Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:
2020-07-07

Deposited version:
Post-print

Peer-review status of attached file:
Peer-reviewed

Citation for published item:
Brandão, F., Paio, A. & Lopes, A. (2020). Triangulation algorithms for generating as-Is floor plans. Nexus Network Journal. N/A

Further information on publisher's website:
10.1007/s00004-020-00491-3

Publisher's copyright statement:
This is the peer reviewed version of the following article: Brandão, F., Paio, A. & Lopes, A. (2020). Triangulation algorithms for generating as-Is floor plans. Nexus Network Journal. N/A, which has been published in final form at https://dx.doi.org/10.1007/s00004-020-00491-3. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0
The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Serviços de Informação e Documentação, Instituto Universitário de Lisboa (ISCTE-IUL)
Av. das Forças Armadas, Edifício II, 1649-026 Lisboa Portugal
Phone: +(351) 217 903 024 | e-mail: administrador.repositorio@iscte-iul.pt
https://repositorio.iscte-iul.pt

Triangulation Algorithms for Generating As-Is Floor Plans

ABSTRACT

Precisely capturing context is a fundamental first step in dealing with built environments. Previous research has demonstrated that existing methods for generating as-is floor plans of non-orthogonal rooms by non-expert users do not produce geometrically accurate results. The present paper proposes the adaptation of empirical triangulation methods, traditionally used by architects and other building professionals in surveying building interiors, to the development of semi-automated workflow of room survey. A set of triangulation algorithms that automate the plan drawing stage are presented.

KEYWORDS

Building renovation, As-is plans, Polygon partition, Triangulation algorithms, Computational geometry

INTRODUCTION

Architects and other building professionals are increasingly working on interactive digital representations of objects through information modelling software that gives prominence to the collaborative and decision-making aspects of the design process (Carpo 2011; Marble 2012). Concurrently, the idea of using digital design systems to provide context specific solutions to generic clients, originally envisioned by Yona Friedman (1971) and Nicholas Negroponte (1976), is being once again brought to the fore by developments in mass-customization systems in architecture (Kolarevic and Duarte 2019). Context specific solutions must link variation with the physical, socio-cultural, technological and individual context of the user (Kolarevic and Duarte 2019). Literature on mass customization in architecture places little focus on processes for users to provide information on their physical context, which is a fundamental first step for mass customization in building renovation. The previously stated problem of capturing built context can be viewed as a geometric and semantic problem of modelling the as-is condition of the surveyed space. Research on automatic as-is BIM methods is ongoing (Pătrăucean et al. 2015; Becker et al. 2019). Workflows involving photogrammetry or 3D laser scanning have gained popularity in the architecture, engineering and construction industry as a process of capturing point-clouds of existing structures for BIM modelling (Xiao and Furukawa 2012; Jung et al. 2014). While extremely precise and rich in detail, these methods are computationally intensive, mostly manual, time consuming and require advanced modelling knowledge (Brilakis et al. 2010; Tzedaki and Kamara 2013).

The present paper was developed in the context of research on a demountable and mass customizable construction system of partition walls for building renovation by building owners. It is composed of a construction system, a parametric model for digital design and fabrication and a graphical user interface. The interface must allow a user to design solutions for an existing space. To generate a design solution for building renovation, the shape of the space boundaries needs to be input to the design system. Specifying height, length and thickness of the wall is not enough to capture boundary conditions. Also, the complexity of the planning process grows as the number of generated walls increases. Ideally, the building owner should be able to survey the space to be partitioned, input this to the model and design the partition walls. The present paper focuses on the geometrical and mathematical foundation of triangulation algorithms for generating accurate as-is plans of convex or non-convex rooms by building owners.

Architects and construction professionals deal with this problem daily. Using empiric methods, a surveyor or an architect sketches the existing space and takes the needed field measurements for latter geometrically reconstructing plans, sections and elevations. In orthogonal spaces, measuring the perimeter of the space and registering the turn, left or right, at each corner is enough to accurately reconstruct the space. Several commercial applications, such as Magicplan (IOS 12.1, version 7.2.2) and RoomScan (IOS 12), leverage these empiric methods to provide semi-automated workflows for generating as-is models onsite involving user interaction, smartphone sensors and camera. These methods are fast and cost-effective for as-is surveys of a few rooms. But, if the rooms are not ortho-polygons, the results are geometrically inaccurate, since no triangulation strategy is provided (Brandão and Paio 2019). Experienced surveyors circumvent the non-orthogonal room problem by using triangulations between internal corners, subdividing the room's polygonal shape into triangles. The hypothesis is that their processes can be captured in a digital workflow, that users can execute them on their own, and that the result produced is a sufficiently accurate parametric model of the surveyed space for further design steps (Fig. 1).This problem is similar to the classical art gallery problem (AGP) (de Berg et al. 2008; de Rezende et al. 2014), albeit with a significant difference, the actual geometry of the polygon is undefined a priori, consequently triangulations are a means to define the polygon, i.e., close it.

Fig. 1 Semi-Automated workflow for user produced as-is plans

This paper starts by framing context of the problem. Methodology is discussed in Sect. 2. Then, in Sect. 3, we introduce algorithms to automate the process of determining the internal angles of the polygonal chain with user-provided diagonals. In Sect. 4 we demonstrate how these may be applied to determine the geometry of an existing space with three different case studies. In Sect. 5, some limitations of the algorithms are discussed and finally a general conclusion is drawn with some future research directions.

1 RESEARCH BACKGROUND

Advocacy for larger user involvement in the architectural design process can be traced back to the 1960s, with Christopher Alexander, John Habraken, Yona Friedman and Nicholas Negroponte being some of the most vocal supporters. Both Friedman and Negroponte proposed the use of computers to enable the user to design his own environment. In fact, a core pursuit of the research developed at the Architecture Machine Group was enabling user participation in the decision-making process of spatial configuration (Ameijde 2019). An example of that pursuit was the application YONA: Architecture-by-Yourself (Weinzapfel and Negroponte 1976), a computer implementation of Friedman's design methods. Friedman had previously also pursued similar ideas with the Flatwriter (1971), a machine for users to design their own houses. In the twenty-first century, the architecture, engineering and construction industry is open to increasing user involvement in the design and procurement process. Thus, users take on some tasks that were previously ascribed to professionals. This is made possible by the development of frameworks that encode design, professional or technical knowledge with which users engage (Duarte 2005; Brown 2016; Khalili-Araghi and Kolarevic 2020). However, the step of capturing context was either simplified or left to be pursued at a later stage.

Some commercial applications provide semi-automated workflows for generating as-is floor plans involving user interaction, phone sensors and camera. The 2D parametric models are approximate and require the user to introduce wall lengths into the model in a second step. Filipe Brandão and Alexandra Paio (2019) have shown that these workflows only produce sufficiently accurate results for digital fabrication if the room plan is an ortho-polygon, which is frequently not the case with old buildings in historic neighbourhoods. Otherwise the results are always indeterminate as there is no way to define the internal angles directly, by inputting angular dimensions, or indirectly by using diagonals. However, these methods require simple inputs from users which makes them an appealing option for mass-customizable construction systems for non-experts. Triangulation of simple polygons – that is, all non-self-intersecting polygons – is a classic problem of computational geometry with applications in computer graphics and optimization problems such as the placement of surveillance cameras or lights (de Berg et al. 2008). Algorithms for partitioning the polygon into monotone pieces were proposed by Garey et al (1978) and Bernard Chazelle (1982), who later proposed a deterministic linear time algorithm for any simple polygon (Chazelle 1990). Yet, these algorithms are meant to partition a polygon whose boundary is known à priori, which is not the case with our problem. Thus, triangulation in our case is a means to find the internal angle between edges with known lengths. Ear clipping (ElGindy et al. 1993) or divide-and-conquer (Bernard Chazelle and Incerpi 1983) strategies could be adequate approaches but since room clutter may prevent certain diagonals to be measured by the user, a non-deterministic approach is better suited. Furthermore, the limitation to non-intersecting sets of diagonals is not required in our case.

2 METHODOLOGY

To address the previously stated problem we looked at the empiric workflows of architects in surveying and designing as-is surveys. We identified several strategies that can be translated into algorithms: diagonal selection, iterative triangulation and polygon closure. The latter two strategies relate to the drawing process of the plan while the first is based on heuristics for onsite measurement. In this article we will focus on the strategies for drawing the polygon with measured diagonals. Our approach is based on the following assumptions: room corners are tendentially orthogonal (Steadman 2006), users of the system are able to represent the room plan with the correct number of sides and differentiate convex from concave corners, users cannot directly measure angles on-site and they tend to represent rooms as orthogonal. Thus, the user-provided polygon P_u is an approximation or a sketch of the actual room geometry, the planar polygonal shape P (Fig. 2). Consequently, the algorithms should rule out the following: a convex angle to turn into a concave one; zero or negative side lengths; or that the sum of any two consecutive edges is equal to or smaller than their diagonal.

In order to lay out a more formal framework, it is useful to start by defining corner type. A corner $P_u[i]$ is of the type convex if the line segment from $P_u[i-1]$ to $P_u[i+1]$ is entirely within the polygon P_u. Elsewhere in the literature the vertex is also called an "ear" of the polygon (ElGindy et al. 1993). Conversely, if such segment lies entirely outside P_u, the corner $P_u[i]$ is concave. Hence, a non-convex polygon is a closed polygonal chain that contains corners of both types.

Another concept worth formalizing relates to similarity of polygonal shapes. We say that P_u and P are morphologically similar if and only if both have *n* corners and there is a polygonal chain in P_u starting in the corner $P_u[i]$ and ending $P_u[i-1]$ such that the sequence of corner types is exactly that of the polygonal chain from $P[0] to\ P[5]$. Figure 2 depicts the concept.

Fig. 2 P is the actual plan of the room, and P_u is a polygon with arbitrary side lengths and a similar morphology to P

3 ALGORITHMICS

Having set up the underlying framework, the current section presents algorithms for iterative triangulation and polygon closure. The objective is to generate P_{out} , an approximation of P, given the following user inputs:

1. P_u, a morphologically similar closed polyline to P, with arbitrary side lengths, sketched by the user;
2. The length l of each edge of P measured on-site by the user, supplied as counter-clockwise ordered list L_l;
3. A list L_d of iteratively requested diagonals by the algorithm while P_{out} is not closed.

Let P_u be a closed counter-clockwise oriented polygonal chain representing a polygon with n vertices. The list contains n+1 points, such that $P_u[0]$ is the first or starting point and $P_u[n] = P_u[0]$. From this list of points, we may compute a generalized version of the polyline P_u in the form of a list of vectors P_{vec} with length n. With these vectors, a starting point and lengths l we may iteratively construct the polyline P_{out}, an approximation to P. P_{out} is said to be closed if the sum of all vectors in P_{vec} is approximately 0:

$$f = \sum_{i=0}^{n-1} v_i \approx 0.$$

Each v in P_{vec} may be scaled independently while keeping the polygon closed if and only if the sum of all vectors remains approximately 0. In this instance the resulting polygon P_{out} is assumed to have corners with angular dimensions approximating those of the polygon P and the algorithm terminates. It is important to note that this assumption is not always correct, and the implications will be discussed below. If on the other hand $|f| > 0$, the transformations of each vector in P_{vec} generates an open polygonal chain P_{out} with the same internal angles as P_u. Thus, one or more internal angles of P_u are different from those of P.

Iterative triangulation

The internal angles of P cannot be inferred from the dimensions of its edges unless P is cyclic, which is an unwarranted constraint. Also, we assume the user has no available tools to directly measure the room internal angles on-site. Our approach is to request valid diagonals of P_u and use them to rotate the vectors in P_{vec}. As not all diagonals of P_u are necessarily diagonals of P, or vice-versa, we provide the user with the option to skip to the next diagonal. Furthermore, this option is also useful if it is not practical to measure the diagonal onsite, because of room clutter, for example. To assist the triangulation process and the selection of diagonals while processing, we create a list V_s that contains the state of triangulation of each vector or internal corner of the polygon. There are two processes by which internal corners of the polygon can become triangulated by user-provided diagonals: polygonal chain closure and triangulation. Polygonal chain closure applies the previously described polygon closure test and is useful for ortho-polygonal chains. Let the indices of the endpoints of the user-provided diagonal in P_u be i, j with $i < j$, and $L_d d$ its length (Fig. 3). The ij and/or ji polygonal chains are closed if and only if the length of the sum of all vectors in P_{vec} between i and $j-1$ or j and $i-1$ is similar to $L_d d$. In this instance the internal angles of P_u from $i+1$ to $j-1$ are similar to those of P and can be set to triangulated in V_s.

Fig. 3 Example of polygonal chain closure on the ij chain

In the following explanation consider Fig. 4. Triangulation may occur if there is only one corner left to triangulate on the polygonal chain from $i + 1$ to $j - 1$ or from $j + 1$ to $i - 1$. Let the non-triangulated corner be k, and the triangulated and non-triangulated corners be t and nt, respectively.

Fig. 4 An example of a possible triangulation sequence using each of the described triangulation cases

There are three cases worth noting:

- Case 1: $i + 2 = j$, the diagonal from i to j forms a triangle with two polygon edges, and k, the third corner of the triangle is also a corner of the polygon. Whatever the state of the triangulation of the corners i and j, the internal angle of the vector ik with kj may be found with the law of cosines (Fig. 4). It is important to note that this type of triangulation may only occur on the polygon's ears or convex corners, since otherwise the diagonals are external and cannot be measured in practice.
- Case 2: i or j are not triangulated, k is next to the non-triangulated i or j and $j - i > 2$. This triangulation enables the growth of the triangulated sequence of corners (Fig. 4) and may also be used to triangulate concave corners (Fig. 5).
- Case 3: k is an isolated non-triangulated corner such that $k - 1$ or $k + 1$ are triangulated. We call this Pattern101 and this is the quickest process by which a concave corner may be triangulated (Fig. 4).

Fig. 5 An example of a j on the right of the infinite line from *i* to *k* in the triangulation of the *ij* chain

These cases may be used to accelerate the convergence of the triangulation. However, the triangulation of all cases may be processed with a single algorithm. Next, for the sake of brevity, we present a general triangulation algorithm for the ij chain. Suppose all corners from $i + 1$ to $j - 1$ are triangulated except one. Let the non-triangulated corner in the ij chain be k such that $i < k < j$. We can triangulate k by rotating all the vectors from k to the next non-triangulated corner.

Algorithm 01: $(P_{vec}, i, j, k, L_d d, V_s)$

Input: A list of vectors P_{vec} with length n. The indices i, j of the diagonal end points, the index k of the non-triangulated corner with $i < k < j$, the length $L_d d$ of the diagonal, and a list V_s with the state of the triangulation of the polygon's corners.

Output: P_{vec}.

```
1. for $a \leftarrow i \ to \ k$
2.         $\overrightarrow{ik} = \overrightarrow{ik} + P_{vec}[a]$
3. for $b \leftarrow k \ to \ j$
4.         $\overrightarrow{kj} = \overrightarrow{kj} + P_{vec}[b]$
5. Calculate the angle β of the vector $\overrightarrow{kj}$ with $\overrightarrow{ik}$ so that $|\overrightarrow{ik}|$, $|\overrightarrow{kj}|$, $L_d d$ are the lengths of the sides of a triangle
6. Define the new $\overrightarrow{kj_1}(x_1, y_1)$ vector as the rotation of the unit vector $\overrightarrow{ik}(x, y)$ by $\pi - \beta$ and scaling by $|\overrightarrow{kj}|$: $\overrightarrow{kj_1}(x_1, y_1) = R_{ik}(\pi - \beta) * |\overrightarrow{kj}|$
7. if the point $P_u[j]$ is right of the infinite line that goes through $P_u[i]$ and $P_u[k]$
8.         Reflect $\overrightarrow{kj_1}$ about the plane defined by the vector $\overrightarrow{ik}(x, y)$ and the normal vector $p(x_2, y_2)$ such that $p(x_2, y_2) = R_{ik}(\pi)$
9. Reverse $\overrightarrow{kj_1}$
10. Define $\theta$ as the angle of $\overrightarrow{kj}$ with $\overrightarrow{kj_1}$
```

```
Search $V_s$ for the index $l$ of the last fixed corner starting in $k$
if (> $k$)
    for $f \leftarrow k\ to\ l$
        $P_{vec}[f](x,y) = R(\theta_1)$
else
    for $f \leftarrow k\ to\ n$
        $P_{vec}[f](x,y) = R(\theta_1)$
    for $g \leftarrow 0\ to\ l$
        $P_{vec}[g](x,y) = R(\theta_1)$
```

A general algorithm for both chains can be easily obtained by testing to see if $k > j$ and splitting the vector sums accordingly. The triangulation algorithm we presented is used on ij chain of the polygon and the diagonal from i to j is user provided so intuitively it might seem that j should not be on the right side of the infinite line from i to k. Figure 5 presents an example of one such case. Assume that corners 3 and 4 are triangulated while corner 2 is not. The diagonal from 1 to 5 is an application of Case 2 and can be used to triangulate the corner 2.

On polygon closure

Using algorithm 01, the convergence of P_{out} into P terminates when n-1 elements of V_s are triangulated. If only one internal corner of P_{out} is not triangulated, then there is only one triangulated polygonal chain. Since a triangulated polygonal chain is rigid, and assuming absolute precision of the user-provided diagonals, P_{out} is a closed polygon. In practice, the user provided diagonals will contain measurement errors. Consequently, an algorithm for interactive triangulation will have to provide a strategy for closing the polyline in the presence of closing error.

The convergence of P_{out} into P may be terminated earlier without user input if any two or three corners are not triangulated. We call these situations Pattern00 and Pattern000. Figure 6 is an example of the application of Pattern00 during the triangulation of a polygon. In Step 1 the user measured lengths of P are applied to the sides of the polygon P_u. As the polygon is not closed, L_d1 – the length of the diagonal from corner 0 to 2 – is requested. This diagonal successfully triangulates corner 1, rotating $\overrightarrow{12}$ so that $L_d1 = |\vec{ij}|$. Furthermore, as the polygonal chain from 2 to 0/6 is orthogonal, the corners 3, 4 and 5 are set to triangulated. However, the polygon is not yet closed since the corners 2 and 0 remain non-triangulated (Fig. 6, Step 2). Since $|\vec{ij}| = |\vec{ji}|$, we may rotate all the vectors on the polygonal chain ij so that $\vec{ij} = -\vec{ji}$.

Fig. 6 An example of triangulation of a polygon terminating with Pattern00 algorithm

Algorithm 02 implements this strategy: let i be the index of the first undefined corner and j the index of the second one (see Fig. 6).

Algorithm 02: Pattern00(P_{vec}, i, j)

Input: A list of vectors P_{vec} with length n. The indices i, j of the non-triangulated corners with $i < j$

Output: A triangulated list of vectors P_{vec}.

```
for $a \leftarrow i\ to\ j$
    $\vec{ij} = \vec{ij} + P_{vec}[a]$
for $b \leftarrow j\ to\ n$
    $\vec{ji}. = \vec{ji}. + P_{vec}[b]$
for $c \leftarrow 0\ to\ i$
    $\vec{ji}. = \vec{ji}. + P_{vec}[c]$
Calculate the internal angle $\alpha$ of $\vec{ij}$ with $-\vec{ji}$
```

```
Rotate all vectors from $i$ to $j$ in $P_{vec}$ by $\alpha$
```

If a Pattern000 is found, there are only two solutions for the internal angle of the middle corner. As we assume that the polygon provided by the user is correct in terms of left or right turns in the corners, we simply maintain the observed turn. Figure 7 presents a triangulation process ending with an application of the Pattern000 algorithm. Suppose that all internal corners have been triangulated by the diagonals L_d1, L_d2, L_d3, except corner 0, 3 and 6 (Fig. 7, Step 1). Let corner 0 be i, corner 3 be j and corner 6 be k. The angle at j can be found with the law of cosines, using $|\overrightarrow{ij}|$, $|\overrightarrow{jk}|$, $|\overrightarrow{ki}|$ as sides of the triangle, to define $\overrightarrow{jk_1}$. We may then rotate all the polygon's sides from 3 to 5 by the internal angle of $\overrightarrow{jk_1}$ with $\overrightarrow{jk}$ to obtain the result in Step 2 in Fig. 7. A similar process may be applied to the corner k to arrive at Step 3.

Fig. 7 Application of the algorithm Pattern000. t is a corner that can be triangulated by a diagonal

Let the index of the three non-fixed corners found in the polygonal chain be i, j and k. Algorithm 03 implements the previously described strategy.

Algorithm03: Pattern000(P_{vec}, i, j, k)
Input: A list of vectors P_{vec} with length n. The indices i, j, k of the non-triangulated corner with $i < j < k$.
Output: A triangulated list of vectors P_{vec}.

```
for $a \leftarrow i \ to \ j$
    $\overrightarrow{ij} = \overrightarrow{ij} + P_{vec}[a]$
for $b \leftarrow j \ to \ k$
    $\overrightarrow{jk} = \overrightarrow{jk} + P_{vec}[b]$
for $c \leftarrow k \ to \ n$
    $\overrightarrow{ki} = \overrightarrow{ki} + P_{vec}[c]$
for $d \leftarrow 0 \ to \ i$
    $\overrightarrow{ki} = \overrightarrow{ki} + P_{vec}[d]$
Calculate the angle β of the vector $\overrightarrow{jk}$ with $\overrightarrow{ij}$ so that $|\overrightarrow{ij}|$, $|\overrightarrow{jk}|$, $|\overrightarrow{ki}|$ are the lengths of the sides of a triangle
Define the new $\overrightarrow{jk_1}(x_1, y_1)$ vector as the rotation of the unit vector $\overrightarrow{ij}_u(x_u, y_u)$ by $\pi - \beta$ and scaling by $|\overrightarrow{jk}|$: $\overrightarrow{jk_1}(x_1, y_1) = R_{ij}(\pi - \beta) * |\overrightarrow{jk}|$
if the point $P_u[k]$ is right of the infinite line that goes through $P_u[i]$ and $P_u[j]$
    Reflect $\overrightarrow{jk_1}$ about the plane defined by the vector $\overrightarrow{ij}(x, y)$ and the normal vector $\vec{p}(x_2, y_2)$ such that $\vec{p}(x_2, y_2) = R_{ik}(\pi)$
Reverse $\overrightarrow{jk_1}$
Define $\theta$ as the angle of $\overrightarrow{jk}$ with $\overrightarrow{jk_1}$
for $e \leftarrow j \ to \ k - 1$
    $P_{vec}[e](x, y) = R(\theta)$
$\overrightarrow{jk}$ = Reverse $\overrightarrow{jk_1}$
Calculate the angle $\beta_1$ of the vector $\overrightarrow{ki}$ with $\overrightarrow{jk}$ so that $|\overrightarrow{ij}|$, $|\overrightarrow{jk}|$, $|\overrightarrow{ki}|$ are the lengths of the sides of a triangle
Define the new $\overrightarrow{ki_1}(x_1, y_1)$ vector as the rotation of the unit vector $\overrightarrow{jk}_u(x_u, y_u)$ by $\pi - \beta_1$ and scaling by $|\overrightarrow{ki}|$: $\overrightarrow{ki_1}(x_1, y_1) = R_{jk}(\pi - \beta_1) * |\overrightarrow{ki}|$
```

```
if the point $P_u[i]$ is right of the infinite line that goes through $P_u[j]$ and $P_u[k]$
        Reflect $\overrightarrow{k\iota_1}$ about the plane defined by the vector $\overrightarrow{jk}$ $(x, y)$ and the normal vector $\vec{p}(x_2, y_2)$ such that $\vec{p}(x_2, y_2) = R_{jk}(\pi)$
Reverse $\overrightarrow{k\iota_1}$
Define $\theta_1$ as the angle of $\overrightarrow{ki}$ with $\overrightarrow{k\iota_1}$
for $f \leftarrow k\ to\ n$
        $P_{vec}[f](x,y) = R(\theta_1)$
for $g \leftarrow 0\ to\ i - 1$
        $P_{vec}[g](x,y) = R(\theta_1)$
```

In both instances, the algorithms 02 and 03 close the polygon and terminate the processing. Furthermore, they reduce the polygon's closing error caused by user measurement imprecision, more so with Pattern000. On the other hand, Pattern000 will fail if the $\overrightarrow{\iota j}, \overrightarrow{jk}$ and $\overrightarrow{ki}$ are collinear.

4 CASE STUDIES

To test the application of the algorithms we have described above, we present three case studies of triangulation of room plans. All these cases represent situations where the actual room has no right angles, which are the most challenging situations. Valid diagonals of the user-provided polygon are requested to the user in the following order:

1. Shortest diagonals of non-orthogonal corners with internal angles smaller that π;
2. The longest diagonals;

Each case presents the behavior of the algorithms with input polygons ranging from an ortho-polygon to a polygon with no orthogonal angles.

Case Study 1

The first case study is a plan of a concave room in a building to be renovated in Braga, Portugal with six sides and no orthogonal internal angles, the input polygon is an ortho-polygon. Figure 8 presents a complete triangulation process which ends with an application of the Pattern000 algorithm. The first step of process is determining if the user measured sides of P when applied to P_u sides generate a closed polygon (Fig. 8c). Since it is not closed and the provided polygon is an ortho-polygon, the longest diagonal is requested first. This diagonal allows the triangulation of corner 2, the second diagonal triangulates corner 1 and the third triangulates the third corner. At this stage there are only three corners left to triangulate, 0, 4 and 5, so the Pattern000 algorithm is used.

Fig. 8 Case Study 1: triangulation of a polygon terminating with Pattern000 algorithm

Case study 2

The second case study is a seven-sided non-convex room. The user-provided polygon has two non-orthogonal corners while all corners of the actual room plan are non-orthogonal. In the first step (Fig. 9c) scaling the sides of P_u with the lengths of the user-provided room sides will not close the polygon. Since P_u has two non-orthogonal corners, 1 and 2, diagonals that triangulate those corners ordered by ascending angular dimension are requested first, L_d1 and L_d2. After the triangulation of the non-orthogonal corners, the longest diagonals are requested. L_d3 triangulates corner 0 (Fig. 9e) and L_d4 triangulates corner 6 (Fig. 9f). The last step triangulates corners 3,4 and 5 using Pattern000.

Fig. 9 Case Study 2: triangulation of a room starting with a user-provided polygon with some non-orthogonal corners

Case Study 3

The third case study is a seven-sided non-convex room with two reflex corners where the user-provided polygon has no orthogonal angles. As with the previous case, P side lengths do not produce a closed polygon (Fig. 10c). Since all corners are non-orthogonal the diagonals are ordered by ascending angle dimension with the remaining longest diagonals at the end. $L_d 1$ and $L_d 2$ are diagonals of the most acute corners, 6 and 4, which are requested first (Fig. 10c, 10d). Since the triangulation of the corners 4 and 6 creates a Pattern101 at corner 5, the length of the diagonal $|\overrightarrow{04}|$ is requested next (Fig. 10e). Then, the diagonal of the third most acute corner, corner 2, is requested (Fig. 10f). The triangulation of corner 2 creates a Pattern000: corner 0, 1 and 3.

Fig. 10 Case Study 3: triangulation of a room starting with a user-provided polygon with no orthogonal corners

5 DISCUSSION AND CONCLUSION

The algorithms we have presented can be used to implement an interactive workflow of space survey with user-measured diagonals that mimics the empiric surveying workflows used by architects. We have demonstrated how they can expand existing application workflows, thus increasing their scope to a wider range of room geometries beyond ortho-polygons. The initial inputs are a morphologically similar closed polygon, with the correct number of sides, and the lengths of the room's sides. If scaling the user-provided polygon by the lengths does not produce a closed polygon, diagonals are requested following a simple heuristic. The provided measurement is then used to fix related vertices of the polygon, with the goal of achieving polygonal chain closure. This is an iterative process, until the polygon is closed within a specified tolerance.

The algorithms we propose encode the geometric expertise involved in solving triangulations at the drawing stage of empiric surveys. They can be leveraged to increase the accuracy of existing room survey mobile applications. Furthermore, the proposed workflow is a parametric model of the surveyed space based on proven geometric principles with univocal relations between inputs and outputs. The advantages are that the survey is explicitly defined and updatable, making it possible to create associative relations between the design and the underlying survey. For mass-customization systems this provides a process with which user-provided room surveys can be verified and improved upon by professionals at later stages. Naturally these advantages also apply to professional applications. For instance, updating the survey model as demolitions or partial construction occurs and having those changes propagate without the need to redesign. Optimization processes using terrestrial laser scanning point clouds as targets for the mapping of parametric models proposed in Heritage BIM (Chevrier et al. 2010; Murphy et al. 2013; Giordano et al. 2019) may also be used, extending the application of these processes to renovation in vernacular architecture or conventional construction.

Polygonal chain and polygon closure tests have an important limitation. When the existing room plan P is a shear transformation of an ortho-polygon, if the user provides an input polygon P_u that is an ortho-polygon, when it is scaled by L_l, the algorithm will not request diagonals, since the sum of all vectors in P_{vec} will be approximately 0. In practice, user and measuring device precision errors may reduce the likelihood of these instances.

In our case studies we have used a heuristic that favours longer diagonals on ortho-polygons, based on the assumption that room plans are mostly orthogonal. In future work, we intend to test this heuristic on a wider range of polygons and compare it with other alternatives with the aim of minimizing the number of requested diagonals. Also, although the case studies demonstrate that these methods can produce geometrically accurate room plans, user testing is required to determine if users can use them effectively and what the effect of measurement error on geometrical accuracy is.

In conclusion, this study highlights the fact that the empirical knowledge of architects and building professionals may be leveraged for the development of semi-automated workflows, in this instance, of accurate geometrical algorithms for semi-automated empiric surveys of room plans. Future research will involve implementing the algorithms in an application for building owners and a plugin for Grasshopper to assist users to survey non-orthogonal rooms and automating the drawing of the room plan. The latter step will provide the possibility of developing a low-key design interface which requires simple inputs for our mass-customizable system of demountable partition walls for renovation of rooms aimed at non-expert users.

Acknowledgments

This work was supported by a merit scholarship granted by ISTA – Escola de Tecnologias e Arquitetura of ISCTE – Instituto Universitário de Lisboa for the development of the first author's Ph. D. thesis and by Fundação para a Ciência e a Tecnologia (FCT) projects UIDB/04466/2020 e UIDP/04466/2020. The authors would also like to express their gratitude to Editor Kim Williams for her meticulous editing and suggestions. All images are by the author.

All images are by the author.

REFERENCES

Ameijde, Jeroen van. 2019. The Architecture Machine Revisited: Experiments exploring Computational Design-and- Build Strategies based on Participation. *SPOOL* 6. S.I.: 17–34. https://doi.org/10.7480/spool.2019.1.3890.

Becker, Ralf, Elisa Lublasser, Jan Martens, Raymond Wollenberg, Haowei Zhang, Sigrid Brell-Cokcan, and Jörg Blankenbach. 2019. Enabling BIM for Property Management of Existing Buildings Based on Automated As-is Capturing. In: *Proceedings of the 36th International Symposium on Automation and Robotics in Construction*, Banff. https://doi.org/10.22260/ISARC2019/0028.

Brandão, Filipe J. S., and Alexandra Paio. 2019. CONTEXT-AWARE MASS CUSTOMIZATION CONSTRUCTION SYSTEM: Methods for user captured as-built plans. In Intelligent & Informed - Proceedings of the 24th CAADRIA Conference, 2:101–110. Wellington, New Zealand: Victoria University of Wellington.

De Berg, Mark, Otfried Cheong, Marc van Kreveld, Mark Overmars. 2008. *Computational Geometry: Algorithms and Applications*, 3rd ed. Berlin: Springer.

Brilakis, Ioannis, Manolis Lourakis, Rafael Sacks, Silvio Savarese, Symeon Christodoulou, Jochen Teizer, and Atefe Makhmalbaf. 2010. Toward automated generation of parametric BIMs based on hybrid video and laser scanning data. *Advanced Engineering Informatics* 24: 456–465. https://doi.org/10.1016/j.aei.2010.06.006.

Brown, John L. 2016. Going Home: Future Adaptive Building for Aging-in-Place. PhD thesis, Melbourne, RMIT University.

Carpo, Mario. 2011. *The Alphabet and the Algorithm*. Cambridge: MIT Press.

Chazelle, Bernard. 1990. Triangulating a simple polygon in linear time. In *Proceedings*

[1990] 31st Annual Symposium on Foundations of Computer Science, 220–230. St. Louis, MO, USA: IEEE Comput. Soc. Press. https://doi.org/10.1109/FSCS.1990.89541.

Chazelle, Bernard. 1982. A theorem on polygon cutting with applications. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, 339–349. Chicago, IL, USA: IEEE. https://doi.org/10.1109/SFCS.1982.58.

Chazelle, Bernard, and Janet Incerpi. 1983. Triangulating a polygon by divide-and-conquer. *Proceedings - Annual Allerton Conference on Communication, Control, and Computing*, eds. H Vincent Poor, Donna J. Brown; Tamer Başar, 447–456. Urbana-Champaign: University of Illinois.

Chevrier, Christine, Nathalie Charbonneau, Pierre Grussenmeyer, and Jean-Pierre Perrin. 2010. Parametric Documenting of Built Heritage: 3D Virtual Reconstruction of Architectural Details. *International Journal of Architectural Computing* 8: 135–150. https://doi.org/10.1260/1478-0771.8.2.135.

Duarte, José Pinto. 2005. A discursive grammar for customizing mass housing: The case of Siza's houses at Malagueira. *Automation in Construction* 14: 265–275. https://doi.org/10/dsdwg8.

ElGindy, Hossam, Hazel Everett, and Godfried Toussaint. 1993. Slicing an ear using prune-and-search. *Pattern Recognition Letters* 14: 719–722. https://doi.org/10.1016/0167-8655(93)90141-Y.

Friedman, Yona. 1971. The Flatwriter: Choice by Computer. *Progressive Architecture*, March 1971: 98-101.

Garey, Michael R., David S. Johnson, Franco P. Preparata, and Robert E. Tarjan. 1978. Triangulating a simple polygon. *Information Processing Letters* 7: 175–179. https://doi.org/10.1016/0020-0190(78)90062-5.

Giordano, Andrea, Paolo Borin, and Federico Panarotto. 2019. Digital Wood Trusses. Geometry and Parameters/Fabrication and Monitoring. In *Digital Wood Design*, eds. Fabio Bianconi and Marco Filippucci, 24:479–509. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-03676-8_18.

Jung, Jaehoon, Sungchul Hong, Seongsu Jeong, Sangmin Kim, Hyoungsig Cho, Seunghwan Hong, and Joon Heo. 2014. Productive modeling for development of as-built BIM of existing indoor structures. *Automation in Construction* 42: 68–77. https://doi.org/10.1016/j.autcon.2014.02.021.

Khalili-Araghi, Salman, and Branko Kolarevic. 2020. Variability and validity: Flexibility of a dimensional customization system. *Automation in Construction* 109: 102970. https://doi.org/10.1016/j.autcon.2019.102970.

Kolarevic, Branko, and José Pinto Duarte, ed. 2019. *Mass Customization and Design Democratization*. New York: Routledge.

Marble, Scott, ed. 2012. *Digital Workflows in Architecture: Design, Assembly, Industry*. Basel: Birkhäuser.

Murphy, Maurice, Eugene McGovern, and Sara Pavia. 2013. Historic Building Information Modelling – Adding intelligence to laser and image based surveys of European classical architecture. *ISPRS Journal of Photogrammetry and Remote Sensing* 76: 89–102. https://doi.org/10.1016/j.isprsjprs.2012.11.006.

Pătrăucean, Viorica, Iro Armeni, Mohammad Nahangi, Jamie Yeung, Ioannis Brilakis, and Carl Haas. 2015. State of research in automatic as-built modelling. *Advanced Engineering Informatics* 29: 162–171. https://doi.org/10.1016/j.aei.2015.01.001.

de Rezende, Pedro J., Cid C. de Souza, Stephan Friedrichs, Michael Hemmer, Alexander Kröller, and Davi C. Tozoni. 2014. Engineering Art Galleries. *arXiv:1410.8720* [cs.CG] .

Steadman, Philip. 2006. Why are most buildings rectangular? *arq: Architectural*

Research Quarterly 10: 119. https://doi.org/10.1017/S1359135506000200.

Tzedaki, Vasiliki, and John M. Kamara. 2013. Capturing As-Built Information for a BIM Environment Using 3D Laser Scanner: A Process Model. In *AEI 2013*, 486–495. State College, Pennsylvania, United States: American Society of Civil Engineers. https://doi.org/10.1061/9780784412909.047.

Weinzapfel, Guy, and Nicholas Negroponte. 1976. Architecture-by-yourself: an experiment with computer graphics for house design. In *SIGGRAPH '76: Proceedings of the 3rd annual conference on Computer graphics and interactive techniques*, 74–78. ACM Press. https://doi.org/10.1145/563274.563290.

Xiao, Jianxiong, and Yasutaka Furukawa. 2012. Reconstructing the World's Museums. *International Journal of Computer Vision* 10(3). https://doi.org/10.1007/s11263-014-0711-y.