Strokes2BIM: Generating an IFC Model from 4D Concept Design Sketches

Rasoulzadeh S., Stauss P., Reisinger J., Kovacic I., Wimmer M.
TU Wien, Austria
shervin.rasoulzadeh@tuwien.ac.at

Abstract. Throughout the concept design phase in the early stages of design in Architecture, Engineering, and Construction (AEC), architects rely on sketching, either with pen and paper or through digital means, for the exploratory ideation process. However, for the digital modelling phase, depending on the use case, they often need to utilise Computer-Aided Design (CAD) software or Building Information Modelling (BIM) authoring tools, or a combination of both to re-draft and re-model their designs. This re-work, along with the potential loss of information during translation, underscores the need for an automated pipeline that bridges the concept design and digital modelling phases by capturing design intention and reconstructing semantically enriched geometry from such sketches. To this end, this paper extends a machine-learning-based pipeline that reconstructs geometries from 4D architectural design sketches with a post-processing step that generates a semantically enriched shared model based on Industry Foundation Classes (IFC), enabling direct interoperability with BIM workflows already in the early stages of design.

1. Introduction

The early stages of design in Architecture, Engineering, and Construction (AEC) often involve two crucial phases: concept design and digital modelling. The concept design phase, which is highly iterative, relies heavily on sketching as a foundation for the architects or designers to proceed. In this phase, sketching helps to initiate, develop, and communicate ideas, through which the design form evolves. Traditionally, pen and paper used to dominate the ideation process throughout this phase, while with the recent advancements of applications and interfaces utilising 3D drawing canvases for projecting 2D strokes and immersive environments of Augmented and Virtual Reality (AR/VR) (Arora et al. 2018), sketch-pen turns into a stylus or a controller allowing designers to create and draw sketches in 3D space. Throughout the second phase, digital modelling, the sketch serves as a visual reference for the architect or designer to manually create the detailed geometric model and to further enrich it with desired semantics. In this phase, depending on the use case, the architect might use 3D Computer Aided Design (CAD) modelling software such as Blender or Rhino, or a Building Information Modelling (BIM) authoring tool such as ArchiCAD or Revit, or a combination of both so as to reach the desired outcome.

Within this context, although some sketching applications interpret and infer 3D geometry from 2D/3D sketches, they lack the required semantics and interoperability with BIM packages, often requiring the architect to re-draft or re-model in a BIM authoring tool. Additionally, this re-work results in potential information loss during the translation process while hindering the freedom that the architect has throughout the concept design phase. This is because CAD modelling software and BIM authoring tools push the design into a rigid BIM model and compel designers to think and operate in alignment with their own set of constraints. These challenges raise an idea of bridging between concept design and digital modelling through an automated pipeline which is not only responsible for reconstructing a geometric model but also enables BIM interoperability through generating a shared model based on open standards such as Industry Foundation Classes (IFC) which is widely used for the exchange of information in AEC disciplines.

Building upon the previous statements, this paper extends a machine-learning-based pipeline that interprets and reconstructs the 3D geometry of 4D architectural design sketches (Rasoulzadeh et al. 2023) with a post-processing step responsible for further semantic enrichment of the resultant geometry. The pipeline and the post-processing both operate in a completely offline process, meaning that they take place once the designer has decided that he/she is finished with sketching. Through the subsequent post-processing, the pipeline enriches the sketch with semantics by exploiting the structural and material properties attributed to the sketch's constituent strokes by the architect or designer, ultimately generating an IFC file. The pipeline is based on a 4D sketching application targeted for architectural design as a form-finding tool, which simulates traditional sketching behaviour with paper and pen while allowing sketch creation in 3D space via tablet and stylus whilst capturing and recording temporal data as of fourth dimension along with additional geometry and stylus-related properties with the aim of facilitating further sketch analysis (Kovács et al. 2022).

Such an extended pipeline not only improves the design efficiency by allowing the architect or designer to focus solely on the exploratory ideation process but also alleviates the need for re-work in a CAD modelling software or a BIM authoring tool, providing a semantically enriched digital model at the early stages of design. This enriched model can significantly impact the later stages of the AEC workflow by facilitating communication, enhancing collaboration, streamlining documentation, and enabling better-informed decision-making throughout the project lifecycle.

The rest of the paper is organised as follows. First, we begin with a review of relevant previous related work. Next, we describe the overall workflow and provide details of each of its components, including the 4D sketching application, geometry reconstruction pipeline and its following post-processing that enables IFC model generation. Subsequently, we present a proof of concept by showcasing the sketch of the Barcelona Pavilion by Mies van der Rohe (Kroll 2011) as it is processed through the geometry reconstruction pipeline. Finally, the paper is closed with conclusions and an outlook on future work.

2. Related Works

Our work is inspired by and builds upon several studies in the literature, encompassing digital sketching applications and interfaces, sketch beautification and consolidation, sketch-based modelling, and building information modelling, as briefly described next.

(Dorsey et al. 2007) introduced Mental Canvas which supports conceptual architectural design by allowing the designers to draw a set of sketches in 2D on planar canvases. These sketches can then be positioned in 3D space to form a three-dimensional representation. However, the tool neither interprets the sketch nor infers the 3D geometry. Sketch beautification and consolidation methods aim toward transforming input sketch strokes into aggregate geometric curves that accurately represent designers' intentions. (Orbay et al. 2011) proposed a method that parses the digitally created 2D strokes into beautified segments by identifying stroke groups using a supervised clustering model. Furthermore, in the realm of sketch-based modelling, BuildingSketch (Lie et al. 2021) provides a VR modelling system that generates building blocks using a deep neural network by interpreting a few 3D strokes and translating them into primitive types (e.g., building mass, roof, stair). Vitruvio (Tono et al. 2022) is a conditional deep generative model that creates 3D building meshes from a single 2D perspective sketch. Their model generates a single watertight mesh, and they suggest producing the BIM as a post-processing step using the overall mass of the mesh as a starting point to extract floor and facade information. In relation to BIM, (Langenhan et al. 2013) proposed a method for querying building information models with a sketch-based input to retrieve previously designed and stored buildings designs to support the early stages of design.

To the best of authors' knowledge, no existing work in the literature addresses automated geometry reconstruction of architectural design 3D/4D sketches while providing direct interoperability with BIM workflows.

3. Methodology

To ensure that the designer has complete freedom throughout the concept design phase and that the come-along pipeline bridging it to digital modelling can interpret the sketch and reconstruct 3D geometry while outputting an IFC file, it is essential to pay special attention to certain aspects, as described in the following.

Firstly, architects generally prefer not to have real-time beautification or neatening of their sketched strokes, opting for the reconstruction pipeline to take place after the drawing is completed. Secondly, as described in Strokes2Surface (Rasoulzadeh et al. 2023) observations reveal that architects primarily use two sets of strokes to communicate their design ideas: *Shape Strokes*, which outline the boundaries and edges of the intended geometry, and *Scribble Strokes*, which mark the enclosed areas that form faces of the intended geometry. Moreover, we noticed that most designers tend to draw planar building elements, such as ramps, roofs, slabs, and walls, with single faces, while linear building elements like beams and columns are often drawn with a single stroke with a specified brush width (see Figure 1). Lastly, in order to extend the geometry reconstruction pipeline with a post-processing step for IFC model generation, certain properties recorded via the MR.Sketch (Kovács et al. 2022), the sketching interface, are exploited. These comprise geometry- and stylus-related properties such as stroke ink colour and stroke ink width, along with additional semantics, like material and structural properties, that the architect attributes the sketched strokes.

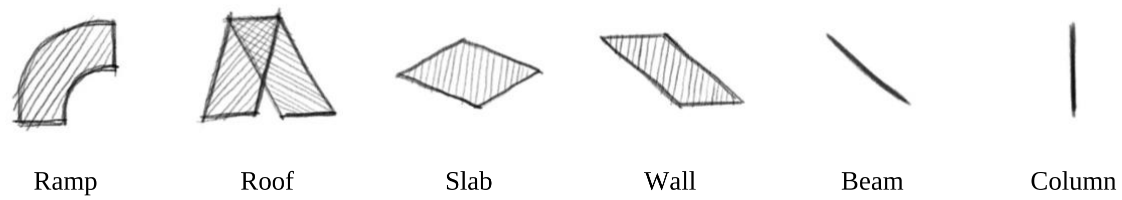


Figure 1. Examples of how planar and linear building elements.

Building on the aforementioned information, we extend the Strokes2Surface pipeline built on top of the 4D sketching application MR.Sketch with a post-processing step. The pipeline is responsible for reconstructing the intended geometry of the architect or designer once they have finished sketching. Through the post-processing step, we further enrich the reconstructed geometry by leveraging the pre-defined properties semantics coupled to the strokes by the designer to generate an IFC model, and save it. In the following subsections, we delve deeper into the details of the sketching application, geometry reconstruction pipeline, and subsequent post-processing step for the generation of the IFC file.

3.1 4D Sketching Interface

The pipeline is based on a 4D sketching application named MR.Sketch targeted for architectural design (Kovács et al. 2022) utilising a tablet (iPad) and a stylus (Apple Pencil) as its primary drawing medium. The application allows the creation of 4D sketches by employing 3D drawing canvases and projecting 2D stroke coordinates onto them while simultaneously capturing temporal information throughout the drawing process coupled as of the fourth dimension to strokes' polyline vertices.

The interface is composed of a ground plane within its scene and offers various geometric primitives as canvases, including plane, cube, sphere, and cylinder, which can be arbitrarily transformed in the scene. Architects or designers have control over the camera view's position and rotation, allowing them to freely navigate the 3D space of the scene. Once the canvas is in the desired transform, the designer can lock it in place and begin drawing from their current viewport. In this way, as the designer draws strokes on the tablet's surface, the ray originating from the camera view is intersected with the canvas and the resulting 3D point is stored as the continuation of the being drawn stroke polyline positions, forming a 3D stroke. It is noteworthy that the designer is not just limited to using one single canvas; the overall sketch can be drawn using multiple canvases of the same or different types in combination with each other, allowing for the sketching of complex design ideas. Also, regarding scale measurement, the designer has the option to enable the 3D grid of cells of size $1m^3$ spanning the interface's scene to serve as a guide when sketching. See Figure 2 for a sample sketch drawn using MR.Sketch.

(a) (b)

Figure 2. Sketch drawn by Philipp Stauss using MR.Sketch while taking the Buga Wood Pavilion (ICD/ITKE University of Stuttgart 2019) as a visual reference. (a) Depicts the top-view and (b) shows the perspective view of the sketch. Images are rendered using Polyscope (Sharp 2019) using the data recorded in the exported JSON file from MR.Sketch.

Along with temporal data, which could carry useful information relating to the design intent, several other geometry and stylus-related properties are also attributed and recorded throughout the sketching, which could also benefit the come-along analyses and processes afterwards. These attributes are stored within two different levels, namely per strokes and per stroke polyline vertices; per stroke properties are named *inkWdith*, *inkColour*, *cameraViewPosition*, *cameraViewRotation*, *canvasID*, *canvasTransform*, *materialInfo*, and *structuralInfo*. The materialInfo and structuralInfo properties associated with each stroke represent the material and type of building element that the designer envisions the stroke is going to partially or entirely depict. Also, as for each stroke polyline vertex, properties such as *timestamp*, *position*, *normal*, *tilt*, *twist*, and *pressure* are recorded. Once the sketching is finished, all this recorded information can be exported into a single JSON file for further sketch analysis.

Utilising the aforementioned properties, the geometry reconstruction pipeline infers the design intent and reconstructs the 3D geometry and subsequently employs materialInfo and structuralInfo properties to semantically enrich the reconstructed geometry, facilitating the generation of an IFC file.

3.2 Geometry Reconstruction

The nature of data recorded within the sketching application offers a promising opportunity for utilising machine learning to infer the design intent and reconstruct geometry. To this end, we use Strokes2Surface (Rasoulzadeh et al. 2023) pipeline to reconstruct the geometry. According to the observations of how designers sketch in MR.Sketch (Shape versus Scribble strokes, as mentioned earlier), the pipeline processes sketched strokes. This pipeline consists of three machine learning models: one classification model and two clustering models. These models are responsible for recognising stroke types and subsequently clustering strokes of each type into groups, each representing an edge or a face of the intended geometry. Next, groups representing edges are approximated using cubic B-spline curves and further processed for the creation of a well-connected curve network. Then, groups indicating enclosed areas and faces defined by edges are utilised to create bounding patches corresponding to their respective cycles within the curve network, ultimately outputting the final reconstructed geometry.

Shape vs. Scribble Stroke Classification

When drawing strokes, designers often represent intended edges or enclosed faces by using multiple, tightly clustered or overdrawn strokes that may lack proper inter-stroke connectivity. Utilising the properties recorded and coupled with each drawn stroke in MR.Sketch, several features such as average pressure, average speed, density, straightness, etc., are defined and further examined quantitatively and qualitatively so as to determine how each of these features evolve throughout drawing among these two types of strokes in comparison to one another. The trends indicate that these properties differ notably when designers draw strokes representing edges compared to when they use strokes to fill in enclosed areas forming faces.

Strokes2Surface trains and uses random forest classifier (Breiman 2001) to predict stroke types using the features defined earlier on a manually labelled collected dataset of architectural design 4D sketches created with MR.Sketch. with 80 % data for training and 20 % for testing. Given a new unseen sketch, once each of its constituent strokes are classified, they are further clustered into different subgroups within two clustering models. Shape strokes are clustered into distinct groups, with each group ideally comprising strokes that depict single edge of the geometry faces. In contrast, Scribble strokes are placed into separate groups with each group representing a single face of the geometry. This process prepares the Shape strokes groups for curve approximation and topology recovering, ultimately forming a curve network. Concurrently, the clustered Scribble strokes guide the determination of which cycles in the final curve network should create bounding patches.

Clustering Shape Strokes

To compute the similarity score for clustering Shape strokes, strokes' polyline points and strokes' ink width (brush width) are used to compute the overlapping sequence(s) of points for every pair of shape strokes. For each pair of strokes S_i and S_j, let P_i and P_j be the set of points along their corresponding polylines. All pairs of consecutive point sequences along each stroke whose distance is at most $2.5 \times \frac{w_i+w_j}{2}$, are computed and matched with the other, where w_i, and w_j are the stroke ink widths of strokes S_i and S_j, respectively. Subsequently, we obtain set of matching sequences, $\mathcal{M}_{i,j} = \{M_1, \cdots, M_{|\mathcal{M}|}\}$ where each M_k comprises of a subset of ordered points along of P_i and P_j that are matched. For each matching M_k, let $t_{k,i}$ and $t_{k,j}$ be the average tangents along the corresponding subset of points in P_i and P_j, respectively. Using the dot product of the average of tangents for each such matching sequences the set $\{t_{1,i} \cdot t_{1,j}, \cdots, t_{|\mathcal{M}|,i} \cdot$

$t_{|\mathcal{M}|,j}\}$ is obtained. To compute the similarity score we take average of these dot products over all matching matchings M_k, ending up with a number between 0 and 1 representing the similarity score of the two strokes:

$$\text{Score}(S_i, S_j) = \frac{1}{n}\sum_{k=1}^{n} t_{k,i} \cdot t_{k,j} \tag{1}$$

The similarity score computed as described above is used with the DBSCAN algorithm to cluster Shape strokes into different groups, see Figure 3 (b). Once the Shape strokes are clustered, the points forming each group representing a single edge are approximated with a cubic B-spline curve using the least squares method (Piegl 1996), preparing them for the formation of a curve network.

Clustering Scribble Strokes

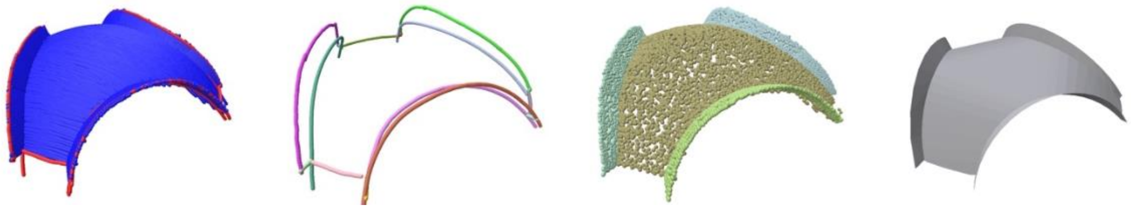
Similar to the way Shape strokes are clustered, Scribble strokes are also clustered into different groups based on the extent of their overlapping area, and the type and position of their corresponding canvases. Firstly, for each two scribble strokes S_i and S_j, all pairs of their polyline vertices whose distance is at most $\frac{w_i+w_j}{2}$ are computed. Then, assuming that $N_{i,j}$ shows the cardinality of all such pairs, their initial corresponding similarity score is computed as follows.

$$\text{Score}(S_i, S_j) = \min\left(C \times 12.5 \times \frac{N_{i,j}}{N_i + N_j}, 1\right) \tag{2}$$

Where C denotes a coefficient that is computed based on the canvas types and positions of the two strokes, N_i and N_j are the number of polyline points on strokes S_i and S_j, and the coefficient 12.5 is empirically chosen. Akin to the Shape clustering, Scribbles are also clustered using DBSCAN algorithm into groups where each group ideally denotes a face of the designer-intended geometry, See Figure 3 (c).

Topology Recovering and Finding Cycles

As previously described, the Shape stroke clusters are approximated using cubic B-spline curves, representing edges, and Scribble strokes are clustered, representing each face of the intended geometry. The focus subsequently shifts to recovering designer-intended connectivity to form a well-connected curve network and surfacing its cycles that are bounding patches as delineated by clustered Shape and Scribble groups, respectively. To this end, for each curve representing a Shape cluster, the proximities with respect to (mid) endpoints of other curves are examined. If the distance between a midpoint on the curve and another curve(s) is below a certain threshold, the curve is split at that point. Subsequently, given several endpoints with their corresponding tangent directions, the connecting points between the obtained set of curves can be determined by solving a minimisation problem (Cao et al. 2016). Next, a graph representation of the sketch is defined, by means of the connectivity information of the curves in the previous process. The challenge lies now in identifying the cycles that should bound surface patches and the ones that should not. To solve this, for each Scribble cluster it is then simply checked whether the curves that fall within its bounding box to determine form a cycle or not in the graph representation of the sketch. If a cycle is formed by these bounding curves, it is subsequently triangulated, resulting in the final reconstructed geometry (Zou et al. 2013) (see Figure 3 (d).



(a) (b) (c) (d)

Figure 3. The stepwise output of the geometry reconstruction pipeline on Buga Wood Pavilion's sketch drawn using MR.Sketch. (a) Shape vs. Scribble strokes classification results, where strokes coloured with red (blue) are predicted by the classifier as Shape (Scribble). (b) Clustering Shape strokes into groups each representing an edge or a boundary curve with each group coloured differently. (c) Clustering Scribble stroke into groups each representing a single face of the geometry with each group coloured differently, and (d) depicts the final reconstructed geometry, derived from the obtained curve network.

3.3 IFC Model Generation

As the geometry serves as the core building block for defining an IFC model, by relying on the reconstructed geometry, as well as the material and structural properties attributed to the sketched strokes, the focus now shifts to automatically defining a model and generating an IFC file as a post-process coming after the geometry reconstruction pipeline.

With this goal in mind, firstly, an IFC hierarchy is automatically created, consisting of an IfcSite and an associated IfcBuilding, which serves as a container for the geometries of the spatial structures it comprises. Secondly, given that each Scribble cluster comprises set of strokes where each of which are coupled material and structural information, these properties are inherited by the triangulated surface patch and boundary curves forming it, leading to a single IfcMaterial and IfcBuildingElement for each patch. Currently, occurrences of IfcBuildElements are limited to six fundamental types, namely, IfcRamp, IfcRoof, IfcSlab, IfcWall, IfcBeam, and IfcColumn. Also, in cases where the architect or designer has not assigned any structural information to the strokes, patches without such semantics are defined with IfcBuildingElementProxy. Similarly, an IfcMaterial entity is correspondingly created.

As IfcProduct allows multiple geometric shape representations of the same product, two IfcShapeRepresentation objects are used to represent the triangulated surface patch and the boundary curves forming it, namely GeometricCurveSet and Tessellation types representing the boundary curves and the patch, respectively. Additionally, IfcCurveStyle and IfcSurfaceStyle are then assigned with the original colour designer used throughout the drawing, ensuring that IFC model visualisation closely resembles the original sketch.

Finally, an IFC file is generated and written, capturing the details of the model and serving as the starting point for further development and refinement in BIM packages. Once completed, the model can be viewed and inspected using any software that supports the IFC format, allowing for seamless compatibility and collaboration across platforms.

4. Proof of Concept

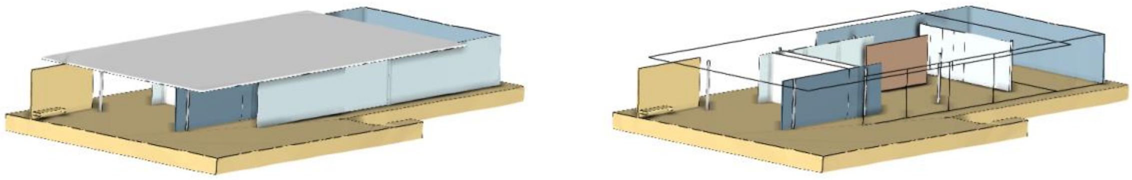
As a proof of concept, the whole pipeline for reconstructing the geometry and IFC model generation is tested on the sketch of the Barcelona Pavilion drawn using the MR.Sketch application. The artist incorporated a total of five distinct building elements and three different materials as semantic information in the sketched strokes. These included columns, a ramp, a roof, slabs, and walls as building elements, as well as concrete, glass, and steel as materials.

Designer has chosen black colour mainly for drawing the outlining boundaries and edges while varying colours are used for enclosing areas to mimic the visual appearance of the original built structure. However, the geometry reconstruction pipeline is by design not dependant on the stroke ink colour and uses other properties for inference as described in Section 3.2. Additionally, the designer has chosen different stroke ink width values to depict the thickness of the structure's constituent building elements. The pipeline uses the average of stroke ink widths forming each Scribble group defining a surface patch as offset distance for extrusion. As for scale measurements, the designer used the MR.Sketch's 1-meter 3D grid as a guide while sketching, so that the sketch and the reconstructed geometry follow the 1:1 scale with the original built structure.

(a) (b)

Figure 4. The sketch of the Barcelona Pavilion drawn using MR.Sketch. (a) Depicts the orthographic view, while (b) shows the perspective view from the back side.

The sketch is then fed into the geometry reconstruction pipeline and undergoes its subsequent post-processing to generate an IFC model. Compared to the original sketch, in its current state, all areas except for two parts - the opening on the roof and one of the columns - were successfully reconstructed, with all patches correctly inheriting the material and structural properties from their corresponding strokes. Additionally, for the generation of an IFC file, all inferred triangulated surface patches are extruded along their normal vectors by the amount of the average scribble stroke ink widths used to mark the patch's enclosing area during sketching. Figure 5 and Figure 6 display the generated IFC model by the pipeline as viewed by the OpenIFCViewer and Solibri, respectively.



(a) (b)

Figure 5. The generated IFC model from Barcelona Pavilion as viewed in the OpenIFCViewer. (a) shows the entire structure, while in (b) the roof and four walls are hidden to provide a glimpse of the interior.

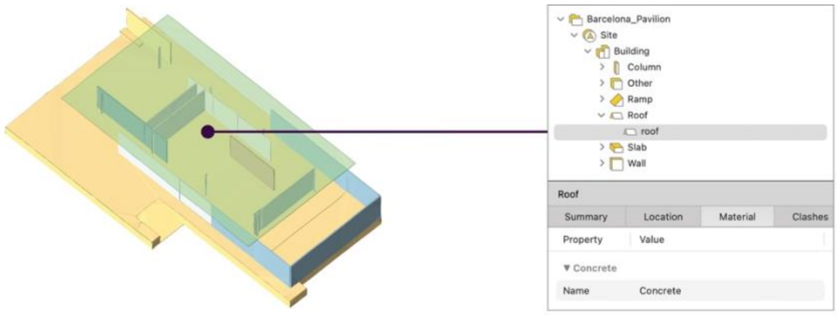


Figure 6. Generated IFC model tree as viewed in the Solibri.

4.1 Computational Cost

The Strokes2Surface takes 206, 34, 280, and 145 seconds on the classification, Shape clustering, topology recovering, and Scribble clustering processes on the Barcelona Pavilion (402 strokes, 278K vertices), respectively. Also, the post-processing process takes 3 seconds to define and generate the IFC model. Notably, the Barcelona Pavilion' sketch is the biggest test case we have fed to the pipeline. All the computations were carried out on an Apple M1 chip featuring an 8-core CPU with 16GB memory.

5. Conclusions

This paper presents extends the Strokes2Surface pipeline built on top of the 4D sketching application, MR.Sketch, which bridges concept design with digital modelling through a machine-learning-based pipeline responsible for reconstructing geometry for further generating an IFC file as a subsequent postprocessing. This enables BIM interoperability with the early stages of design, comprising the following two main processes:

- The geometry reconstruction pipeline recovers the curve network as a rich geometric structure and creates the triangulated bounding surface patches by solving a binary classification problem for detecting the stroke type, and two clustering models in parallel to further cluster each stroke types into groups representing the edges and boundaries and their corresponding enclosed areas.
- As the post-processing step, the reconstructed geometry automatically inherits the strokes' associated semantics such as material and structural info, enabling the generation of an IFC model that includes building elements along with their corresponding materials.

6. Outlook

While the proposed pipeline, along with its subsequent post-processing, marks progress toward BIM interoperability in the early stages of design, certain aspects require further improvements. Firstly, the geometry reconstruction pipeline should be extended to support more intricate and modifiable feature elements, such as IfcOpeningElement. Secondly, additional developments to enable the generation of more generic geometry representation types, beyond Tessellation and including Brep, would greatly enhance the applicability and versatility of the pipeline. Additionally, in its current state, the IFC model generation relies on manually coupled material and structural information with the designer's sketched strokes, which are automatically inherited by the patches. This makes the workflow not fully automated in its current state, suggesting the use of datasets such as IFCNet (Emunds et al. 2021) to automatically classify entities.

Acknowledgements

The research is supported by grant F77 (SFB “Advanced Computational Design”, subproject 2, Integrating AEC Domain Knowledge - Synthesis 2.0) of the Austrian Science Fund (FWF).

References

Andrew Kroll. "AD Classics: Barcelona Pavilion / Mies van der Rohe" 08 Feb 2011. ArchDaily. Accessed 12 Apr 2023. <https://www.archdaily.com/109135/ad-classics-barcelona-pavilion-mies-van-der-rohe> ISSN 0719-8884

Arora, R., Habib Kazi, R., Grossman, T., Fitzmaurice, G. and Singh, K., 2018, April. Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (pp. 1-15).

Breiman, L., 2001. Random forests. Machine learning, 45, pp.5-32.

"BUGA Wood Pavilion / ICD/ITKE University of Stuttgart " 12 May 2019. ArchDaily. Accessed 14 Apr 2023. <https://www.archdaily.com/916758/buga-wood-pavilion-icd-itke-university-of-stuttgart> ISSN 0719-8884

Emunds, C., Pauen, N., Richter, V., Frisch, J. and van Treeck, C., 2021, August. IFCNet: A benchmark dataset for IFC entity classification. In Proceedings of the EG-ICE 2021 Workshop on Intelligent Computing in Engineering (pp. 166-175).

Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996, August. A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

Cao, Y., Nan, L. and Wonka, P., 2016. Curve networks for surface reconstruction. arXiv preprint arXiv:1603.08753.

Chen, X., Kang, S.B., Xu, Y.Q., Dorsey, J. and Shum, H.Y., 2008. Sketching reality: Realistic interpretation of architectural designs. ACM Transactions on Graphics (TOG), 27(2), pp.1-15.

Deng, Z., Liu, Y., Pan, H., Jabi, W., Zhang, J. and Deng, B., 2022. Sketch2PQ: freeform planar quadrilateral mesh design via a single sketch. IEEE Transactions on Visualization and Computer Graphics.

Dorsey, J., Xu, S., Smedresman, G., Rushmeier, H. and McMillan, L., 2007, October. The mental canvas: A tool for conceptual architectural design and analysis. In 15th Pacific Conference on Computer Graphics and Applications (PG'07)(pp. 201-210). IEEE.

Kovács, B.I., Sharifmoghaddam, K., Jauk, J. and Erb, I., 2022. Integrative Mixed Reality Sketching. In Creative Construction e-Conference 2022 (pp. 233-240). Budapest University of Technology and Economics.

Langenhan, C., Weber, M., Liwicki, M., Petzold, F. and Dengel, A., 2013. Graph-based retrieval of building information models for supporting the early design stages. Advanced Engineering Informatics, 27(4), pp.413-426.

Liu, Z., Zhang, F. and Cheng, Z., 2021, October. BuildingSketch: Freehand Mid-Air Sketching for Building Modeling. In 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (pp. 329-338). IEEE.

Rasoulzadeh, S., Wimmer, M. and Kovacic I., 2023. Strokes2Surface: Recovering Curve Networks From 4D Architectural Design Sketches. Technical Report TR-056-02-2023-01.

Schubert, E., Sander, J., Ester, M., Kriegel, H.P. and Xu, X., 2017. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Transactions on Database Systems (TODS), 42(3), pp.1-21.

Sharp, N., 2019. Polyscope. URL www. polyscope. run.

Tono, A. and Fischer, M., 2022. Vitruvio: 3D Building Meshes via Single Perspective Sketches. arXiv preprint arXiv:2210.13634.

Zheng, Y., Liu, H., Dorsey, J. and Mitra, N.J., 2016, May. SmartCanvas: Context-inferred Interpretation of Sketches for Preparatory Design Studies. In Computer Graphics Forum(Vol. 35, No. 2, pp. 37-48).

Zou, M., Ju, T. and Carr, N., 2013, August. An algorithm for triangulating multiple 3D polygons. In Computer graphics forum (Vol. 32, No. 5, pp. 157-166). Oxford, UK: Blackwell Publishing Ltd.