainty of straight lines can be found in (Wolf, 1938). In (Faugeras, 1993), the visualizations of epipo-

lar lines include hyperbolic error bands. Efficient representations for uncertain straight line segments can be found in (Beder, 2004).

The proposed automatic recognition of relations by hypothesis testing follows the standard testing procedure according to Neyman and Pearson (Neyman and Pearson, 1933) as discussed in classical textbooks, e.g., (Koch, 1999). Common geometric constraints are listed as multivariate polynomials in (Heuel, 2004) and (Brenner, 2005). They are also the subject of many textbooks, e.g., (McGlone et al., 2004) or (Förstner and Wrobel, 2016).

The identification of independent and non-contradictory constraints can be accomplished exactly by utilizing algebraic methods, e.g., by exploiting the concept of Gröbner bases (Buchberger et al., 1988) or by automatic theorem proving using Wu’s method (Wu, 1986). However, computing a Gröbner base can take substantial time and space, so that its computation is in general not feasible in an interactive environment (Brenner, 2005). In (Meidow, 2014), a greedy algorithm utilizing numerical criteria is used to identify sets of consistent and nonredundant constraints. This publication also provides the adjustment procedure utilized in this work.

2. GEOMETRIC REASONING

The proposed reasoning is based on straight lines and their relations. The input and the construction, however, consist of straight line segments. Since the segments are derived from uncertain user input, we apply the standard procedure for hypothesis testing (Neyman and Pearson, 1933) to determine connectivity and to recognize geometric constraints. To do so, we formulate the null-hypotheses that the assumed relations are valid, i.e., the values of the distance measures serve as test statistics.

2.1 Straight Line Segments and Connectivity

The obvious representation of a straight line segment is a point pair (x, y) , where the points delimit the segment s , see especially Definition 3 in Euclid’s *Elements* (Heath and Euclid, 1908). Thus, the corresponding straight line l can easily be constructed by connecting the two endpoints. The utilization of points in homogeneous representation offers the possibility to define points at infinity and thus to represent half-lines, too.

For the determination of connectivity, we have to check if one of the endpoints of a given segment touches or intersects another segment. Thus, we have to check if a point is incident to a straight line and if the point is between two given points of this straight line. This test can efficiently be performed with the representation of a straight line segment

$$s : \{l, m, n\}, \quad l \perp m, \quad l \perp n \quad (1)$$

with three straight lines l , m , and n , where the two delimiting straight lines m and n are orthogonal to the straight line l (Beder, 2004, Meidow et al., 2009), see Figure 2.

Formally, a point z is incident with a straight line segment s if

$$z \in s \iff (z \in l) \wedge (z \in^+ m) \wedge (z \in^- n) \quad (2)$$

holds where \in^- and \in^+ means “left of” and “right of”, respectively. Thus, the two delimiting straight lines m and n have to feature a proper orientation.

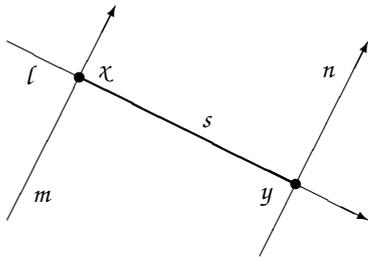


Figure 2. Utilized representations $\{\chi, y\}$ and $\{l, m, n\}$ of a straight line segment s . The oriented straight lines m and n delimit the segment s and are perpendicular to the straight line l .

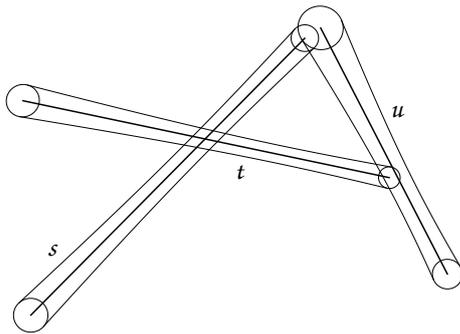


Figure 3. Three adjacent segments with their uncertainty regions: s intersects t , t touches u , and s and u touch each other, i.e., form a chain.

Given the point and the straight line in homogeneous representation \mathbf{z} and \mathbf{l} , the incidence can be checked with the algebraic distance

$$d_{\in}(\mathbf{z}, \mathbf{l}) = \text{sign}(z_h) \mathbf{z}^T \mathbf{l} \quad (3)$$

with the homogeneous coordinate z_h of the point. The distance can also be used to decide whether the point lies on the left or right side of the straight line.

Using this, a segment $s: \{\chi, y\}$ touches another segment t if

$$(\chi \in t) \vee (y \in t) \quad (4)$$

holds and two segments intersect, if and only if the endpoints of the first segment lie on different sides of the line defined by the second segment and vice versa (Cormen et al., 2009).

We consider two segments s and t to be adjacent if they intersect, if one segment touches the other, or if both segments touch each other, i.e., form a chain. Thus, the potential relationships “intersect” and “touch” have to be checked for each segment pair taking their uncertainty into account, see Figure 3. Adjacent segments have to be checked for the presence of geometric relations.

2.2 Constraints

As geometric relations, we allow for parallelism, orthogonality, concurrence, and identity. Table 1 summarizes for three straight lines l , m , and n the used constraints, the corresponding distance measures used as test statistics for hypothesis testing, and their degrees of freedom. For further applicable constraints, we refer to (Heuel, 2004) and (Brenner, 2005).

Exploiting the power of algebraic geometry, we utilize the homogeneous representations \mathbf{l} , \mathbf{m} , and \mathbf{n} for the three lines and for the

constraint	distance	dof
normalization	$d'(\mathbf{l}) = \mathbf{l}^T \mathbf{l} - 1$	1
orthogonality	$d_{\perp}(\mathbf{l}, \mathbf{m}) = \mathbf{l}^T \mathbf{D} \mathbf{m}$	1
parallelism	$d_{\parallel}(\mathbf{l}, \mathbf{m}) = -\mathbf{l}^T \mathbf{S} \mathbf{m}$	1
concurrence	$d_{\circ}(\mathbf{l}, \mathbf{m}, \mathbf{n}) = \det([\mathbf{l}, \mathbf{m}, \mathbf{n}])$	1
identity	$d_{=}(\mathbf{l}, \mathbf{m}) = \mathbf{J}^T (\mathbf{l}' - \mathbf{m}')$	2

Table 1. Distance measures for three straight lines l , m , and n with corresponding degrees of freedom (dof). The normalization $d'(\mathbf{l})$ to unit length ensures unambiguous adjustment results.

formulation of the constraints, we apply the diagonal matrix $\mathbf{D} = \text{Diag}([1, 1, 0])$ and the skew-symmetric matrix $\mathbf{S} = \mathbf{S}([0, 0, 1])$. The columns of the 3×2 matrix \mathbf{J} span the nullspace of \mathbf{l}^T and the projection leads to two independent equations for the identity constraint. Due to the chosen over-parametrization, we apply spherical normalization to fix the ambiguity of the homogeneous representation, see (6) in Section 4.1.

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.

3. CONNECTED COMPONENTS

The acquisition of building outlines or roof parts implies the adjustment of many segments because of potentially many recognized constraints. To cope with such large-scale scenes, we decompose the adjustment and snapping processes into sub-problems. A construction with two not connected parallelograms, for instance, consists of two connected components in terms of adjacency but features four constraints (parallelism) which can be enforced independently. To identify independent subtasks, we exploit concepts of graph theory and computer science, as explained in the following sections.

3.1 Adjacent Segments

To represent the adjacencies of segments, we utilize a symmetric adjacency matrix $\mathbf{A} = (a_{ij})$ filled with zeros and ones, where an entry $a_{ij} = 1$ denotes two adjacent segments, i.e., segment s_i touches segment s_j , segment s_j touches segment s_i , or both segments intersect. This matrix \mathbf{A} can be used to determine second-order adjacencies conveniently. For testing ternary relations, e.g., the concurrence of three straight lines, we have to check if two adjacent segments feature a common neighbor. This can easily be accomplished by squaring $\mathbf{A} = (a_{ij}^{[1]})$, since the entries of $\mathbf{A}^2 = \mathbf{A} \mathbf{A} = (a_{ij}^{[2]})$ denote the number of walks of length 2 between the vertices i and j .

3.2 Identification of Subtasks

For efficiency, we solve one subtask after the other. The subtasks can be identified by considering segments and constraints as vertices of a bipartite graph. With S segments and C constraints we obtain an $S \times C$ incidence matrix \mathbf{B} , sometimes called biadjacency matrix, which relates the segments to the constraints and vice versa. Figure 4 shows an example. The $(S + C) \times (S + C)$ adjacency matrix of the bipartite graph reads

$$\mathbf{G} = \begin{bmatrix} \mathbf{O} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{O} \end{bmatrix} \quad (5)$$

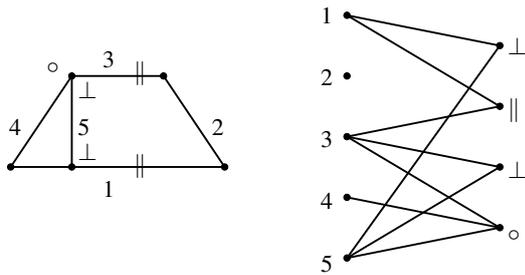


Figure 4. Left: Straight line segments and constraints. Right: Corresponding bipartite constraint graph.

with zero matrices \mathbf{O} . Its connected components represent subsets of segments to be adjusted by enforcing the related constraints.

After adding a new straight line segment to the scene or after removing segments and constraints, the connected components have to be determined anew. This determination can fast and efficiently be done by a depth-first search (Cormen et al., 2009).

4. RECURSIVE ADJUSTMENT

The interactive acquisition is a step-by-step procedure. Adding a straight line segment or deleting existing segments and constraints is followed by update operations, i.e., the recognition of possibly additional constraints, the re-determination of connected components, the identification of consistent and nonredundant constraints, and eventually the subsequent adjustment of all involved straight lines.

When drawing a further stroke, an operator usually refers to the constructed entities up to now. These are the adjustment results obtained before. Thus, we argue that a recursive adjustment is appropriate to model this construction process. Additional constraints affect new straight lines and already adjusted lines.

4.1 Adjustment

To obtain unique results, we apply a spherical normalization to each vector \mathbf{l} representing a straight line ℓ

$$\mathbf{l}' = \frac{\mathbf{l}}{\|\mathbf{l}\|} \quad (6)$$

and retain this property with the explicit constraint

$$\mathbf{l}'^T \mathbf{l}' = 1 \quad (7)$$

for each straight line in the adjustment process. We call these constraints intrinsic constraints and drop the superscript for clarity in the following.

The covariance matrix of the vector \mathbf{l} features a rank deficiency and $\Sigma_{\mathbf{l}} \mathbf{l} = \mathbf{0}$ holds. Thus, the nullspace of the covariance matrix reflects the explicit constraint (7), and the implicit constraint comprised in the covariance matrix and the explicit constraint by normalization must be consistent to handle singular covariance matrices properly in the adjustment model (Meidow et al., 2009).

Observing this, the estimated corrections $\hat{\epsilon}$ for N unconstrained straight lines ℓ_i , $i = 1, \dots, N$, stacked in the vector $\mathbf{x}^T =$

$[\mathbf{l}_1^T, \dots, \mathbf{l}_N^T]$ are

$$\hat{\epsilon} = -\Sigma_{xx} \mathbf{H}^T (\mathbf{H} \Sigma_{xx} \mathbf{H}^T)^{-1} (\mathbf{h}_0 - \mathbf{H} \mathbf{K}^T \mathbf{k}_0) - \mathbf{K}^T \mathbf{k}_0 \quad (8)$$

with the $3N \times 3N$ covariance matrix Σ_{xx} of parameters denoting the straight lines and the auxiliary variables $\mathbf{k}_0 = \mathbf{k}(\mathbf{x}_0) + \mathbf{K}(\mathbf{x} - \mathbf{x}_0)$ and $\mathbf{h}_0 = \mathbf{h}(\mathbf{x}_0) + \mathbf{H}(\mathbf{x} - \mathbf{x}_0)$ computed with approximate values \mathbf{x}_0 (Meidow, 2014). The approximate values are usually given by the unconstrained parameter values and the matrices \mathbf{H} and \mathbf{K} are the Jacobians of the geometric and intrinsic constraints respectively. As demanded, the estimate (8) requires no inversion of the singular covariance matrix Σ_{xx} . Note, that the size of the matrix $\mathbf{H} \Sigma_{xx} \mathbf{H}^T$ to be inverted depends only on the number of geometric constraints, but not on the—usually much larger—number of straight lines.

For the formulation of a recursive adjustment, we are considering new, additional straight lines and the results of the preceding adjustment to be the unconstrained input.

4.2 Selection of Independent and Consistent Constraints

The application of standard adjustment procedure requires the existence of independent and consistent constraints. A greedy algorithm appears to be appropriate to evaluate the relevance of newly recognized relations. Rank estimates and condition numbers of the coefficient matrices are serving as criteria to detect redundancy and contradictions. Besides these numerical criteria, algebraic methods are conceivable, e.g., automatic theorem proving. Unfortunately, the real-time capabilities of these methods are still limited in our view.

Constraints which have been discarded might become subsequently necessary. For example, constraints which have initially been marked *redundant* can become *required* after the interactive deletion of segments and corresponding constraints. Therefore, we have to keep track of the rejected constraints for possible re-evaluation.

5. EXPERIMENTS

We created a computer program to demonstrate the usability of the proposed approach and to perform experiments. In the following, we initially discuss the utilized data set and the computed assessment metric. This assessment is the basis for comparison with ground truth and results obtained with traditional acquisition tools.

5.1 Data Set and Ground Truth

True orthophotos are the ideal basis for acquiring the outlines of buildings and roof parts efficiently. To demonstrate the usability of the proposed approach, we utilize the Vahingen data set provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (Cramer, 2010). Figure 5 shows the used orthophoto with superimposed ground truth colored in yellow. The channels of the images contain the near-infrared, red and green bands.

Besides true orthophotos with 9 cm ground sampling distance, the data set contains ground truth for the class *building* represented by raster information. The reference data has been acquired interactively and thus is subjected to human interpretation and generalization. For example, roof parts occluded by overhanging trees



Figure 5. Used true orthophoto of the Vaihingen data set with superimposed ground truth for the class *building* in cyan.

have been classified as vegetation. Thus, the reference is uncertain to some degree and the building outlines are partial irregularly shaped, see Figure 6. Nonetheless, the reference data can be utilized for comparisons and to check the results for completeness.

Given two acquisition in the form of binary masks \mathcal{A} and \mathcal{B} , we compute the *intersection of union* (IoU) (Nowozin, 2014) or *quality rate* as the ratio of the areas of intersection and overlap:

$$q_{AB} = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|}, \quad q \in [0, 1] \quad (9)$$

The value of the quality rate is independent of the assignment of \mathcal{A} and \mathcal{B} and can be used to compare two results and to compare a result with ground truth. Its optimum is 1.

5.2 Acquisition with Geometric Reasoning

Figure 7 shows the acquisition of the building outlines by the proposed reasoning process based on strokes as input. 250 straight line segments have been defined by strokes of the utilized stylus. 420 geometric constraints have been recognized, of which 208 are required, i.e., independent. This scene features only orthogonality (\diamond) and parallelism (\parallel); the symbols for redundant constraints are drawn in gray. The result depends on the order of consideration for the constraints within the greedy algorithm. That is why most of the rejected constraints stand for parallelism.

The sets of straight lines and constraints decompose into 41 sub-tasks or components which can be adjusted separately. Please note that such a component does not reflect connectivity. A parallelogram, for instance, features parallelism twice which can be

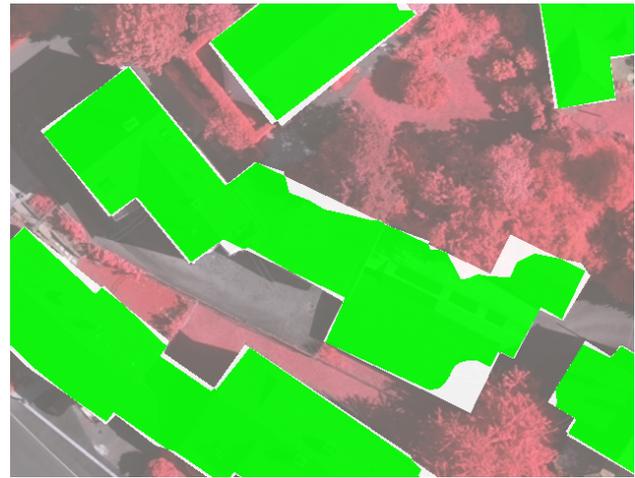


Figure 6. Ground truth (green) and acquisition (white). Both are uncertain especially because of the possible interpretations of roof-edges occluded by trees and the intended degree of generalization.

enforced by two consecutive adjustments. The number of connected components depicted in Figure 7 is 29 which corresponds to the number of buildings in the scene.

The polygon area for each connected component can be rasterized and the resulting masks \mathcal{R} can be compared to the ground truth \mathcal{G} . The quality rate or intersection of union (9) is $q_{RG} = 0.936$.

Moreover, the approach can be used to capture the shapes of roof areas defined by step or crease edges. Figure 8 shows the acquisition of such parts with outlines also featuring three times concurrence (\circ).

5.3 Acquisition with a Geographic Information System

To further evaluate the developed method, the extracted building outlines will be compared with outlines created by QGIS as an example of a non-commercial GIS application (QGIS Development Team, 2009). For this purpose, the buildings are shown in Figure 5 were captured by polygons by marking the corner points of the building in the underlying orthophoto. To compare the results, we calculated the IoU-score for both methods from the rasterized outlines \mathcal{Q} . With a value of about $q_{QR} = 0.955$, it can be stated that the results show only very small differences. The remaining differences result mainly from the different degrees of generalization obtained by the two approaches. While programs such as QGIS enable a pixel-accurate acquisition of the buildings without any contribution about the shape, the proposed method already delivers generalized outlines. This is highlighted in comparison with the ground truth data. Here, the QGIS results show a slightly higher IoU-score of $q_{QG} = 0.941$ ($q_{RG} = 0.936$, see section 5.2). The effects described can also be seen visually in the results. Figure 9 to 11 shows a comparison of the results obtained by the two methods.

6. CONCLUSIONS AND OUTLOOK

We propose a modelless approach to outline polygonal shapes guided by geometric reasoning. This approach implies that the user's selection of appropriate models and editing tools is omitted—which saves labor and working time. For example, the

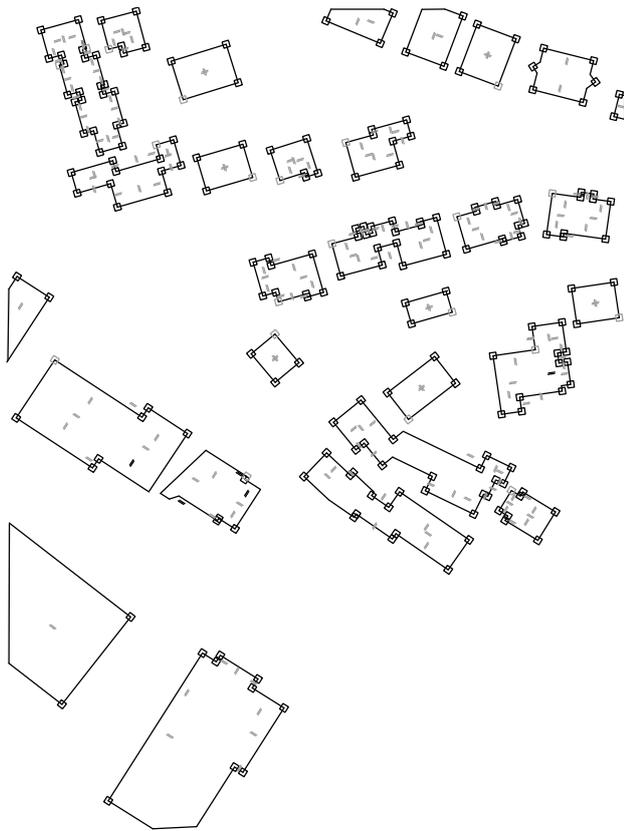


Figure 7. Interactively acquired building outlines with 250 straight line segments constrained by 420 geometric constraints. Redundant constraints are denoted by gray symbols.



Figure 8. An acquisition of roof areas featuring orthogonality, parallelism, and concurrence. Redundant constraints are marked in gray.

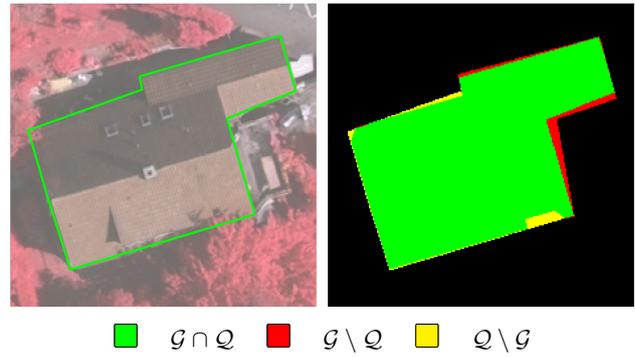


Figure 9. Evaluation of the unconstrained acquisition Q from QGIS compared to the ground truth G .

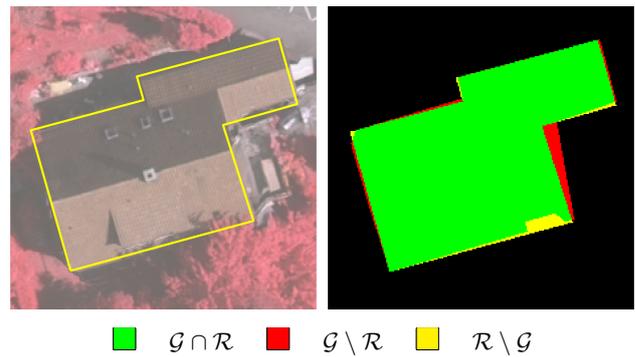


Figure 10. Evaluation of the constrained acquisition R guided by geometric reasoning compared to the ground truth G .

traditional drawing of a polygon by marking the polygon's vertices in consecutive order is replaced by drawing strokes in arbitrary order as known from free-hand drawing. Geometric relations such as orthogonality or parallelism are recognized and enforced automatically in real-time. Furthermore, the endpoints of straight line segments are snapped to adjacent segments where appropriate. Thus the user's intention has not to be captured explicitly.

6.1 Conclusions

Using strokes as input data, we consider emerging input devices with their new ergonomics. Touch pens, styluses or just fingers are utilized with digitizer tablets or touch screens. By tracking the position of a pen, we obtain sets of points representing straight line segments. The scattering of these positions depends on the

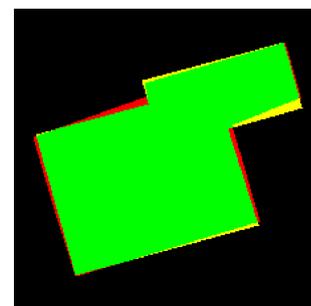


Figure 11. Comparison of both acquisitions (constrained R and unconstrained Q).

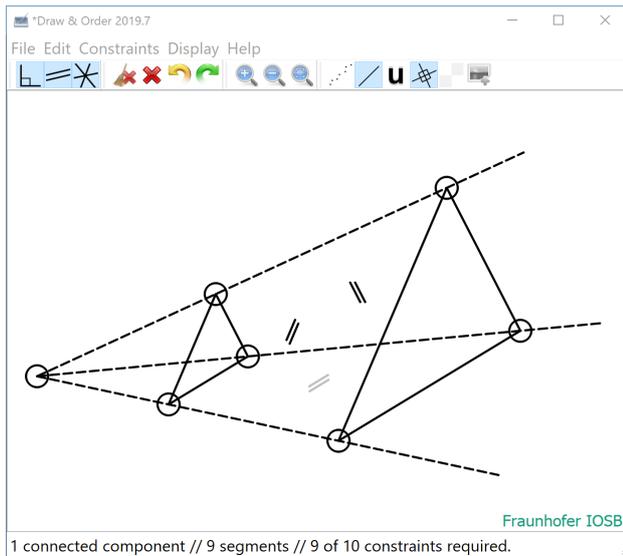


Figure 12. Desargues's theorem in its affine form: The sides of the two triangles are pairwise parallel. Nine of the ten constraints are required.

environmental conditions and on the user's willingness and capability to draw accurately. However, we obtain estimates for the noise level by fitting straight line segments to the points and thus consider the uncertainty of the acquisition. In any case, the accuracy can be increased by zooming into the scene. The recursive adjustment process and the identification of subtasks allow for the acquisition of extended scenes in real-time guided by geometric reasoning.

To demonstrate the feasibility of the approach and the usability of the methods, we created a computer program featuring editing possibilities. The latter is decisive since we cannot expect all automatically drawn decisions to be correct in terms of "intended". The screenshot is shown in Figure 12 illustrates a construction following Desargues's theorem in its affine form: The dependent constraint is drawn in gray. This example shows that the approach is not limited to the acquisition outlines—it can be used for constructions in general.

We evaluated the reasoning results with the help of results obtained with conventional tools and ground truth. We archived comparable results concerning completeness and conclude that if the completeness is high, the accuracy will be so, too.

6.2 Outlook

For the future, further investigations and experiments are required to foster the acceptance of the approach and to document the saving of labor and working time. This includes the considerations of different users and further data sets.

ACKNOWLEDGMENT

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

We thank the anonymous reviewers for their valuable comments and suggestions which significantly improved the quality of the article.

REFERENCES

- Beder, C., 2004. Fast statistical geometric reasoning about uncertain line segments in 2D- and 3D-space. In: *Pattern Recognition*, Lecture Notes in Computer Science, 3175, 375–382.
- Brenner, C., 2005. Constraints for modelling complex objects. In: U. Stilla and F. Rottensteiner (eds), *Int. Arch. Photogramm. Rem. Sensing*, 36-3/W24, 49–54.
- Buchberger, B., Collins, G.E., Kutzler, B., 1988. Algebraic methods for geometric reasoning. *Annual Review of Computer Science*, 3, 85–119.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. *Introduction to Algorithms*. 3 edn, The MIT Press.
- Cramer, M., 2010. The DGPF test on digital aerial camera evaluation – Overview and test design. *Photogrammetrie – Fernerkundung – Geoinformation*, 2, 73–82.
- Faugeras, O., 1993. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA, USA.
- Förstner, W. and Wrobel, B.P., 2016. *Photogrammetric Computer Vision*. Springer.
- Health, T. L. and Euclid, 1908. *The Thirteen Books of Euclid's Elements*, 1, Cambridge University Press.
- Heuel, S., 2004. *Uncertain Projective Geometry. Statistical Reasoning in Polyhedral Object Reconstruction*. Lecture Notes in Computer Science, 3008.
- Johnson, G.G., Do, E.Y., Gross, M.D., Hong, J.I., 2014. Sketch it, make it: Freehand drawing for precise rapid fabrication. Technical Report CMU-HCII-14-103, Human-Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.
- Koch, K.-R., 1999. *Parameter Estimation and Hypothesis Testing in Linear Models*. 2 edn, Springer.
- McGlone, J.C., Mikhail, E.M., Bethel, J. (eds), 2004. *Manual of Photogrammetry*. 5th edn, American Society of Photogrammetry and Remote Sensing.
- Meidow, J., 2014. Geometric reasoning for uncertain observations of man-made structures. In: X. Jiang, J. Hornegger and R. Koch (eds), *Pattern Recognition*, Lecture Notes in Computer Science, Vol. 8753, Springer International Publishing, pp. 558–568. 36th German Conference on Pattern Recognition, GCPR 2014, Münster, Germany, September 2–5, 2014.
- Meidow, J., Beder, C., Förstner, W., 2009. Reasoning with Uncertain Points, Straight Lines, and Straight Line Segments in 2D. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(2), 125–139.
- Neyman, J., Pearson, E.S., 1933. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of The Royal Society*, 231, 289–337.
- Nowozin, S., 2014. Optimal decisions from probabilistic models: The intersection-over-union case. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 548–555.

QGIS Development Team, 2009. QGIS Geographic Information System. Open Source Geospatial Foundation. <http://qgis.org>.

Richter-Gebert, J., Kortenkamp, U.H., 2012. *The Cinderella. 2 Manual*. Springer.

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

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