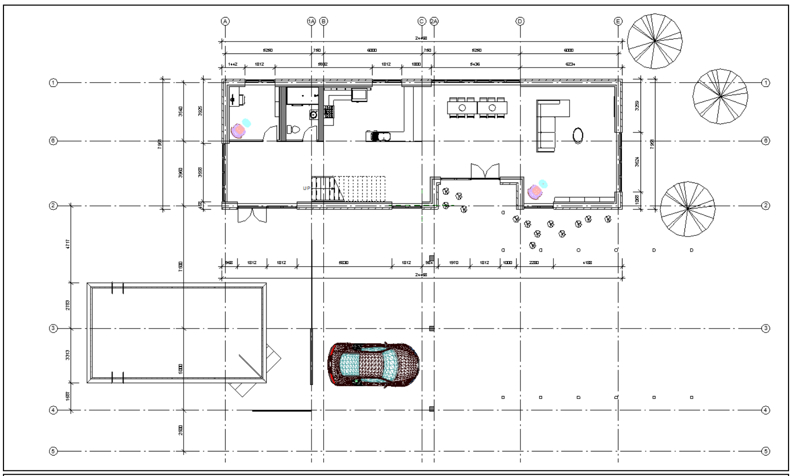
IT UNIVERSITY OF COPENHAGEN

3D reconstruction of buildings based on architectural floor plans, using Open Computer Vision and Optical Character Recognition



Course Title: Research Project
Course Code: KIREPRO1E
Student Name: Alexandru Filip
e-mail: alfi@itu.dk
Study Line: Software Design
Supervisor: Stella Graßhof
Date: 16.05.2022

ABSTRACT

The following Research Paper sets out to investigate possible solutions for an automatized 3D Reconstruction method based on architectural floor plans. Having an emphasis on Computer Science, the paper sets out to explore the larger context of the topic through a structured literature review whilst also presenting terminologies and prevalent technologies within the architecture, engineering and construction (AEC) field, as well as the relationships and dependencies between them.

The second part of the paper focuses on the process of experimentation with different Computer Vision and Optical Character Recognition libraries, in an attempt to design and test an initial reconstruction program prototype from a 2D architectural floor plan onto a 3D environment.

Finally, the paper concludes with the literature review, the results of the implementations, and key takeaways and sets a course for future work.

Index Terms: Computer Vision, 3D Reconstruction, BIM, Building Construction, Architecture

CONTENTS

1 INTRODUCTION

Demands for innovation on the topic of digital dematerialization in managing transformation processes from the building to the territory are putting pressure on the construction industry [54].

The process of creating digital artefacts out of old technical drawings is one of the difficulties that the architectural industry is experiencing in terms of archive updates. Digital recreation to current format standards in which businesses operate is a time-consuming and often manual-done procedure. Concurrently, to conduct performance assessments of existing buildings, and schedule maintenance and repair works, owners and project teams require up-to-date and accurate information [12].

3D reconstruction is an emerging field which aims to create virtual object models or new scenes from image sets [32]. The use of Computer Vision for digital 3D reconstruction of buildings has been an active research topic in recent years [52]. Simultaneously, in the spheres of Architecture Engineering and Construction, Building Information Modelling (BIM) is becoming the global standard in the field, with an integrated approach containing 2D and 3D designs together with spatial linkages, geographical data, building component characteristics, as well as material quantities. It has acquired widespread recognition in the construction sector since it is predicted to save money and increase productivity in construction projects [41].

The ability to harvest data is becoming increasingly correlated to a company's worth. It is evident in the IT business, where companies like Google and Facebook have gained billions by deciphering the data that their users generate. Data is having a comparable impact in the realm of architecture, but unfortunately, not a lot has been written about these changes [8].

Building energy performance (BEP) analysis is becoming more significant as decision-makers have a better understanding of their building's performance and can make timely decisions to improve energy efficiency and reduce environmental consequences [58].

Municipalities, architecture companies and governments could benefit from a better overview of their built environment. By harnessing Data Science and Business Analytical processes to assess embedded carbon footprint in dwellings, understand overall energy needs, or identify new opportunities within the market, all charged by quantified comprehensive arguments based on regional, national or supranational data sets.

Since AEC-related Computer-Aided Design (CAD) software products started being popularized in the 1970's [24], one can still assume with fair certainty that a fair majority of architectural and building plans around the world are still on paper.

By assembling these perspectives, it is not hard to imagine a software product or service that could automatically create a 3D BIM model based on a set of architectural drawings, old 2D CAD formats, images as well as other relevant documentation, which would not only aid in maintaining and migrating relevant information from a 'legacy format' onto a digital format, but it will also allow data analytic processes to be implemented on top of it.

Strangely enough, even though there is a need for such a solution, there isn't an industry-standard product or service on the market that successfully and fully addresses this demand. Attempts have been of course made, and this paper intends to present a general outlook on the topic.

The rest of the paper is structured as follows: First, the findings and conclusions of a structured literature review on the topic of 3D reconstruction attempts within the AEC industry will be presented. Secondly, in the methods chapter, the libraries and tools used for developing reconstruction software will be presented. Thirdly, in the experiments chapter, the results of the implementation attempts of the libraries and tools are presented, together with a comprehensive description and testing of the author's reconstruction software. Finally, a conclusion will be presented, which touches on threats to validity, limitations and a short perspective on future work.

2 RELATED WORK AND LITERATURE REVIEW

This section will report and discuss the findings of a structured literature review, with the intent of discovering the technologies and practices used for digital reconstructions in the AEC field, with the goal of understanding and establishing the best point to start this reconstruction attempt. The *Related Work* section below is mainly meant to introduce the reader to a set of recurring concepts, processes and software products, commonly found within the field of AEC.

2.1 RELATED WORK

Building Information Modelling is a framework of policies, processes and technologies that work together to create a "methodology for managing important building design and project data in digital format throughout the life-cycle of the building" [42]. It can be described as a technological and procedural transformation [11] compared to previous design and construction management processes. It is also a prevalent format standard, which encompasses specific data files, usually containing a 3D design, but can also contain structural tests, material quantification and even construction/renovation processes. It can therefore be used as a "living artefact" of the building throughout its lifespan.

Scan-to-BIM is the process of transferring scanned data into BIM models. It has a popular coupling with laser scanning technologies, which offers the data for it. Feeding laser scan data into BIM design tools is considered a standard industry practice, but mixed processes and technologies are also popular, such as including image data as a source. However, for large-scale projects with multiple construction parts and complicated geometries, the manual approach is time-consuming and error-prone. To replace the manual approach, researchers have tried to develop semi-automated scan-to-BIM approaches. [59]

Photogrammetry is the science and technology of obtaining, recording, measuring, and analyzing photographic images to get reliable information about physical objects and the properties of surfaces and objects without physical contact with the objects. [46]. It is an engineering subject, which is highly affected by advances in computer science and electronics. The discipline, like many others, is constantly changing. This is especially true as approaches transition from analogue to analytical and digital.

Blender is a free and open-source 3D computer graphics software toolkit that may be used to make animated films, visual effects, art, 3D-printed models, motion graphics, interactive 3D apps, virtual reality, and, formerly, video games. 3D modelling, UV mapping, and digital drawing are just a few of Blender's features [33].

Speckle is a free and open-source digital infrastructure for 3D design. It manages software interoperability, real-time collaboration, data management, versioning, and automation between different CAD or BIM file formats, mainly found in AEC and Design industries [23]. One popular key feature of this is the Speckle Connector, which can extract and exchange data between the most popular AEC applications. It is comprised of a broad set of connectors that allow you to exchange geometry and data directly within the AEC's most popular design and analysis applications. These add-ins not only convert data from one file format to another, but also converts project data into an open, transparent, and accessible format [22].

2.2 STRUCTURED LITERATURE REVIEW

Given one of the research paper's objectives, namely to discover and report on current possible solutions for an automatized 3D Reconstruction method based on architectural floor plans a *Structured Literature Review* [25] was conducted, to discover the state of current research on the topic of digital 3D reconstruction for AEC, its technologies, practices, processes and tools.
For this purpose, the following research questions were asked:
RQ1: What are the existing solutions, practices and processes for digital 3D reconstruction in AEC?
RQ2: What technologies and tools are being utilised in the pursuit of addressing **RQ1**?
To complete this task, a structured process was used, involving three phases: Planning, Conducting and Reporting [25].

2.2.1 Data Collection and Study Selection

Initially, a search strategy was developed, which included a a list of online digital libraries specific for Computer Science such as: *IEEE Xplore*, *ACM*, *Science Direct* and *Springer Link*, but also more broad digital tools such as *Google Scholar*, *Research Gate* and *Det Kongelige Bibliotek*.

Next, a set of specific search terms were identified such as: *3D Re-modelling*, *Visual Reconstruction*, *Building Construction*, *Architecture*, *BIM*.

The search strings were grouped based on their semantic relevance and in connection to the research questions. *AND* and *OR* operators were also included during the process to widen the search criteria [25].

Based on the initial hits and upon a primary review, forward and backward snowballing techniques were used to identify new and relevant material. It is important to mention that some papers and publications had to be dropped from the primary review since there was no way to access the full version of said papers.

After the removal of duplicates, the selection of primary studies [25] was put together by reviewing the titles and abstracts of all the papers. Out of 39 papers reviewed, 27 research papers were further selected for analysis. The selection criteria for inclusion were: The paper's main concern has to be within digital 3D reconstruction in AEC, it has to describe or reflect upon a method, process or technology, and it has to be exemplified. The conducted literature review covers a multi-disciplinary range including varying papers deemed relevant from the AEC industry, Computer Science as well as software documentation of Computer Vision Libraries.

2.2.2 Data Analysis

In the pursuit of structuring the review after the Study Selection phase, a mapping study was made as seen in **Figure 1**, in an attempt to categorize the results of each research question. At the two extremities, the mapping had the papers in which, on one end, the implementation was highly dependent on a specific technology, whilst at the other end, the papers that were focusing on a process were grouped. The papers that were covering the topic as a whole were grouped and placed in a small bubble in the middle of the figure (green labels). A colour coding was applied as well, having the technology-oriented papers represented with blue, the process-focused ones in red, with the rest being represented with different shades of the combination of the two primary colours. It is to be noted that this should not be considered a definitive way of categorizing these studies. It is also worth mentioning that most of these studies describe mixed processes and technologies and therefore, **Figure 1** could also be interpreted as an initial plotting of the primary studies.

Figure 1: Mapping of primary studies

2.2.3 Findings

When it comes to the state of the technologies and processes utilised and presented in the selected studies, out of 27 papers selected 11 have the word *"Automated"* in their titles. It is imperative to mention, that when it comes to the actual solutions presented, none of them can be considered automatic. Something automatic implies that it works by itself with little or no direct human control. This is not the case in any of the papers surveyed. Some authors such as **Lim et al. (2018)** [27] make this remark, by implying it towards the end of their paper, even though their paper titles explicitly express an automated process. Perhaps this is a good indication of the state of this pursuit within research. Radical steps still have to be made, to put forward a practical and commercially-viable product.

Haala et Kada (2010) [16] address this topic in their paper titled ***"An update on automatic 3D building reconstruction"***. They conclude by affirming that despite tremendous effort, the challenge of automatic interpretation has kept operational 3D modelling to systems requiring a lot of manual work and only a few moderately automated tools. As a result, a significant number of researchers are still working on developing entirely autonomous systems.

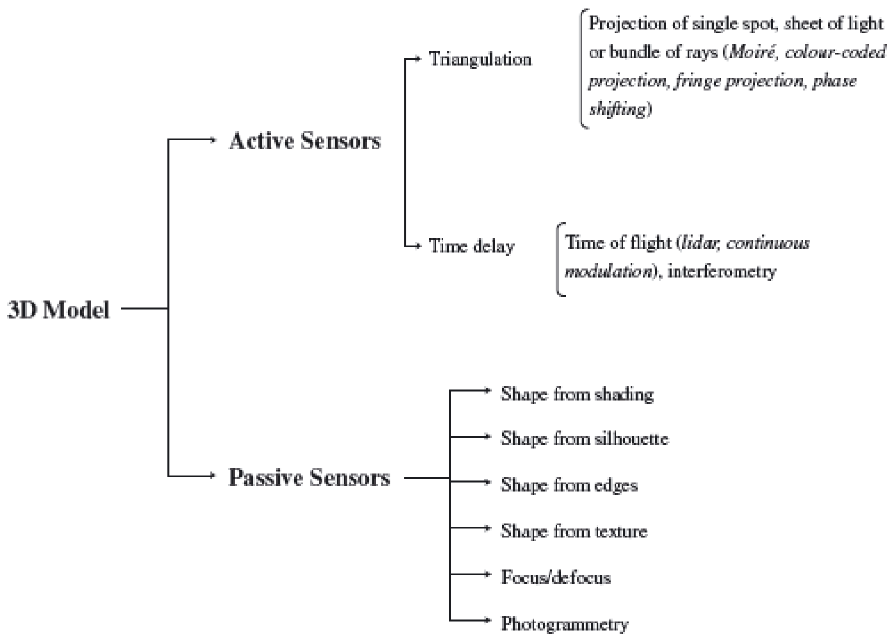


Figure 2: Acquisition Systems Tree by **Remondino et al. (2006)** [43]

Remondino et al. (2006) [43] describe a comprehensive and complete approach to the acquisition, processing and visualisation of 3D information from images that have been examined. The significant contribution to the scope of this review represents the categorisation and overview of possible technologies used in data collection but grouped based on active or passive types of sensors. This is represented in Figure 2.

The study conducted by **Murthy et al. (2012)** [32] sets out to assess the value of 3D reconstruction within the AEC industry. What is worth highlighting from this article is the introduction of a comprehensive list of artefacts that are used in construction projects that might also be used in a reconstruction process. These are synthesized and revised:

- Paper and electronic-based drawings such as Floor plans, Details, Isometric Views, Elevations and Sections;
- 3D models such as Revit, ArchiCAD and Rhino models;
- Photo and/or video materials;
- Textual project documentation;
- Laser Scanning (not included in this paper);

The paper goes on and also presents early findings from a series of user studies that evaluate the utility of 3D reconstruction in the construction industry. These findings mainly point towards issues of

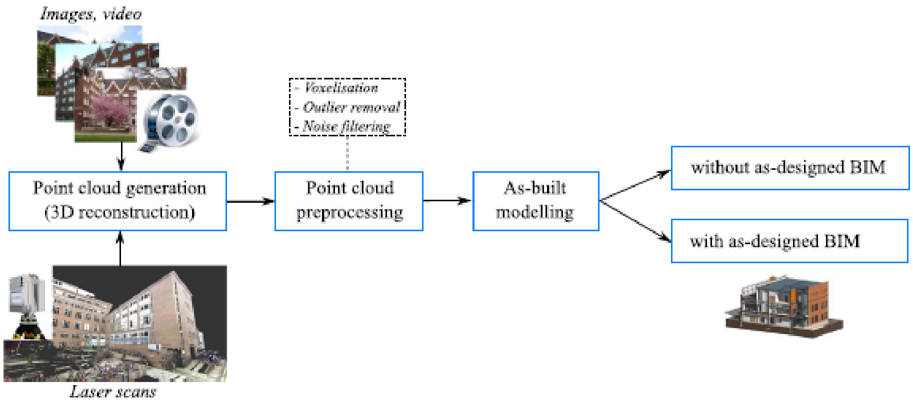


Figure 3: As-built modelling process by **Pătrăuceanu et al. (2015)**[41]

miscommunication of visual information due to incomplete or incorrect information, as well as the difficulties in orientation in current visual information due to a lack of a comprehensive and more robust body of data.

Pătrăuceanu et al.(2015) [41] give a comprehensive high-level overview of the as-built BIM (AB-BIM) modelling process, with a concentration on the data modelling side (as opposed to data collection), and discusses the potential of relevant works from various research communities to be used to automatically build AB-BIMs. It is noted that the 3D reconstruction challenge has been addressed by different research communities such as Computer Vision, Geometry Processing and Civil Engineering. It is reported that Computer vision from images and/or 3D laser scanning technologies were utilised which provided off-the-shelf tools for creating 3D representations of models. These models are made up of a collection of 3D points containing 3D Cartesian coordinates and sometimes colour information. They are valuable for visualizations and augmented reality, but it is still far from a robust solution, due to the need for an extensive human intervention (semi-automatic or manual processes), making them error-prone and expensive. What stands out in this paper is the prototypical process overview of a typical 3D reconstruction process.

Figure 3 summarizes the typical as-built modelling process, which includes point cloud production from either a form of laser scanning or from a collection of media. Next, it goes on to a point cloud preprocessing and finally, towards the as-built modelling itself. Similar processes or variations of this process will be recurring throughout this review since it represents a comprehensive process in its most simple form. This model does not include Machine Learning, but it will be covered in this literature review.

The paper concludes by affirming that in practice, obtaining a complete BIM model today, can result in high costs. It goes on by presenting a positive view on the belief that automatically generated

BIM models are possible, though it will require more interdisciplinary projects with a significant emphasis on the consolidation and integration of existing techniques.

Tang et al. (2010) [55] survey techniques developed in civil engineering and computer science that can be utilized to automate the process of creating as-built BIMs. What is interesting to highlight from this article is the discussion part on the technology gap, where they argue that even though there is a definite need for semi-automated or even automated ways toward 3D BIM reconstruction, this research area is still in its early stages. Overall the methods that address geometric modelling, object recognition, and relationship modelling can be important building blocks that as-built BIM automation can build upon. Lastly, BIM representation and data transmission protocols now in use, presume idealized geometries that are rarely seen in real-world facilities. Walls are not perfectly flat, and corners are rarely exactly 90 degrees. Thus, new interpretations or modifications to existing representations are required to close these gaps.

2.2.3.1 *Photogrammetry*

Photogrammetry can come in a variety of forms. Extraction of three-dimensional measurements from two-dimensional data is one example; for example, if the scale of the image is known, the distance between two points on a plane parallel to the photographic image plane may be derived by measuring their distance on the image. Another is the extraction of precise colour ranges and values from material images for physically based rendering. Traditional aerial (or orbital) photogrammetry collects photographs from a greater distance than close-range photogrammetry. Photogrammetric analysis can be performed on a single snapshot or it can be used to identify, measure, and classify objects using high-speed photography and remote sensing [40].

Baillard et al. (2011) [3] present reconstructions of buildings from aerial photos. The first part of the study focuses on an algorithm for matching line segments across many photos automatically. The second advancement presented in this paper is a method for computing a planar reconstruction based on matched lines automatically. The uniqueness here is that by utilizing an inter-image homography applied to the line neighbourhood, a planar facet hypothesis may be constructed from a single 3D line. From numerous photos, the system has successfully built nearly complete roof reconstructions.

Remondino et al. (2006) [43] argue that the key advantages of image-based modelling over laser scanners are that camera-based sensors are generally inexpensive and more portable compared to laser scanners. Furthermore, 3D information may be reliably recovered regardless of the object's size. Although image-based modelling can yield precise and realistic-looking models comparable to those produced by laser scanning in some circumstances, it remains highly interactive because most automated methods have yet to be demonstrated in real-world applications. It is also argued that without making assumptions about surface shape, automated BIM reconstruction cannot capture detail on unmarked or featureless surfaces. As a result, creating geometrically precise, detailed, and full 3D models of complex objects remains a tough task and a popular research area.

The study by **Suveg et al. (2003) [52]** presents a mixed 3D reconstruction method by using aerial images and GIS maps. This approach generates building models by deriving 3D information from stereo images. By combining the images with a GIS map, the specific strengths of both the images (high resolution, accuracy, large information content) and the map (relatively simple interpretation) can be exploited.

The paper stands out with its proposal of having the building detection and reconstruction separated in this method. Given that the reconstruction is focused on a single structure decreases the reconstruction's complexity significantly, as opposed to other techniques like presented in **Baillard et al. (2011) [3]**, which focus on the bulk capture of shapes and buildings from the building detection phase. The ground plan of the structures featured in the map can be used to localize the buildings in the photographs. Buildings have a wide range of shapes, but even complicated structures can be created by combining simple building models. As a result, it is beneficial to divide a complicated building into simple building elements, each of which corresponds to a fundamental building model.

After this, the ground plan information from the map can be used to complete this process. The use of the Constructive Solid Geometry (CSG) representation for describing buildings is suggested by this approach of modelling buildings, using a set of basic building models (primitives). After that, the primitive assumptions are tested by fitting them to the photos. It is argued that merging the building primitives with the best fitting results delivers the 3D reconstructed building. The paper proposes future work toward the automatization and standardization of the method.

Dick et al. (2000) [9] describe a system which can automatically derive 3D models of architectural scenes from multiple ground images. The scope of this paper is targeted toward building facades thus not covering a comprehensive 3d building reconstruction, and therefore it will not be covered in depth. In essence, the study introduces a planar model which consists of detecting and defining each planar model by being specified by spatial parameters. From this, a so-called planar model is generated. After a point matching step is applied from the images, the model is processed by recursively applying a RANSAC

(Indoor plane extraction on point cloud algorithm [60]) plane estimate technique to the 3D reconstruction extracts planes.

Schindler et Bauer (2003) [47] attempt to solve the same issue as **Dick et al. (2000) [9]**. The difference is that **Dick et al. (2000) [9]** devised a method that uses a sampling scheme to detect only vertical planes, which is then upgraded by picking and fitting template primitives with a Bayesian model selection method. **Schindler et Bauer (2003) [47]** introduce a model-based reconstruction method that use close-ranged images. Primitives generic templates ('models') of the geometric features in the data are utilized, and their parameters are determined to ensure that they match the data, a method very similar to what is presented by **Sepasgozar et al. (2016) [48]**. There is a recognition step and an optimization step in this process. In the global coordinate system, the recognition step provides some approximate locations for each template. The parameters of the templates are modified to the image data during the optimization process. The total number of parameters to be estimated is determined by the complexity of the primitive employed and the number of transformations we allow during the fitting process. With the use of a regression algorithm, the 3D points generated by the picture matching are segmented into a coarse polyhedral model, thus determining the 3D shape of the model.

2.2.3.2 Scan-to-BIM

Son et al. (2015) [51] offer a review of the current technologies and methods used in Scan-to-BIM. It is a useful compendium of technologies which compares technologies such as photo/video-grammetry, terrestrial laser scanning as well as "off-the-shelf" 3D modelling technologies such as **Trimble RealWorks**, **Leica Cyclone** or **ClearEdge3D EdgeWise**. The outstanding element of this study, which brings value to this literature review is the fact that it addresses the issues of integrating facility management (FM) systems, which necessitate accurate as-built BIM. Facility management (FM) is a professional management discipline focused on the efficient and effective delivery of logistics and other support and maintenance services related to real property. It encompasses multiple disciplines to ensure the built environment's functionality, comfort, safety, and efficiency by integrating people, place, process, and technology [34]. BIMs, specifically mechanical, electrical and plumbing-focused models (MEP BIMs), are evaluated and updated regularly during project coordination meetings amongst stakeholders. However, some coordination difficulties may be identified in the BIM but resolved in the field; these modifications are rarely updated in the BIM model because presenting an exact as-built BIM model is not required as part of the project deliverables. Instead, 2D as-built drawings are generated from BIM models, manually updated to reflect as-built circumstances, and delivered to the owner. Typically, no FM information systems, such as computerized maintenance, are coupled to these 2D as-built drawings, thus there is a significant loss of interoperability between the BIM model and FM maintenance systems. Therefore the comparisons are also made by testing the technologies and tools by analyzing MEP elements in buildings.

Esfahani et al. (2021) [12] build on top of the work done by **Son et al. (2015) [51]** and they attempt to take this further by trying to assess the accuracy and precision of Scan-to-BIM. They conclude that with the current development of technologies, Scan-to-BIM is becoming more feasible as they test different techniques as well as a commercial software solution that offers a semi-automatic process. It is also noted, that the main challenge in this field is the lack of a set of comprehensive standards and guidelines, but even so, automated modelling systems have the potential to significantly enhance both accuracy and precision when modelling key building objects (along with time savings). Another notable mention useful for the scope of this literature review is that the authors explain the basic process used in Scan-to-BIM, as well as an up-to-date compendium of techniques used in the industry. Scan-to-BIM has become a feasible approach for collecting and modelling 3D as-built information and has three phases: (1) scanning, (2) registration, and (3) modelling, somewhat overlapping with the model proposed by **Pătrăuceanu et al.(2015) [41]**. The paper focuses on the modelling phase, which can currently be conducted either manually or semi-automatically. Scan-to-BIM research advances it divides into two categories:

A) Scan-to-BIM state-of-the-art automated data collecting and modelling methodologies. These are automated data acquisition techniques such as video-grammetry, terrestrial laser scanning, 3D video camera ranging, and Simultaneous Localization and Mapping (SLAM). Also, modelling techniques for Scan-to-BIM: Spatial correlation (meshing or grid generation by creating a 3D surface model representative of the overall topology of the scanned facility), Object feature recognition, Material-based approach (detects objects' texture and structure), size fitting (algorithms that search along the available bank of parametric elements (e.g., walls, windows, doors, pipes, and structural elements and fitting the best element to the recognized object the maximum conformance).

B) Performance evaluations of Scan-to-BIM. This section focuses on identifying the main aspects that affect performance, though there is no standard evaluation protocol presented to handle these issues. The paper argues that during the scanning stage, many factors influence the accuracy of sensed 3D data, but mainly by the calibration of the utilized devices. Researchers have utilized precision as a metric to assess the accuracy of 3D BIMs. It is characterized in those studies as a percentage of positive (correct) prediction values, and recall is defined as the true positive (correct) rate of the modelling outcomes of the needed items.

In contrast, **Badenko et al. (2003) [2]** propose a six-stepped methodology as well as two case studies proving the suitability of the process. The steps of the methodology are:

- Classification of considered elements from exiting assets
- Definition of the required level of detail (GI)
- Scan data acquisition
- Point cloud registration and segmentation
- As-built BIM creation
- Analysis

To a similar extent, **Wang et al. (2019) [59]** present an application-oriented framework for scan-to-BIM that explains four fundamental steps of the scan-to-BIM process and their connections, as seen in **Figure 4**. This is somewhat congruent to the framework presented by **Badenko et al. (2003) [2]**, though this study is more thorough and more exhaustive in presenting and arguing for the steppes used in this framework. Another notable difference here is that the **Badenko et al. (2003) [2]** methodology is a linear one, as opposed to a iterative one as shown in **Figure 4**. Though the iterative nature is not made very explicit, it is made visible in the graph. The framework is focused on a specific BIM application that will be deployed using the as-is BIM. These are the determination of needed scan data quality, the identification of information requirements, the collection of scan data, and the reconstruction of BIM as-is.

The first step consists of identifying the specific requirements of the BIM application, which include the required architectural elements to be modelled, the appropriate Levels of Design (LOD) and the required non-geometric properties. Secondly, the necessary quality requirements of the scan are established. Thirdly, the data acquisition process is done, and finally the reconstruction. The paper concludes by highlighting the importance of specific requirements elicitation for various BIM applications, as well as the importance of identifying the quantitative correlations between modelling accuracy or point cloud quality and the trustworthiness of the data. It is argued that BIM should be explored as-is for its intended usage, and scan planning approaches should be investigated further in circumstances when a BIM as-designed does not exist, as well as for lasers.

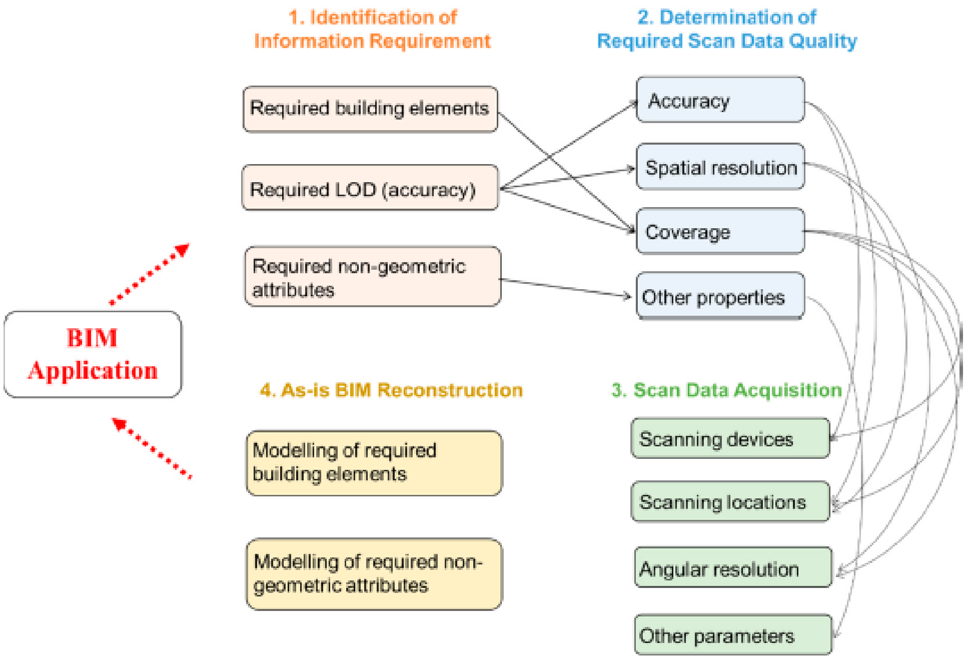
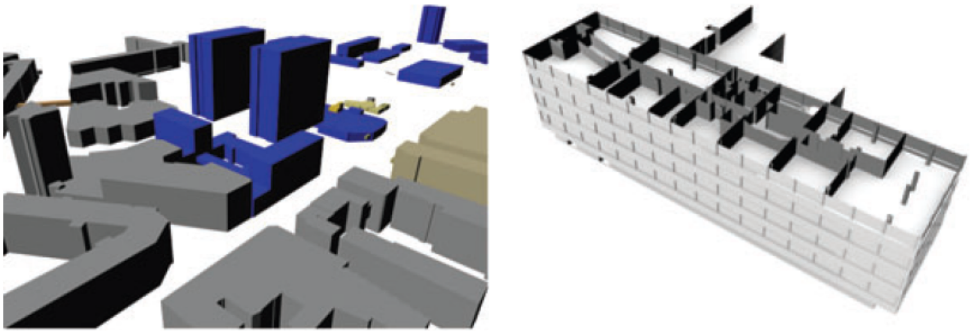


Figure 4: Scan-to-BIM framework by **Wang et al. (2019)** [59]

2.2.3.3 Terrestrial and Airborne Laser Scans

Haala et Kada (2010) [16] review several current approaches to comprehensively elaborate the state of the art of reconstruction methods and their respective principles. The paper offers an overview of some of the mixed methods of photogrammetric 3D measurements using airborne stereo imagery or Light Detection and Ranging (LIDAR), used in urban 3D modelling. These are: Reconstruction based on areal photos and LIDAR, combined with a parameterisation process. **Vosselman et al. (2001)** [57] when attempting to assess the quality of models reconstructed from airborne laser scanning data they remark on the dependency between the input data and processing technique quality of the models, which is determined by processing technique, as well as the user's requirements and expectations. More importantly, they note that some features, such as hip roof ridgelines, are more precise to extract than others, such as flat roof margins. To assess the validity of assumptions, an empirical quality check can be performed. Feature Recognition is based on "the topological relations between segmented regions are commonly described in a region adjacency graph". The paper also touches on the reconstruction of building Facades from airborne imagery, and terrestrial data collection with images, lasers or LIDAR.

Budroni et Boehm (2010) [5] present an approach to using terrestrial laser scanning for building interiors, with a process of reconstruction based on that. The approach seems to have familiarity with the study done by **Son et al. (2015)** [51]. LODs are also discussed



(a) Level of Design 1 (b) Level of Design 4

here, and the part worth mentioning from this study is the presentation of the Open Geospatial Consortium (OGC) being mentioned here, as a source of inspiration for providing LOD standards. The OGC standard proposes 5 levels of detailing. LOD1 (**Figure 5a**) and LOD4 (**Figure 5b**) present the differences in detailing between.

2.2.3.4 Neural Networks

Zeng et al. (2019) [61] present a method for identifying elements in floor plan layouts. Apart from walls and rooms, the goal is to be able to distinguish various floor plan aspects in the floor plans, such as doors, windows, and different sorts of rooms. This is important since BIM models independently recognize these object elements. To achieve this, a deep multi-task neural network is created to fulfil two tasks: one to learn to predict room-boundary elements, and the other to learn to anticipate room types. To achieve this, first, a hierarchy of floor plan elements is established, which helps establish room boundaries. Based on this hierarchy, a deep multi-task network is created, with one job of predicting room-boundary elements and another of predicting room-type elements. The data sets used for this study could not be found online, and therefore I could not be used as a basis to further this study. They go on and compare their networks with an edge detection method and rightfully so, argue that the neural networks identify structural elements with semantics from the floor plans not merely edges. This is indeed a very powerful step forward. The paper underlines this, by saying that recognizing edges in floor plan photos alone is ineffective for floor plan recognition. The paper concludes by remarking on the success of their network over the others in terms of the overall accuracy and metrics.

Joshi et al. (2020) [19] attempt to do something similar as well, though they explore the extraction of information from architectural floor plans by using a mixed approach between Deep Learning Networks and combining them with Computer Vision methods such as *findContour* and *Hough Transform*. It is done by designing a two-branch multi-task deep learning network in which one branch identifies boundary type (wall, door or window) and the other branch identifies room type. It is done by using the **Zeng et al. (2019) [61]** Deep Floor Plan architecture. The same data set was used here as in the **Zeng et al. (2019) [61]** article, thus unfortunately not furthering this literature review with any usable data set. The shortcomings of this approach as reported by the authors are that some of the work such as the identification of pixels close to boundaries is set by observation which can only mean manually, decreasing thus the efficiency radically. Thus the *findContour* functionality does not always work due to inaccuracy and some contours are mere approximations.

2.2.3.5 Computer Vision and OCR

Gimenez et al. (2015) [15] present a study that aims to improve approaches for creating 3D building models from 2D blueprints. The prototype can extract data from 2D plans and construct IFC-compliant 3D models of the building's essential components, such as walls, openings, and spaces.

Even though, as admitted by the authors, their method is error-prone due to, limitations around the reliable shape and space recognition as well as the lack of knowledge of wall heights due to the limitations imposed by 2D plans. What is important to highlight from this study is the process of converting 2D scanned plans into a 3D building model, as it gives a replicable model, which involves 2D plans, as well as textual processing as well, and algorithms based on Hough methods.

2.2.3.6 Processes, Methods and Frameworks

Lu et al. (2005) [28] suggest the *INDAI* approach to an 3D reconstruction process. The fundamental idea is to convert explicit representations such as graphic primitives in-plane or section drawings, annotations, and tables into normalized global graphic primitives of three orthogonal perspectives - top, side, and front views.

The implied data is retrieved from texts, tables, and domain knowledge, and projected onto normalized perspectives. This approach builds on the *SINEHIR* (Self-Incremental Axis-Net-based Hierarchical Recognition) model [29] which is an attempt to design and implement a series of integrated algorithms for recognizing dimensions, coordinate systems and structural components. though it is not explicitly stated, it is presumed that the algorithms rely on Computer Vision functions and methods, based on the explanations given by the authors. What is furthermore interesting in this paper is that it brings back into perspective the idea of retrieving data from textual docu-

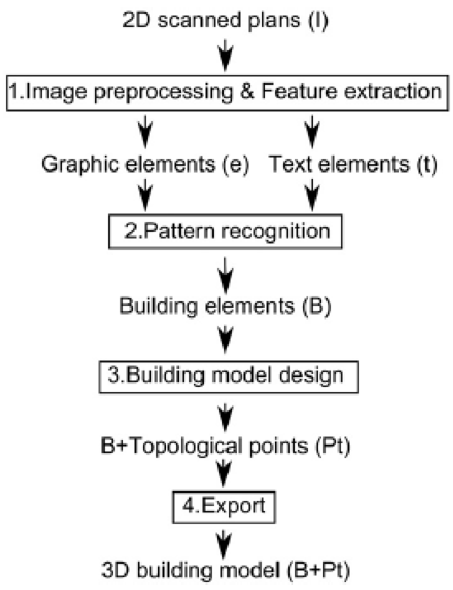
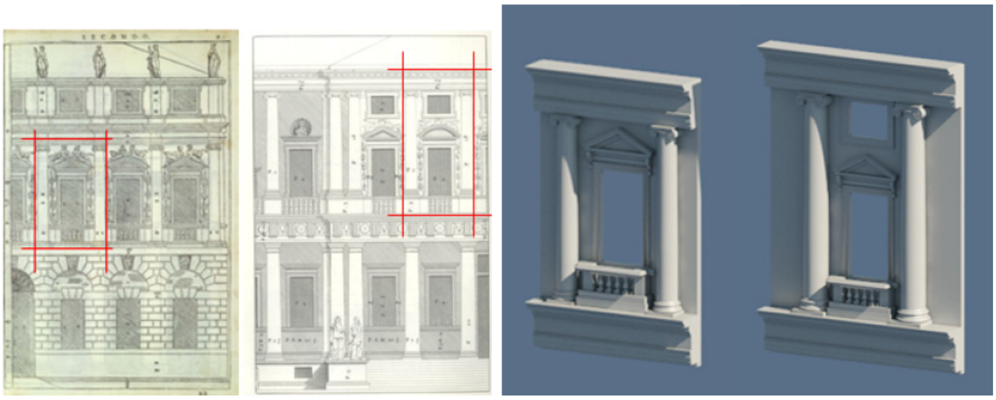


Figure 5: Proposed process by **Gimenez et al. (2015)** [15] [59]

ments, and the idea shared in the **Murthy et al. (2012)** [32] paper as well.

Lim et al. (2018) [27] This study presents a semi-automated BIM generating approach. It attempts to reconstruct its models using 2D CAD drawings. The procedure is divided into two parts: The first phase, named the 2D CAD drawing preparation, deals with cleaning the drawings to acquire simplified 2D input geometry. The second phase being the 3D BIM model production, which entails generating and extracting parameters to build 3D BIM components. Though the technologies and processes are not explicitly presented, based on the figures presented in the paper, is that the first phase of "cleaning up the drawings" is done with the utilization of Computer Vision libraries. Though the paper is not explicitly informative in any way since there is no opportunity for replicating the process, it reinforces the importance of handling doors and windows as objects within a wall in BIM models, a topic highlighted by **Zeng et al. (2019)** [61] as well.

Lytle et al. (2011) [30] make this their central scope of the study, where they propose a framework for object recognition to integrate them within BIM models. The framework proposes integrating feature descriptors to a BIM to aid in the recognition of construction objects. It is based on the Industry Foundation Classes(IFC) which is the most used product data type for BIM deployment. The IFC is an object-oriented product data model that is impartial, standardized, and designed for information flow between AEC stakeholders. The framework is based on *Object Recognition* which is a branch of Computer Vision.



(a) 6. Facade Drawings (b) 6. 3D Isometric Views

Finally, the topics presented by **Apollonio et al. (2013)** [1], **Fassi (2001)** [13] and **Remondino et al. (2009)** [44] do not bring novelty in their presentations of methods and technologies, but rather what is important here to mention, is the specificity within architecture that these papers focus, namely towards Architectural Heritage. **Apollonio et al. (2013)** [1] present mostly a methodology consisting of 2 different techniques, namely a global illumination rendering technique from facade and elevation drawings with an example presented in **Figures 6a and 6b**, while the second revolves around interpretative work, obtained from a simplification operation of a model.

Fassi (2001) [13] explores manual, topographic, photogrammetric and laser scanning on a Italian Villa. To assess a cultural heritage piece, photogrammetry and laser scanning take distinct approaches. He concludes that the distinction between photogrammetry and laser scanning is mostly determined by software development and costs. The photogrammetric method he argues is the most cost-effective option, but the software for close-range applications is still rare and undeveloped. The laser scanner is set to be an excellent option (except for a few exceptional circumstances) If an exact model is required, it is the better option. With a laser scanner survey, a total complex form of the object can be retrieved. The acquisition is very fast, simple, doesn't require particular skin but is expensive and requires instrumentation that is portable but not extremely manageable.

Remondino et al. (2009) [44] provide an integrated approach for the virtual reconstruction of complex architectures based on a mix of multiple 3D measuring techniques of medieval castles. It argues for an efficient workflow for castle 3D modelling, increased automation, smooth integration of models developed independently from various data sets, integration of 3D models and GIS data, and the development of relevant visualization tools. What is notable here, beyond the scope of the research are the discussions around required Levels of Design in the reconstruction process of buildings, something also established as an important decision to be made early in the reconstruction process by both **Wang et al. (2019) [59]** and **Badenko et al. (2003) [2]**.

2.2.4 Discussion

The purpose of this study was to conduct and report on a Structured Literature Review designed to answer the questions of what are the existing solutions and processes to 3D reconstruction in AEC and what technologies are being utilised in the pursuit of addressing these solutions.

The motivation for this is was to of course to understand the state of research and technology, but also that being the basis or a good background if one would want to pursue the idea of developing a product that could solve this issue in the AEC industry. Therefore, here are a few aspects to consider when in pursuit of conceptualizing a 3D reconstruction software product:

2.2.4.1 Input Data and Technologies and Solutions

These two aspects are highly dependent on each other, as the reconstruction process is either dependent on the input data upon which technology relies on. Here we can exemplify the list of artifacts such as architectural floor plans, elevation, sections described by **Murthy et al. (2012) [32]** and used for implementation by **Gimenez et al. (2015) [15]**, or in the neural networks presented by **Zeng et al. (2019) [61]** and **Joshi et al. (2020) [19]** and their data sets.

Conversely, at the other end, the literature review accounts for data that is highly dependent on the technology which acquires it. Here we can account for the photogrammetry as reported on by **Baillard et al. (2011) [3]**, **Suveg et al. (2003) [52]**, **Schindler et Bauer (2003) [47]**, **Dick et al. (2000) [9]** and **Sepasgozar et al. (2016) [48]** which all present some form of terrestrial, aerial or mixed approaches towards 3d reconstruction. This indicates a strong presence within research, though none of them seems to be able to deliver a 3d model that includes interior construction elements that could be delivered by floor plans, or in other words it can only reconstruct the shell of a 3d model and not a BIM-grade model.

Nevertheless, the reason why these studies were kept in this literature review and reported on was due to the fact that they can provide as a useful technique in a recovery instance where facade detailing is important, as presented by **Fassi (2001) [13]**, **Remondino et al. (2009) [44]** and **Apollonio et al. (2013) [1]**.

The same processes of terrestrial, aerial or mixed approaches towards 3d reconstruction can also be seen when utilizing laser scanning, though **Remondino et al. (2009)** [44] argues that image-based modelling has several benefits over laser scanning, including the fact that camera-based sensors are typically less costly and portable than laser scanners, and that 3D information can be consistently retrieved regardless of object size.

Scan-to-BIM demonstrates the clear need for a BIM format of being the resulting data type after a reconstruction process. **Son et al. (2015)** [51], **Esfahani et al. (2021)** [12], **Badenko et al. (2003)** [2] and **Wang et al. (2019)** [59] all report on semi-automaic processes usinf a variety of combinations of technologies such as video-grammetry, terrestrial laser scanning, 3D video camera ranging, all indicating a BIM-level information retrieval from these processes. Furthermore, the study by **Son et al. (2015)** [51] indicates the most progress done in the field of 3d reconstruction by looking at all the methods and technologies presented in this review, as the study reports on commercial-grade software available though dependent on the laser technology. As a final note on the topic of data vs. hardware dependency, it seems that the most practical source of data remains architectural drawings since a BIM-level of information can be extracted from them.

2.2.4.2 *Levels of Design*

The topic of Level of Design required in the reconstruction process has been an significant one in the studies by **Wang et al. (2019)** [59], **Budroni et Boehm (2010)** [5] and **Badenko et al. (2003)** [2]. The papers on hand remark on the lack of structure and standards so they create their own inspired by other fields such as in **Budroni et Boehm (2010)** [5]. They discuss the importance not only of having one, but also it proves that due to the specificity of projects, in some cases more primitive geometries suffice as opposed to the papers by **Fassi (2001)** [13], **Remondino et al. (2009)** [44] and **Apollonio et al. (2013)** [1] in which the study cases focus on medieval castles, Italian Romanesque villas or Renaissance facades. Here detailed and exact replicas are crucial, namely requiring higher levels of design.

2.2.4.3 *Processes*

Although there are differences in the meaning of processes, methods, methodologies and solutions, they were gathered under a single section in this research project, with the interpretation that they describe a set of actions or procedures to fulfil a purpose.

Some of the papers describing processes such as **Lu et al. (2005)** [28], **Lytle et al. (2011)** [30] and **Lim et al. (2018)** [27] were grouped together as mostly processes are described, in contrast to other papers where processes are also present, but the focus is more technology-oriented. This being said most of the papers present a mixed process of different technologies, hardware and software. This aspect reflects the complexity of the issue at hand as well as the of approaching to solving the problem of 3D reconstruction in AEC.

All in all, **Pătrăuceanu et al.(2015) [41]** seems to offer the blueprint of a comprehensive 3D reconstruction process, that synthesizes all the major steps, with almost all of the other papers seemingly fitting into the description, except the ones that involve machine learning.

2.2.4.4 Reflections

Given the complexities of requirements, the lack of standardization and limited solutions to address these complexities, from a commercial perspective. Radical steps have to be made to forward a practical and implementable solution before anyone can attempt to make a commercial product and address the need of the AEC industry. Therefore, it perhaps seems more feasible in the short term to argue for a range of services that employ a set of techniques and processes designed to speed up the process of a reconstruction project, instead of pursuing a goal of an all-encompassing automatic software solution. The diversity in architectural styles and specificities seems too complex issues at this point to be able to be matched by notions of standardization, and normalization to harness the true computing and processing power that Computer Science and computers offer in other fields. In terms of the format in which the 3D model should be, based on the literature review, it is undeniable that it should be in a BIM format as described by **Pătrăuceanu et al.(2015), [41]**, **Son et al. (2015) [51]** and others.

Though the studies presented by **Zeng et al. (2019) [61]**, **Joshi et al. (2020) [19]** and **Gimenez et al. (2015) [15]** show the highest potential and they are based on a architectural floor plan-based 3d reconstruction. The first two studies involve machine learning models, but unfortunately, they cannot be used since their data sets could not be retrieved. It is important to mention here that other data-based websites such as *Kaggle*, *Papers with Code*, *Github* and *Google Data Se Search* were accessed, with the pursuit of finding relevant studies based on architectural data sets, but unfortunately, nothing was found now. Thus, the next best direction that the research could take was to pursue further research and implementation of Computer Vision and Optical Character Recognition. Unfortunately, the **Gimenez et al. (2015) [15]** paper offers no clear implementation or theory, thus having to start from scratch.

This direction can be reinforced by the **Lu et al. (2005) [28]** study which attempts to create and implement a set of integrated algorithms for identifying dimensions, coordinate systems, and structural elements. Furthermore, it is worth mentioning here that more than half of the papers reviewed make a direct or indirect reference to Computer Vision-based implementations or algorithms. Looking from a Level of Design perspective, this approach has the potential of inherently including a satisfactory level of design since it would offer a complete mapping of the building if all floor plans, elevations and sections were to be used as the data source.

3 METHODS

presented in the previous section, the research project will focus from here on understanding, assessing and implementing Computer Vision and Optical Character Recognition in a reconstruction software which from here onward will be called *ReconstructR*. Therefore, this section will give a theoretical overview of the library's functions and used in building ReconstructR.

3.1 OPTICAL CHARACTER RECOGNITION AND (PY) TESSERACT

Optical Character Recognition (OCR) is the process of converting images or documents that contain text into machine-encoded text [56]. OCR technologies have been created in the early Twentieth Century as reading devices for the blind and the visually impaired [45]. In 1965 at the "World Fair" in New York, the first advanced reading machine was introduced, namely the IBM 1287. In the following decades, The software system of OCR was mainly developed and deployed in educational institutes, but the binarization techniques introduced in the early 2000s gave access to developing new applications for the differently-abled people, as well as new applications for retrieving information from medieval documents written in non-standard fonts [31].

Today, OCR can comprise different machine learning approaches including Support Vector Machine (SVM), Random Forests (RF), Decision Tree (DT), Neural Networks, etc. Furthermore, researchers combined these machine learning techniques with image processing techniques to increase the accuracy of the optical character recognition system [31]. Cluster computing and the rapid advancements in GPU technology have directed researchers toward Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, etc. [4].

Tesseract is an open-source OCR engine that was developed at HP between 1984 and 1994, becoming open-source in 2005. Tesseract assumes that its input is a binary image with optional polygonal text regions defined. It follows a pipeline where text is recognized and gathered together into *Blobs* [50]. Text lines are created from blobs, and the lines and areas are examined for fixed pitch or proportional text. Depending on the kind of character spacing, text lines are divided into words in different ways. Character cells slice fixed-pitch text almost instantly. Using definite and fuzzy spaces, the proportional text is split down into words [50].

The recognition process is then split into two stages. The first pass attempts to recognize each word individually. Each suitable term is used as training data for an adaptive classifier. The adaptive classifier is then allowed to detect text farther down the page with greater accuracy. Because the adaptive classifier may have learnt anything beneficial too late to contribute at the top of the page, a second pass through the page is performed, with words that were not identified well enough recognized again. In the last phase, fuzzy spaces are resolved, and various hypotheses for the x-height are checked to identify small-cap text [50].

The Tesseract-OCR Engine from Google is wrapped in Python-Tesseract. It can read any image formats supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others, making it usable as a standalone Tesseract invocation script. Python-Tesseract will also output the recognized text rather than saving it to a file if used as a script [17].

3.2 COMPUTER VISION AND OPENCV

Computer vision, or simply "making a computer see," is the science of programming a computer to analyze and calculate information from pictures and videos. [53]

OpenCV (Open Source Computer Vision) is an open-source image and video analysis library that was developed by Intel Research in 1999 as part of a series of initiatives that included real-time ray tracing and 3D display walls [20]. In August 2012, the support is taken over by OpenCV.org, a non-profit foundation [7]. Today the OpenCV Library contains several hundreds of computer vision algorithms [21].

The OpenCV library is organized around various modules, each of which is dedicated to a certain set of computer vision issues, the main class being *Mat*, which represents a matrix holding pixel values of an image, together with several attributes associated with said image [7]. The data variable is essentially a reference to the picture data in the allocated memory block. Alternatively, we may define an initial size and type for a matrix when creating a Mat object [7]. Since an image at its core is constructed by picture elements known as pixels, the image matrix will contain each pixel element. Pixel values in a grey level picture are normally unsigned 8-bit values, however, in a colour image, three such 8-bit unsigned values are used per pixel to represent three colours, i.e. channels (Red, Green, and Blue) [7].

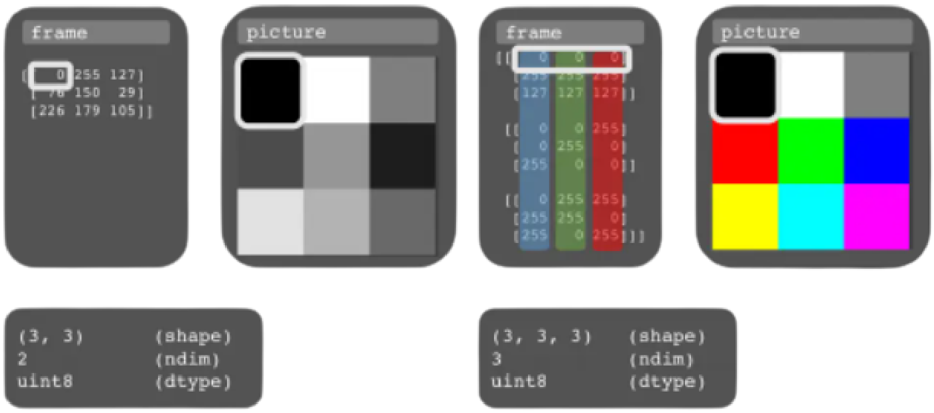


Figure 1: Colour vs. Greyscale pixels

Accessing each pixel is done by specifying each row and column, like with the use of an access method *at(int x, int y)*. Colour pictures have a somewhat more complicated syntax since we are retrieving a vector of three 8-bit values *image.at<Vec3b>(j,i)[channel]*, where the channel is one of the three colour channels [7]. These access methods, together with low-level pointer arithmetic for the basis of pixel access within the *'Mat class'*, open up the possibilities of pixel computation and manipulation.

The *imread()* and *imwrite()* methods in OpenCV handle a variety of file formats for still pictures. Each pixel has a value regardless of format; the distinction is in how the pixel is represented. Thus, an OpenCV image is *a.array* type 2D or 3D array. An 8-bit Grayscale picture is a two-dimensional array of byte values [18].

3.2.1 Grayscaling

The RGB colour model is an additive colour model in which the fundamental colours of light, red, green, and blue, are combined in various ways to give a wide range of colours. The initials of the three additive primary hues, red, green, and blue, are used to name the model. The RGB colour model is mostly used in electronic systems such as TVs and computers for image sensing, representation, and display, while it has also been employed in traditional photography and follows the optical principles of human colour recognition.

Grayscaling is the process of transforming a picture to shades of grey from various colour spaces such as RGB [49]. This is a useful algorithm since it allows other algorithms to work since many algorithms are customized to work only on Grayscale images. A good example is the Canny Edge Detection function pre-implemented in the OpenCV library which only works on Grayscale images.

Grayscale transforms the image from a three-dimensional RGB channel into a single-dimensional one. As shown in **Figure 1**, each pixel in a Grayscale frame has just one number, compared to each pixel in a colour picture which has three numbers.

Transformations within RGB space like adding/removing the alpha channel, reversing the channel order, conversion to/from 16-bit RGB colour (R5:G6:B5 or R5:G5:B5), as well as conversion to Grayscale using the following formulas [36]:

$RGB[A] \quad to \quad Gray: \quad Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$ and respectivelt from Grayscale to Colour, $Gray \quad to \quad RGB[A]: R \leftarrow Y, G \leftarrow Y, B \leftarrow Y, A \leftarrow max(ChannelRange)$. Thus, the same calculations can be done in code as seen below, but it can be done automatically by using the *cv.cvtColor(img, cv.COLOR_ BGR2GRAY)* function.

3.2.2 Dilation and Erosion

Morphological image processing is a set of non-linear processes that deal with the form or morphology of picture features. Morphological procedures are well suited to the processing of binary pictures since they rely solely on the relative ordering of pixel values rather than their numerical values. Greyscale pictures can also be subjected to morphological processes in which the light transfer functions are unknown and the absolute pixel values are of no or small importance.

With a tiny form or template called a structuring element, morphological approaches probe a picture. The structuring element is moved about the picture and compared to the pixels in its immediate vicinity [35].

Dilation, as seen in **Figure 2** is the operation of convolving an image *A* with some kernel *(B)*. During computing, the maximal pixel value overlapped by *(B)* as the kernel *(B)* scans over the picture and replaces the image pixel in the anchor point location with that maximal value. As expected, this maximizing procedure causes bright areas in

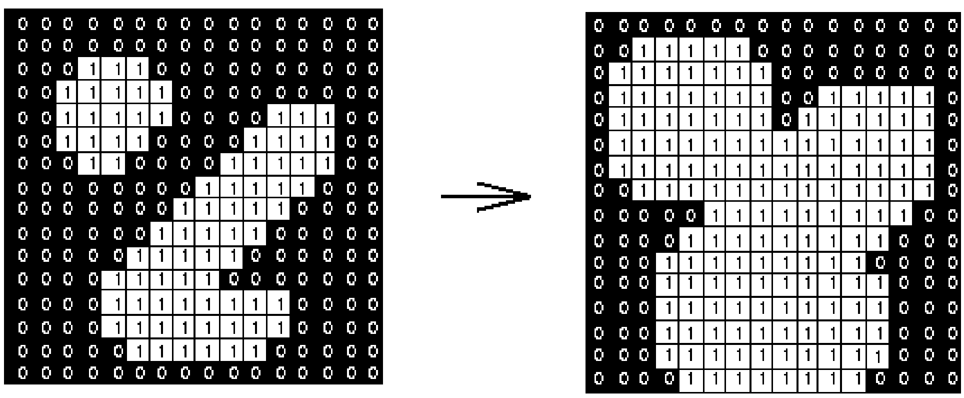


Figure 2: Dilation Representation

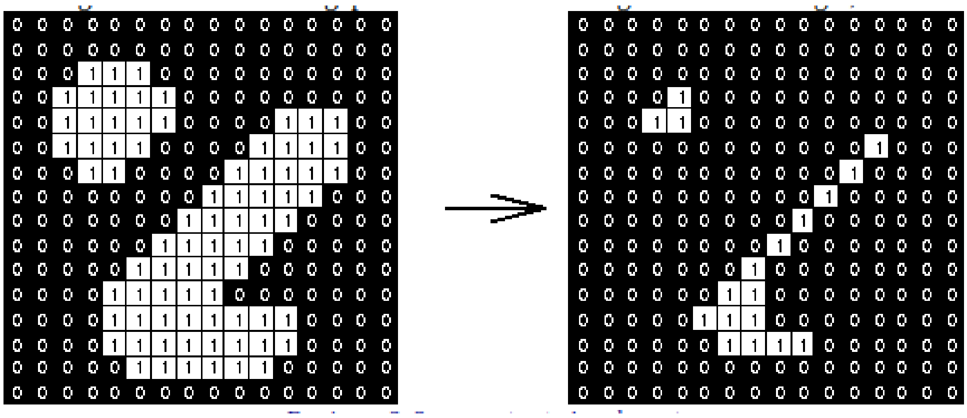


Figure 3: Erosion Representation

an image to "expand" (therefore the name dilation) [37]. The dilatation formula is as follows: $dts(x,y) = max_{(x',y'),element(x',y')\neq 0} src(x+x',y+y')$

Contrary to Dilation, Erosion computes the minimal pixel value overlapped by *(B)* and replaces the image pixel under the anchor point with that minimal value [37], with the formula for Erosion being $dts(x,y) = main_{(x',y'):element(x',y')\neq 0} src(x+x',y+y')$.

3.2.3 Opening/Closing

An opening is simply put as another term for erosion, which is then followed by dilatation. It is beneficial for noise reduction in images.
It is simply, $dst = open(src, element) = dilate(erode(src, element)$.
Closing is the inverse of Opening, with Erosion following Dilation. It may be used to fill minor gaps in front items or to cover little black spots on the object, having the possibility of expressing it as $dst = close(src, element) = erode(dilate(src, element)$.

3.2.4 Canny Edge Detection

The Canny edge detector is a multi-stage edge detection operator that detects edges in pictures. At its core, it uses a multi-stage algorithm to detect a wide range of edges in images. It uses a filter based on the derivative of a Gaussian to compute the intensity of the gradients.

The Gaussian filter in the context of Computer Vision is a smoothing, also known as blurring [39]. In the context of the Canny detector, the use of the Gaussian filter represents a step meant to reduce noise from the image. The most common type of filters are linear, in which an output pixel's value $g(i,j)$ is determined as a weighted sum of the input pixel values, namely $\int(i+k,j+l)$, and the kernel

$h(k,l)$ which are the filter coefficients and expressed by the formula $g(i,j) = \sum_{k,l} \int(i+k, j+l)h(k,l)$.

The Gaussian Filter is done by using a Gaussian kernel to rotate each point in the input array and then adding them together to generate the output array. If a picture is 1D, you can see that the pixel in the centre has the most weight. As the spatial distance between them and the central pixel grows, the weight of its neighbours diminishes. The median filter goes over each signal element (in this example, the picture) and replaces each pixel with the median of its neighbours (located in a square neighbourhood around the evaluated pixel) [39].

The Canny Edge Detection method in the OpenCV library finds edges in an image using the Canny algorithm [6].

There are 4 stages present in the Canny Edge Detection, namely:

1) Noise reduction). This is done by the Gaussian Filter.
2) Finding Intensity Gradient of the Image. The smoothed picture is then filtered in both the horizontal and vertical directions with a Sobel kernel to obtain the first derivative in the horizontal (G_x) and vertical (G_y) directions. From here, the edge gradient and direction of each pixel $canAngle(\theta) = tan^{-1}\left(\frac{G_y}{G_x}\right)$.

The gradient is always perpendicular to the edges. It's rounded to one of four angles: vertical, horizontal, and two diagonal.

3) Non-maximum Suppression). After obtaining the gradient magnitude and direction, the picture is fully scanned to eliminate any undesired pixels that do not form the edge. For this, each pixel is tested to see if it is a local maximum in its vicinity in the gradient direction.

Point *A* is perched on the precipice (vertically). The edge's gradient direction is normal. The gradient directions are *B* and *C*. So point *A* is tested to determine if it forms a local maximum with points *B* and *C*. If this is the case, it will be considered for the following step; otherwise, it will be suppressed (put to zero)[39]. In other words, the result is a binary picture with "thin edges."
4) Hysteresis Thresholding. This stage determines which edges are genuine and which are not. We'll need two threshold values, minVal and maxVal, for this. Any edges with an intensity gradient more than maxVal are certain to be edges, whereas those with an intensity gradient less than minVal are certain to be non-edges, and should be rejected. Based on their connectedness, those who fall between these two criteria are classed as edges or non-edges. They are regarded to be part of edges if they are related to "sure-edge" pixels. Otherwise, they are eliminated as well [39].

The Canny Edge Detection is a part of the Hough Transform, which will be explored below:

3.2.5 Hough Transform

In an image, the Hough transform is a feature extraction method for recognizing basic objects like circles and lines.

A "simple" shape has only a few parameters to represent it. A line, for example, maybe represented by two parameters (slope and intercept), whereas a circle can be represented by three parameters (centre coordinates and radius) (x, y, r). The Hough transform is great at detecting such forms in images.

The Hough transform's key benefit is that it is not affected by occlusion. [26]

A line can be represented by its in Cartesian coordinate system where $Parameters : (m, b)$, or conversely, by its Polar coordinate system $Parameters : (r, \theta)$ [38].

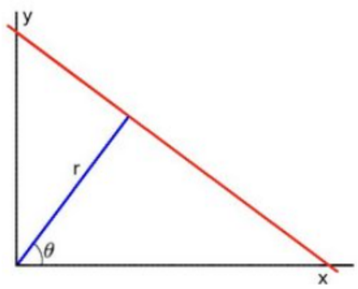


Figure 4: A line in the Polar coordinate system

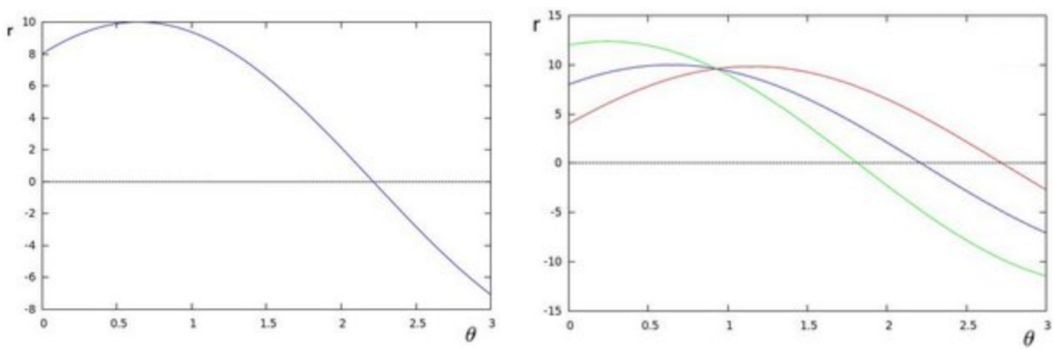
For Hough Transforms, lines will be expressed in the *Polar* system. Thus, **Figure 4** can be represented in an equation as $y = \left(-\frac{\cos\theta}{\sin\theta}\right) x + \left(\frac{r}{\sin\theta}\right)$ or if we rearrange the terms, we get the expression $r = x\cos\theta + y\sin\theta$.

For each point (x_0, y_0), a family of lines can be defined that goes through that point(**Figure 4**) such as:

$r_0 = x_0\dot{c}os\theta + y_0\dot{s}in\theta$

For example, a given (x_0, y_0) if a family of lines goes through it, it generates a sinusoid. So $x_0 = 8$ respectively $, y_0 = 6$ and a plot **Figure 5a** can be represented in a plane $\theta - r$):

The same operation can be done for all of the points in an image. For example, building on the previous one, if two more points are added $x_1 = 4$ respectively $, y_1 = 9$, respectively $x_2 = 12$ respectively $y_2 = 3$ the plot **Figure 5b** can be represented[38].



(a) Plot 1

(b) Plot 2

The three plots intersect in one single point $(0.925, 9.6)$, these coordinates are the parameters (θ, r) or the line in which $(x0, y0)$, $(x1, y1)$ and $(x2, y2)$ lay. This means that a line can be detected by finding the number of intersections between curves. Therefore, the more curves that connect, the more points there are on the line represented by that junction. In general, we may set a minimum number of intersections required to detect a line as a threshold [38].

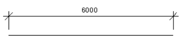
There are two variations of the Hough Line Transform namely, *The Standard Hough Line Transform*, which is a representation of the concepts presented above [38]. *The Probabilistic Hough Line Transform* on the other hand, is considered to be a more efficient implementation, that gives an output of the extremes of the detected lines, as opposed to a vector of couples[38].

4 EXPERIMENTS

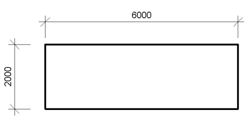
Experimental computer science works best on problems that demand complicated software solutions, such as creating software development environments, organizing non-tabular data, or building tools to address limited optimization problems. The approach focuses on identifying concepts that help solve problems and then evaluating the answers through the creation of prototype systems [10], and this section of the report will exclusively focus on that.

Based on the findings, coupled with the scope and practical goals of the paper, experiments were conducted to find a range or combination of suitable functions to try to build a workflow based on OCR and OpenCV libraries to perform a 3D reconstruction process from architectural floor plans.

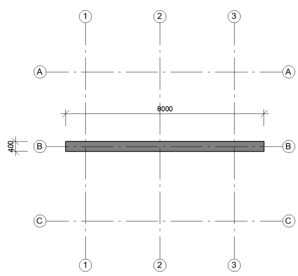
A methodical set of testing materials were developed to aid both in the experiments, as well as in the testing of *ReconstructR* which is the name of the proposed software solution for the scope of this research project. The main idea behind the test set is the increase in complexity when it comes to available data on the floor plans, to also implicitly test the scalability potential of *ReconstructR*.



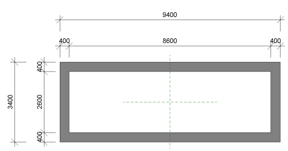
(a) A Simple Line



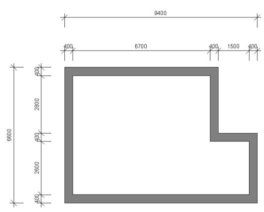
(b) A Line-based Rectangle



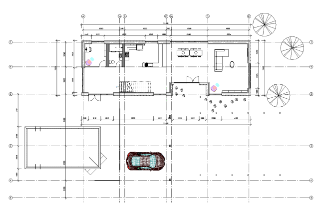
(c) Wall with Constructing lines



(d) A Wall-Based Rectangle



(e) A Complex Wall-based Shape



(f) A Floor Plan

There is a progression in complexity, starting from a simple line (**Figure 1a**) and ending with an actual architectural floor plan (**Figure 1f**).

Furthermore, only 2D floor plans have been selected for testing and the reason behind this choice has to do with a narrowing of the scope of this research project. Taking the *Complex Wall-based Shape* (**Figure 1b**) and then comparing its elevation drawing (**Figure 1a**) with its floor plan (**Figure 1e**), it can be argued that the complexity of the floor plan is higher, and is the floor plan reconstruction is solved, the process of detecting elevations would also be automatically solved, one would just have to scan the elevation.

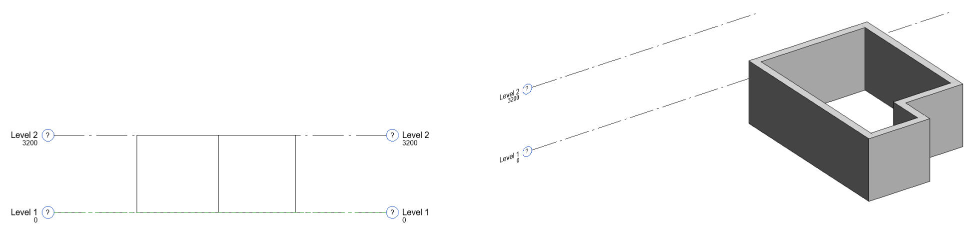
Thus, for the scope of this research project *ReconstructR* will only feature a 2D floor plan reconstruction.

In the next section, the functions of the presented libraries in **Figure 3** will be practically applied with the test floor plans in order then to propose a suitable implementation of *ReconstructR*.

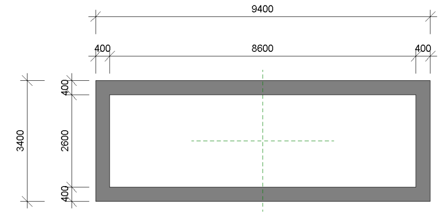
4.1 OPENCV AND OCR IMPLEMENTATION

As mentioned in the previous section, the images mimicking floor plans (**Figure 1a** - **Figure 1f**) were tested against OpenCV and OCR functions in Python, namely with *Grayscale*, *Dilation*, *Erosion*, *Opening*, *Closing*, *Canny*, *Hough Transform* and *Pytesseract*. The Python code for these implementations can be found under the ***Experimentation*** folder. The testing suite followed the progression in complexity given by the test images, to observe the quality and effects of the functions, and to try to find out a suitable workflow for ReconstructR.

Not all of the images will be presented here since it would generate a lot of redundancy, but a description of the functions and intentions will be described as well as the point in complexity where either the goal was achieved, or the function did not work anymore. Lastly, the functions that stand out are *Grayscale Dilation* and *Eorsion*, which were successful with all the test data, but it was not included since they serve as a part of other functions.

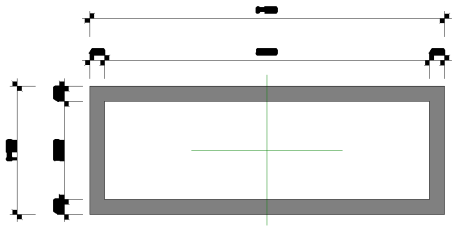


(a) Elevation   **(b)** 3D Isometric View

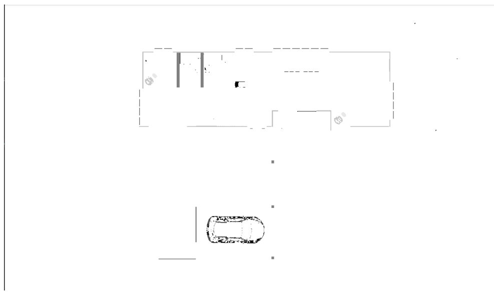


(a) Original image of a Wall

(b) Erosion of the Walls



(c) Open/Closing of a Walls



(d) An Eroded version of the floor plan

4.1.1 Opening and Closing

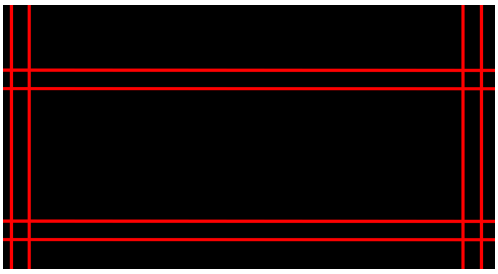
The reason for testing these two functions was with the intent of trying to lose some of the noise that might come with the images, but also to bring back the contour of the shapes to their initial form. As seen in **Figure 1b**, the erosion would lose a bit of the original width of the shapes compare to the original image (**Figure 1a**). Unfortunately, the opening and closing functions did not give a satisfactory result, even though extensive testing was done with all the test images as well as changing the kernel variables. Thus, the decision was taken to move forward only with an erosion function.

When scaling up towards the floor plans, we can start seeing some of the shortcomings of erosion as well, as seen in **Figure 1d**, concluding that this step needs more research.

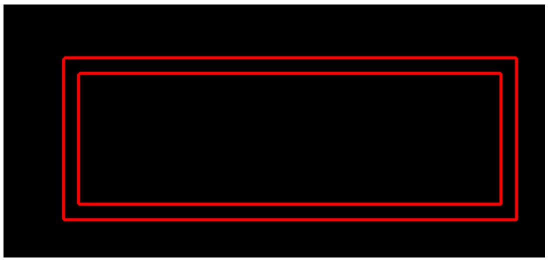
4.1.2 Hough Transform

Hough Transform was chosen for testing as it can automatically detect line edges. After extensive testing, the decision was taken to use the *Probabilistic Hough Transform* over the *The Standard Hough Line Transform*. The reason for this can be explained and exemplified when by observing the differences between **Figure 1a** and **Figure 1b**. **Figure 1a** detects and draws lines throughout the image size, making the *Probabilistic Hough Transform* the better choice for extracting coordinates, since the lines, in this case, mimic the actual lines of the shapes.

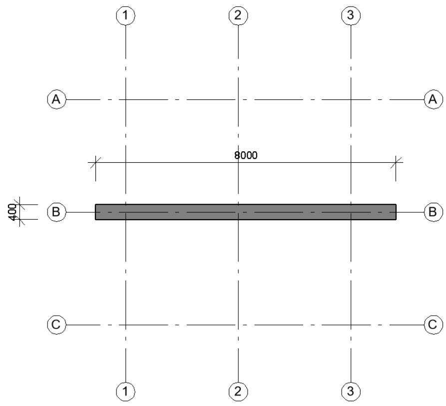
Unfortunately, there seem to be discrepancies in the performance of the *Probabilistic Hough Transform*, when attempting some tests compared to others. The best example is when testing it on the simple wall (**Figure 1c**). As seen in **Figure 1d**, the algorithm does not capture the two vertical sides of the wall. In conclusion, more experimentation needs to happen, especially with the kernel variables.



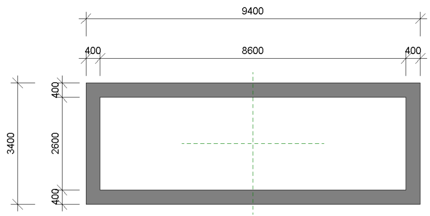
(a) The Standard Hough Transform



(b) The Probabilistic Hough Transform

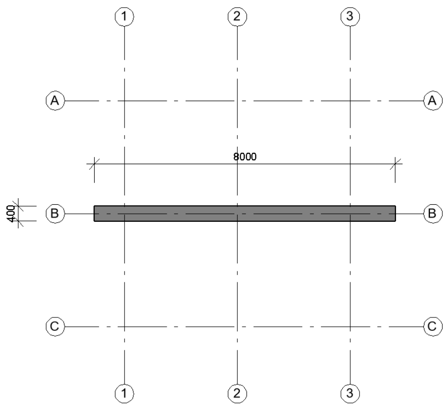


(c) Eroded Simple wall



(d) The Probabilistic Hough Transform on simple wall

(e) The OCR algorithm when testing it against the simple wall with constructing lines



(f) Simple wall with constructing lines

4.1.3 OCR

The implementation of *Optical Character Recognition* was rather straightforward. Most of the time, during testing the *OCR* managed to successfully recognize the dimensions of the shapes. In cases where it didn't recognize characters, when the image was flipped, the issue was solved. The notable issues that need work have to do with instances where the *OCR* is scanning images with constructing lines (**Figure 1f**). In this example (**Figure 1e**), erroneous data was retrieved.

4.2 RECONSTRUCTR

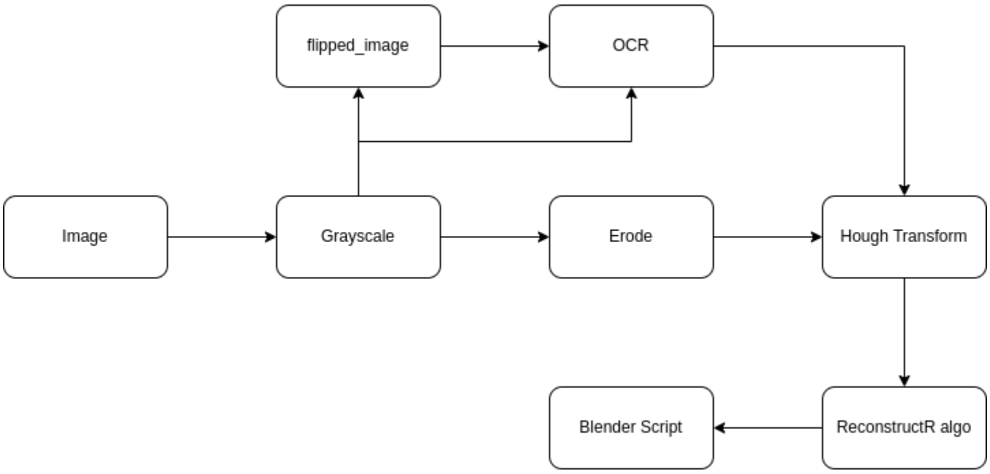
Based on the findings from the previous section, with the reflections given at the end of the structured research and finally with the level of deliverable which was agreed in the *Project Agreement*, namely "a program prototype that will try to perform a simple 3D reconstruction of a shape that might be interpreted as walls, out of 2D technical drawings, onto a format which is intermediate to a BIM model".

To be more explicit, the goal was to develop a script that could retrieve useful information from architectural floor plans and reconstruct a 3D representation onto a BIM-grade data format. For V. 1.0, the decision was taken to introduce some intermediary steps, since the data structures behind making BIM models were not covered in this research project. Therefore, Speckle which is an open-source 3D computer graphics software toolkit gives the opportunity to seamlessly convert between 3D file formats, including Blender. Thus, as seen in **Figure 1a**, for ReconstructR V. 1.0, the data retrieved will be made into a Blender script, afterwards, the file in itself being converted to a BIM format with Speckle.

As seen in **Figure 1a**, the logical process was to scan a 2D architectural floor plan and *Grayscale* it (even though normally floor plans are B&W, this is implemented mainly as a safety measure).

Next, a flipped version of the image will be created, since based on previous experiments, in some situations the *OCR* and the *Probabilistic Hough Transform* have difficulties detecting lines and respectively text that is flipped 90 degrees. The image will go next through *Erosion* for noise removal, through the *Probabilistic Hough Transform* for line and coordinate data retrieval. Finally, the data from *Probabilistic*

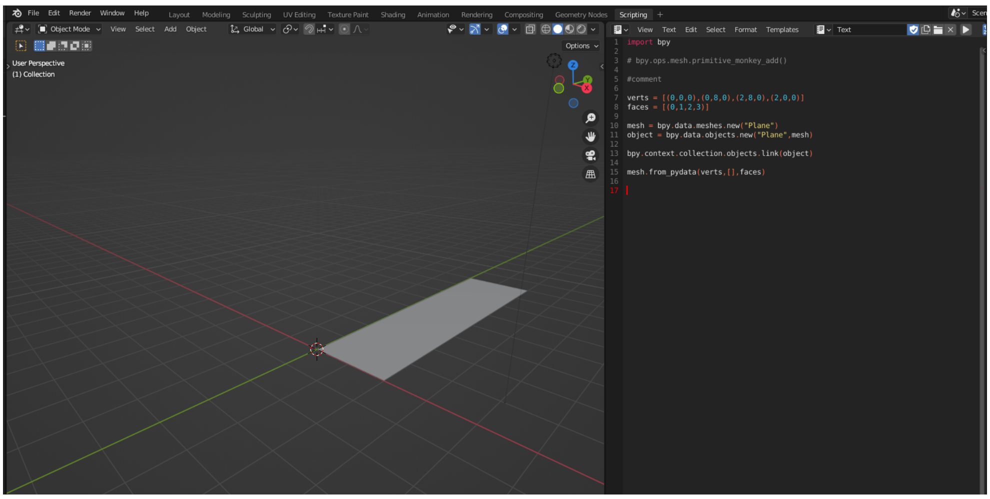
(a) Data Flow



(a) Data Flow

Hough Transform and *OCR* will be computed by *ReconstructR algorithm*, generating a *Blender Script*.

The Python code for this implementation attempt can be found under the ***ReconstructR/src*** path, but unfortunately, the algorithm charged with mapping the coordinates from the *OCR* and *Probabilistic Hough Transform* could not be fully implemented. Nevertheless, the *ReconstructR.py* file contains at the end a simulation of how the script would be generated if an image of the 8000 x 400 mm. simple wall was scanned. The script was then pasted into Blender and the results can be seen in **Figure 1a**:



(a) Blender Python Script Execution of simple wall

4.3 DISCUSSION AND REFLECTION

One of the big takeaways from experimenting with the functions with the test images is the size of the image being used as a source.

This was prevalent, especially when trying to work with the architectural floor plan (**Figure 1f**). A way to improve the *Opening/Closing* solution could be done by just adding a function that colours white all the black pixels. But there might be a risk of colouring the thin black contour at the edge of the shape, thus facing the same predicament as previously presented. Another approach could be with *Image Segmentation* with contour layering [14]. The same could be done to the OCR issue, where the text could be successfully "detached" from its original image and scanned separately. It is important to mention that these are only premises for future work and they need further experimentation. Overall the usage of *Computer Vision* and *OCR* seem plausible, but at this stage, they, unfortunately, do not scale. They of course do help, even though the results of this research project's experiments do not prove that. But the frequent usage identified in the *Structured Literature Review* does support that claim.

As a whole, these two tools would not suffice in solving the issue of reconstruction, something also shown by the reviews of the studies. Nevertheless, they cannot be dismissed as useful tools in the pursuit of building a software system.

5 CONCLUSION

5.1 LIMITATIONS AND THREATS TO VALIDITY

As our study was limited in the time and resources available, the findings are far from conclusive. The two most significant aspects of the first part of this research project that necessitates presenting are:

The inherent limitations brought on by a Structured Literature Review. A systematic literature review can map out existing solutions before a researcher attempts to tackle an area and it can help researchers avoid duplication and repetition in their work. Though, such an endeavour does not guarantee that all relevant material in a particular field will be found.

The inability to have accessed all the research papers that were included in the initial primary set of papers. This was due to the limitations imposed by some publications which required either a paid subscription or the purchase of the papers. In the case of those papers, if the paper was deemed useful based on its Abstract, a reverse snowballing was conducted based on their bibliographies.

In terms of the experimentation part of this research project, it is important to recognize that not all possibilities of using Computer Vision and OCR were exhausted. Thus, what can be said is that the work performed on the second part of this research project represents just a start in exploring and building more efficient software systems.

5.2 CONCLUSION

Within the background of a need for a robust solution for 3D reconstruction of buildings based on materials such as architectural floor plans to Building Information Modelling (BIM), the research project sets out to explore the current technologies, processes and software solutions available today, that attempt to address this need. In the first part of the research project, a structured literature review is performed with the intent of answering the questions of what are the existing solutions and processes to 3D reconstruction in AEC and what technologies are being utilised in the pursuit of addressing these solutions.

The findings suggest that from a commercial standpoint, given the complexities of requirements, lack of standardization, and limited options to meet these difficulties. To advance a workable solution, radical leaps must be taken before anyone can attempt to build a viable and implementable solution and meet the AEC industry's needs. One thing is certain: BIM is undeniably important since it is a standard that the entire AEC sector is moving towards. As a result, it may appear more practical in the short term to advocate for a series of services that apply a variety of methodologies and processes meant to speed up the reconstruction process, rather than seeking an all-encompassing automatic software solution. What looks promising though, is a combination of Machine Learning, Computer Vision and Optical Character Recognition, by trying to gather any information possible, not only from architectural floor plans, but also terrestrial and aerial imagery, as well as any other technical documentation that might come with the original paper folder. Based on the findings of the first part of the project, in the second part, the implementation of 3D reconstruction software development is attempted using Computer Vision and Optical Character Recognition. The experiments performed indicate that these two technologies alone are not enough to build a software system capable of a semi-automatic or automatic 3D reconstruction, though this research project cannot definitively conclude that since it was not an exhaustive implementation of these two technologies. As suggested in the Future Works section, further work of experimenting with Segmentation. Line and Object detection should be pursued, as well as Machine Learning models if the appropriate data set is found.

BIBLIOGRAPHY

[1] F. I. Apollonio, M. Gaiani, and Z. Sun. 3d modeling and data enrichment in digital reconstruction of architectural heritage. *International archives of the photogrammetry, remote sensing and spatial information sciences*, 5:W2, 2013.

[2] V. Badenko, A. Fedotov, D. Zotov, S. Lytkin, D. Volgin, R. Garg, and M. Liu. Scan-to-bim methodology adapted for different application. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:1–7, 2019.

[3] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon. Automatic line matching and 3d reconstruction of buildings from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, volume 32, pages 69–80, 1999.

[4] T. M. Breuel, A. Ul-Hasan, M. A. Al-Azawi, and F. Shafait. High-performance ocr for printed english and fraktur using lstm networks. In *2013 12th international conference on document analysis and recognition*, pages 683–687. IEEE, 2013.

[5] A. Budroni and J. Boehm. Automated 3d reconstruction of interiors from point clouds. *International Journal of Architectural Computing*, 8(1):55–73, 2010.

[6] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[7] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek. A brief introduction to opencv. In *2012 proceedings of the 35th international convention MIPRO*, pages 1725–1730. IEEE, 2012.

[8] D. Daniel. How big data is transforming architecture | architect magazine. 04 2015. (Accessed on 05/09/2022).

[9] A. R. Dick, P. H. Torr, and R. Cipolla. Automatic 3d modelling of architecture. In *BMVC*, volume 1, pages 372–381, 2000.

[10] G. Dodig-Crnkovic. Scientific methods in computer science. In *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden, Skövde, Suecia*, pages 126–130, 2002.

[11] C. M. Eastman. *Building product models: computer environments supporting design and construction*. CRC press, 2018.

[12] M. E. Esfahani, C. Rausch, M. M. Sharif, Q. Chen, C. Haas, and B. T. Adey. Quantitative investigation on the accuracy and precision of scan-to-bim under different modelling scenarios. *Automation in Construction*, 126:103686, 2021.

[13] F. Fassi et al. 3d modeling of complex architecture integrating different techniques–a critical overview. *3D-ARCH 2007 Proceedings: 3D Virtual Reconstruction and Visualization of Complex Architectures, ETH Zurich, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36:5, 2007.

[14] M. Gaurav. 4 image segmentation techniques in opencv python - mlk - machine learning knowledge. `https://machinelearningknowledge.ai/image-segmentation-in-python-opencv/#:~:text=%2C%200%2C%20128`, 09 2021. (Accessed on 05/14/2022).

[15] L. Gimenez, S. Robert, F. Suard, and K. Zreik. Automatic reconstruction of 3d building models from scanned 2d floor plans. *Automation in Construction*, 63:48–56, 2016.

[16] N. Haala and M. Kada. An update on automatic 3d building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):570–580, 2010.

[17] L. M. Hoffstaetter Samuel, Juarez Bochi. pytesseract · pypi. (Accessed on 05/09/2022).

[18] J. Howse and J. Minichino. *Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning*. Packt Publishing Ltd, 2020.

[19] S. M. Janamejay Joshi, Junaid Shaikh. Extracting semantic and quantitative information from architectural floor plans using deep learning networks. *Not Specified*, pages 1–15, 2020.

[20] A. Kaehler and G. Bradski. *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc.", 2016.

[21] U. known Author. Opencv: Introduction. (Accessed on 05/08/2022).

[22] U. known Author. Speckle connectors. `https://speckle.systems/features/connectors/`, 05 2022. (Accessed on 05/09/2022).

[23] U. known Author. Speckle systems: About. https://speckle.systems/about/, 05 2022. (Accessed on 05/09/2022).

[24] U. known Author. The evolution of engineering and architectural technicians. https://www.digitalschool.ca/the-evolution-of-cad-for-engineering-and-architectural-technicians/#:~:text=In%20the%201970%27s%2C%20early%20CAD,architectural%20and%20engineering%20firms%20everywhere., no date. (Accessed on 05/09/2022).

[25] A. Kofod-Petersen. How to do a structured literature review in computer science. *Ver. 0.1. October*, 1, 2012.

[26] B. Krutika. Hough transform using opencv | learnopencv. https://learnopencv.com/hough-transform-with-opencv-c-python/, 03 2019. (Accessed on 05/14/2022).

[27] J. Lim, P. Janssen, and R. Stouffs. Automated generation of bim models from 2d cad drawings. *Not Specified*, 2018.

[28] T. Lu, C.-L. Tai, L. Bao, F. Su, and S. Cai. 3d reconstruction of detailed buildings from architectural drawings. *Computer-Aided Design and Applications*, 2(1-4):527–536, 2005.

[29] T. Lu, C.-L. Tai, F. Su, and S. Cai. A new recognition model for electronic architectural drawings. *Computer-Aided Design*, 37(10):1053–1069, 2005.

[30] A. M. Lytle. *A framework for object recognition in construction using building information modeling and high frame rate 3D imaging*. PhD thesis, Virginia Tech, 2011.

[31] J. Memon, M. Sami, R. A. Khan, and M. Uddin. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, 8:142642–142668, 2020.

[32] U. Murthy, D. Boardman, and C. Garg. Assessing the value of 3d reconstruction in building construction. *arXiv preprint arXiv:1201.3172*, 2012.

[33] U. nknown Author. Blender (software) - wikipedia page. https://en.wikipedia.org/wiki/Blender_(software). (Accessed on 05/11/2022).

[34] U. nknown Author. Facility management - wikipedia page. https://en.wikipedia.org/wiki/Facility_management#cite_note-Eltringham-3Aug2017-7. (Accessed on 05/12/2022).

[35] U. nknown Author. Morphological image processing. https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm. (Accessed on 05/12/2022).

[36] U. nknown Author. Opencv: Color conversions. https://docs.opencv.org/4.4.0/de/d25/imgproc_color_conversions.html. (Accessed on 05/11/2022).

[37] U. nknown Author. Opencv: Eroding and dilating. https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html. (Accessed on 05/12/2022).

[38] U. nknown Author. Opencv: Hough line transform. https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html. (Accessed on 05/14/2022).

[39] U. nknown Author. Opencv: Smoothing images. https://docs.opencv.org/4.x/dc/dd3/tutorial_gausian_median_blur_bilateral_filter.html. (Accessed on 05/13/2022).

[40] U. nknown Author. Photogrammetry - wikipedia. https://en.wikipedia.org/wiki/Photogrammetry. (Accessed on 05/11/2022).

[41] V. Pătrăucean, I. Armeni, M. Nahangi, J. Yeung, I. Brilakis, and C. Haas. State of research in automatic as-built modelling. *Advanced Engineering Informatics*, 29(2):162–171, 2015.

[42] H. Penttilä. Describing the changes in architectural information technology to understand design complexity and free-form architectural expression. *Journal of Information Technology in Construction (ITcon)*, 11(29):395–408, 2006.

[43] F. Remondino and S. El-Hakim. Image-based 3d modelling: a review. *The photogrammetric record*, 21(115):269–291, 2006.

[44] F. Remondino, S. El-Hakim, S. Girardi, A. Rizzi, S. Benedetti, and L. Gonzo. 3d virtual reconstruction and visualization of complex architectures: The 3d-arch project. *International archives of the photogrammetry, remote sensing and spatial information sciences*, 38(5/W1), 2009.

[45] H. F. Schantz. *History of OCR, optical character recognition*. Recognition Technologies Users Association, 1982.

[46] T. Schenk. Introduction to photogrammetry. *The Ohio State University, Columbus*, 106, 2005.

[47] K. Schindler and J. Bauer. A model-based method for building reconstruction. In *First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003. HLK 2003.*, pages 74–82. IEEE, 2003.

[48] S. M. Sepasgozar, S. Shirowzhan, et al. Challenges and opportunities for implementation of laser scanners in building construction. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 33, page 1. IAARC Publications, 2016.

[49] S. Sinha. Python | grayscaling of images using opencv - geeksforgeeks. `https://www.geeksforgeeks.org/python-grayscaling-of-images-using-opencv/`. (Accessed on 05/11/2022).

[50] R. Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

[51] H. Son, C. Kim, and Y. Turkan. Scan-to-bim-an overview of the current state of the art and a look ahead. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 32, page 1. Citeseer, 2015.

[52] I. Suveg and G. Vosselman. 3d reconstruction of building models. *International archives of photogrammetry and remote sensing*, 33(B2; PART 2):538–545, 2000.

[53] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[54] C. Talamo and M. M. Bonanomi. The impact of digitalization on processes and organizational structures of architecture and engineering firms. *Digital Transformation of the Design, Construction and Management Processes of the Built Environment; Daniotti, B., Gianinetto, M., Della Torre, S., Eds*, pages 175–185, 2020.

[55] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in construction*, 19(7):829–843, 2010.

[56] C. Tappert. Cy suen. and t. wakahara, the state of the art in on-line handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell. PAM]-12, l990*, pages 787–808, 1990.

[57] G. Vosselman, S. Dijkman, et al. 3d building model reconstruction from point clouds and ground plans. *International archives*

of photogrammetry remote sensing and spatial information sciences, 34(3/W4):37–44, 2001.

[58] C. Wang and Y. Cho. Automated 3d building envelope recognition from point clouds for energy analysis. In *Construction Research Congress 2012: Construction Challenges in a Flat World*, pages 1155–1164, 2012.

[59] Q. Wang, J. Guo, and M.-K. Kim. An application oriented scan-to-bim framework. *Remote sensing*, 11(3):365, 2019.

[60] L. Yang, Y. Li, X. Li, Z. Meng, and H. Luo. Efficient plane extraction using normal estimation and ransac from 3d point cloud. *Computer Standards & Interfaces*, 82:103608, 2022.

[61] Z. Zeng, X. Li, Y. K. Yu, and C.-W. Fu. Deep floor plan recognition using a multi-task network with room-boundary-guided attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9096–9104, 2019.