Moving From Targeted Acquisition To Urban Area Modelling - Increasing The Scale Of Point Cloud Processing

Matt Courtney, Yacine Rezgui, Tom Beach, Jean-Laurent Hippolyte, Jonathan Reynolds
BRE Trust Centre for Sustainable Engineering
Cardiff University
Cardiff, United Kingdom
courtneym1@cardiff.ac.uk

***Abstract*—Handling real world, acquired point cloud data within a sector such as architecture, engineering, and construction (AEC) is currently a difficult task. A highly desirable, future goal is to fully automate the scan-to-BIM process which at this time has a high dependency on manual work effort. Improvements within this workflow will speed up the production of detailed 3d building models and reduce associated costs. By increasing the level of automation in the scan-to-BIM process it becomes possible to speculate the expansion of the typical use case from a single structure, targeted acquisition towards urban area data collection and modelling. The scale and characteristic differences of an urban area point cloud dataset and that of a single structure create opportunities to validate the applicability of novel analytical approaches to process automation. A decrease in process complexity could be achieved by reducing both the depth of prerequisite knowledge and the level of intervention expected from an operator by a modelling platform. This would also provide an alternative perspective and an opportunity to model operator tasks at a higher, more abstract level. There lacks a completeness of modern documentation within preexisting civil structures. Building information modelling of the as-built condition can reduce overheads associated with key areas such as collaboration, maintenance, and future modifications. However, it is often made more difficult by a lack of accurate documentation due to the age of the building. A common trend that can be observed in countries such as the United Kingdom is that much of future building stock is already standing. Therefore, it is apparent the need to record the detailed, as-built condition of structures essentially from scratch will not resolve itself in the near future. This paper will overview a case study of an urban area modelling conducted in Ebbw Vale, Wales and introduces an abstract scan-to-BIM process automation methodology. This will be supported by a review of a selection of applied research literature. This paper is part of the early development stages of a point cloud processing platform, pcl_toolkit. The command line software aims to simplify approaches commonly associated with various point cloud processing tasks and provide a foundation for rapid development in the near future.**

***Keywords*—building information modelling; urban area modelling; scan-to-bim; point cloud; data processing**

I. INTRODUCTION AND BACKGROUND

There is currently a high level of complexity associated when working with point cloud data, especially with regards to semantic model generation. There are numerous sources of perceived complexity that could be considered. Examples such as the commonly large size of the point cloud data set [1], the influence of various acquisition conditions such as light [2], or the potential for major differences between one point cloud and another. The unique attributes of each point cloud can make it difficult to describe behaviour through any means that is too rigid such as using traditional, rule based techniques. Huber et al [3] exemplifies the need to capture a deeper level of contextual data for classification operations in point cloud processing. The common characteristics of the generic dataset are key considerations when trying to address point cloud processing as a mass task after more granular operations have been implemented.

In the context of the scan-to-BIM process and current efforts to streamline this multiple staged task there are obvious benefits from increasing the depth and breadth of a point clouds semantisation. The works of Dimitrov and Golparvar-Fard [4] employed material recognition within a construction progress monitoring environment. This allowed a system to associate material with planes; automating a subprocess. Innovations such as these which add further semantic information to a point cloud act to reduce the current active workload through automation. They are essential progressions towards a goal of automated scan-to-BIM.

The Point Cloud Library (PCL) [5] is a C++ library that provides access to often state of the art point cloud tools at a fundamental level [6]. It is therefore a common choice for experimental development projects that require a standalone piece of functionality. At the moment there are few efforts to ease the complexity of the development task. PCL provides powerful tools but does not offer any major development framework and a point cloud processing pipeline must essentially be implemented from scratch. If the task of developing a novel point cloud processing system was reduced in complexity whilst maintaining flexibility offered through the breadth of functionality of PCL an application such as scan-to-BIM could likely be further streamlined.

The C++ implementation of Point Cloud Library adds what could be considered limitations due to the language itself [7]. Point cloud processing should benefit from rapid development and native interpreted environments that add extra, powerful

tools within a language such as Python. A simplification at this level will allow for the development of more complex systems to become more attainable and therefore boost larger scale, potentially multiple instance processing architectures.

The development described throughout this paper aims to progress towards answering the following research questions:

- What is the level an automation methodology can relieve perceived complexities currently associated with the scan-to-BIM process?
- What is the relationship between point clouds within the desired acquisition catchment?
- How applicable are unsupervised learning approaches suited to large datasets within this domain?
- How robust is such a methodology with regards to performance on unseen, test data?

Following this introduction, a case study of an urban area point cloud project in Ebbw Vale, Wales. What is described illustrates some of the current complexities within a typical scan-to-BIM workflow using a generic representation of the available commercial tools. This will lead into a review of a selection of applied literature which gives insight into the field from the perspective of potential for experimental development. The following section defines a strategic framework that is being progressed until it becomes a tangible methodology for real world application and even future deployment.

System design details how factors such as system architecture influence the project and plan to boost future development efforts. The next section (system features) contains a description of the behaviours currently implemented and made available via the pcl_toolkit. The paper concludes with the near future, planned work that will further the project in the direction described throughout.

II. CASE STUDY - EBBW VALE

The authors have completed a model generation operation of a pilot site in Ebbw Vale, Wales as show in Fig. 1. The project went from the point cloud data acquisition stage to production of a BIM representation for each structure within a designated urban area. This made for an opportunity to discover the internal processes and resultant difficulties of the scan-to-BIM workflow first hand. The regeneration project is located on the site of an old steelworks and now hosts a range of facilities such as a school, a leisure centre, and a higher-education college. The site also includes a dedicated energy centre that supplies to a district heating network for the utilisation of the whole urban area.

The external of the site was scanned in the space of a week using a Faro 3D Focus X 130 with a range of up to one hundred and thirty metres. Due to a lack of time as well as other constraints such as a restricted level of internal access only the outside of the structures were scanned. Instead, up to date architectural plans were provided by site owners in order to support the creation of complete BIM representations. The end goal of the operation was to produce models to support an energy profiling of the site. As a result, provided alongside the architectural drawings were a set of detailed mechanical

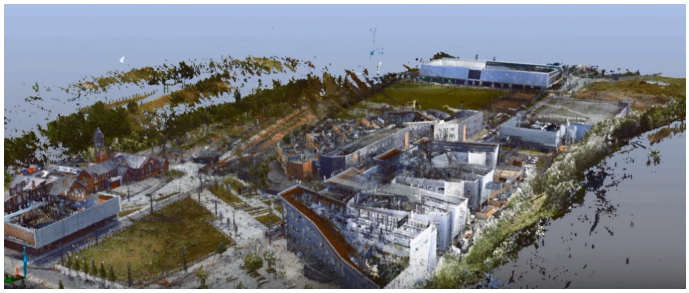

Fig. 1. Ebbw Vale, Wales - Completed Point Cloud

and electrical technical drawings that provided a knowledge base for both air handling and air conditioning. This provided an excellent means for producing an energy model as well as quantifying the value of such information being available on new build sites.

From the point of view of attempting to conduct an assessment into the potential level of total process automation feasible the difference between a civil structure and what can be considered an urban environment is apparent. There is already great potential for variance between structures in a single target, civil context but in an urban area this complexity is likely to be further increased. As a result, any novel automations to sub processes must be highly robust in order to better address not just the increased potential for high-level variance in the produced dataset but also the greatly increased quantity of acquired data.

The challenges faced during the process of BIM production in this project can be interpreted as an indication of some of the areas to target in future work to optimise the scan-to-BIM workflow and progress towards a higher level of automation. The main area being how the generated point cloud was utilised for the generation of a BIM. Common to industry software packages such as Autodesks Revit provide native support for many point cloud formats allowing for an easy import into the platform. Due to the nature of the acquisition task the completed point cloud could be broken down into individual structures; allowing the operator to produce a parametric representation of a selected building. Once a point cloud is imported it is used as a referencing framework and within the software a digital model can be developed on top of the point cloud. This is in a similar way an image might be projected onto a canvas to provide the initial references for a painting. It could be argued that the current approach outlined is not fully utilising all potentials for subprocess automation [8] as the point cloud is never interpreted by a system, instead a system is used as a presentation medium only.

When creating a structural model conforming to the outlined methodology the skill of the modeller can be considered highly relevant to the quality of the produced model [9]. In some cases where a model is very complex and the difficulty of using a point cloud for references is heightened further a high skilled modeller may be considered a hard constraint. In order

to identify what tasks should be simplified or automated first a philosophy of reducing the likelihood of this occurrence should be adopted as a heuristic in order to target functionality. If the assumed operator competence were to be reduced to being described as savvy the system could be considered to have actively reduced both the quantity and complexity of the remaining manual workload.

This project acts as a representation of the current scan-to-BIM process which is likely to be reproduced in a similar fashion elsewhere. Inside this one case study it is not difficult to find areas of the total process that lack a deep enough integration within a developed specialised system. As the creation of a detailed BIM from a point cloud currently requires operator intervention in order to be able to interpret a given dataset at even the basic, parametric level there is likely potential for new optimisations offered through intelligent automation.

The authors also undertook an acquisition task as part of the collaborative REACH project, generating a large point cloud dataset. The modelling process is currently under way to map the influence of seismic activity within the area of Sichuan, China. This undertaking and produced documentation will later facilitate a case study to support validation of the pcl_toolkit in a similar manor to the work conducted at the Ebbw Vale site.

III. LITERATURE REVIEW

Scan-to-BIM is dependent on point cloud technology. These can be acquired through various means with the most modern and powerful being via 3d laser scanner [10]. Over other available techniques this offers the highest resolution and so the dataset representation generated is at its largest [11]. Full conclusions have not yet been drawn of how much this richer dataset translates into perceivable benefits or whether this higher resolution is removed at a filtering stage. The 3d laser scanner offers a self-contained point cloud acquisition and generation system requiring far less steps from an operator perspective over more attainable solutions such as photogrammetry using a DSLR camera [12]. It is therefore likely that the trend of generating point clouds from laser scanners to support the creation of an as-built BIM will continue to grow as price points for such units lower [13].

Characteristically, the scan-to-BIM process is a multiple staged task and it can be broken down into a set of more manageable subprocesses. This breaking down of the overall challenge of automated BIM production allows for research to be specialised to addressing a part of the whole problem; progressing the field towards an end goal. At the early stages that this research area finds itself in the task for a researcher is as much problem discovery as it is problem solving. This will settle and at this point in time the breadth of issues can be considered fully known.

The works of Barazzetti [14] exemplifies this current trend in approach to research where Non-Uniform Rational B-Splines (NURBS) were employed in order to automate the forming of a complex roof structure. This allowed for the creation of a parametric, as-built representation of a building roof from a point cloud dataset. By addressing this key difficulty in the project it allowed for the generated roof model to be coupled with conventional, industry approaches in order to produce the remaining elements required of the model. The described NURBS application stands testament to the need for automation at a subprocess level in order to streamline the scan-to-BIM task. A complete solution is likely to be a collection of specialised tools grouped into a multi purpose package over discovering a scan-to-BIM automation silver bullet. As research efforts continue, those specialised solutions that stand unparalleled or unimproved will become standard tools in future methodologies.

Another common issue which is encountered frequently, especially so in interior point clouds is occlusion. The nature of internal laser scans is that they are more often than not performed in already occupied spaces and so occlusion is very likely to be encountered in some form. Xiong et al [15] recognise the significance of occlusion and as a result assume in their work that experiencing occlusion is the norm in this scenario. The real world applicability of developed solutions from research that does not make this assumption could be questioned and is in the aforementioned paper. Such projects are tested in naturally less occluded environments such as hallways and so proposed techniques are not pushed to a feasible level in their test environment. Occlusion is a major factor of overall complexity and so should have a strong influence on research that can potentially encounter it.

A key element of the future of automated processes is the level that a system is able to interpret what is captured, there is a need for a means of understanding the context within a point cloud. A system should be able to recognise this detail and semantically associate the subset of the point cloud dataset without the need of operator intervention. A method of furthering the amount of metadata captured could be through the addition of novel sensors to the current array. This approach was taken by Valero et al [16] who employed RFID technology to physically tag their interior, occluded environment. The system allowed for a standard point cloud acquisition rig to be extended in a way that it could more easily capture important data such as what is the occlusion in the scene scanned. This was a trade off solution as the addition of new sensor technologies adds some minor complexities. However, the reduction of the influence of occlusion during the later processing stages lead to an observed increase in modelling process efficiency within their test environment.

The European Union estimates that of the building stock to be available in 2050 that eighty percent is already built [17]. Of these, eighty percent will have been built before 1991 [18] and as a result are unlikely to have any modern modelling or documentation available to support as-built model production. It can therefore be safely assumed that current issues associated with BIM generation for the civil landscape or in urban areas are not going to resolve themselves soon. Hypothetically, if a mass system was built that could automatically process a point cloud (defined by outputting a desired model representation) it

would surely consist of the most suited atomic solutions where suitability is judged by many criteria such as the robustness of the application. Further, a finalised solution must address and nullify the commonly faced issues being discovered and raised in the surrounding research field.

IV. PROPOSING A FUTURE METHODOLOGY

As the uptake of the developed technologies increases within relevant industry so too will the demands that are put on it. A methodology that offers a high level of process automation must take this likely factor as a key consideration. There is a need to increase testing scale to a real world processing environment in order to progress to a more commercially stable platform.

As previously mentioned, there is a likelihood to up scale a data driven process such as point cloud analysis towards a maximum potential use case. In scan-to-BIM this will be the move to consider an urban area as a modelling target rather than a single structure. A method of creating a platform that supports suitable operations at an urban environment level is by allowing for further semantic associations to be drawn. From the perspective of the challenge of automating scan-to-BIM exploring the benefits of such a change in scale will allow for novel developments at various stages of the process methodology.

Knowledge from data science can be drawn into relation with scan-to-BIM where mass data sets are considered at a high resolution in order to find patterns that support recognition. Patterns in such a dataset may be less easily detected had it been segmented or reduced and smaller scale techniques used instead [19]. There is possibility to trial such an approach and also test the feasibility of creating hybridised processing architectures with the ability to make multiple passes of an inputted point cloud dataset, potentially in parallel. A point cloud representation of an urban area would be a very large dataset and so becomes potentially applicable to a wide range of analytical and recognition techniques suited to large scale datasets.

From an automation perspective a scan-to-BIM process can be logically divided into the following high level subsections: point cloud acquisition, proprietary preprocessing, external preprocessing, recognition or analysis, before concluding with the necessary post processing steps. Within this workflow however is the potential for many disjoint stages as well as the need for transcoding in order to bridge proprietary file formats and different software packages. The scan-to-BIM process from an automation perspective is a complex problem with many of its own stages different to those of the traditional, operator centric approach. Currently, there is an influence on the scan-to-BIM process from laser scanner brands where a proprietary file format is employed [20]. This creates a well-supported but self-contained registration process and a difficult to avoid divide between the preprocessing that needs to be completed within proprietary software and what can be outsourced and potentially automated. The software package bundled with the laser scanner used in the Ebbw Vale case study allowed for a powerful point cloud registration process so boycotting this functionality and reimplementing is difficult to consider worth-while at this current maturity point in laser scanner technologies. If in the future the majority of laser scanners default to supporting open formats then there may be potential for consideration of undertaking optimisation of this process. However, at this present time a scan-to-BIM automation methodology can be considered to begin after the creation of a registered collection of point clouds.

Key areas for future automation should consider external preprocessing and beyond in the outlined scan-to-BIM workflow illustrated in Fig. 2. Within the external preprocessing there is scope for tasks such as applying filters, transformations, or segmentation algorithms. These may be undertaken in order to prepare the dataset to be exported in a new form or to be fed back into the overall procedure. This project focuses on this stage of the potential automation process. The pcl_toolkit to be described works to offer this behaviour in a streamlined manor. This approach brings the workflow to a higher level and creates a more logical scripting style that is suitable for analysis as well as supporting large scale processing architectures.

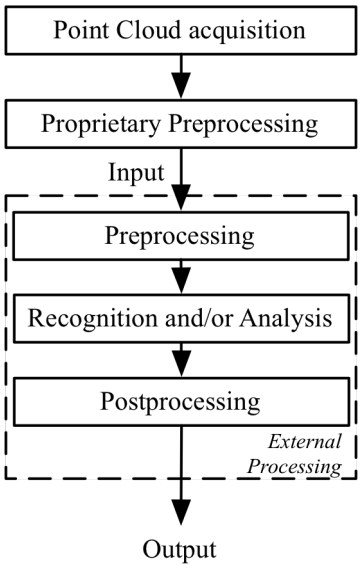

Fig. 2. Scan-to-BIM Methodology

The next logical subsection of automation is the application of recognition or analytical techniques in order to support some logical interpretation. This could consist of any number of different strategies that at a high level could be described as mapping a formatted point cloud to a desired output. Research in this particular area might consider working to include machine learning approaches that are suited to handling large, complex datasets. Creating a deeper, semantically inclusive representation of typical point cloud data may support further automation and black boxing of the entire process. This could be achievable after the processes have been streamlined as far as is perceived possible. For example, the system may support

some form of decision model of the process; allowing the required operator intervention to be reduced further.

The final part of the scan-to-BIM workflow could be considered the postprocessing stage. This should be thought to occur after all major preparation of the output has been completed. In order to support compatibility within the current ecosystem of software and therefore work to maintain an ease of integration an automation methodology must also include formatting and transcoding behaviours. The ability in which a novel application can integrate within a targeted environment is achieved by working to support commonly used formats as input and output [21]. This attribute also has the added benefit of allowing the developed project to become a platform for expansion at a later date. This section of the process is a less important consideration as the postprocessing steps are often fairly specialised and therefore difficult to automate. More research needs to be conducted with regards to a total automation methodology and therefore it is difficult to speculate the extent that a methodology for scan-to-BIM automation can be feasibly implemented. However, progression in this direction has potential to yield a system that requires operator intervention at pre determined intervals or when a decision cannot be reached within an acceptable threshold. A system such as the outlined has potential to reduce the complexity of the total workload of the scan-to-BIM process even if it does not achieve a full level of automation. Developing a system in this area of research will also work to discover a perceivable boundary of automation based on the currently available techniques. Optimisations can then be back propagated from this automation roof and implemented behaviours can be further improved.

The designed methodology as show in Fig. 2 includes a boundary isolating the input and output side effects from the contained external processes. This reflects a membrane that may possess minor behaviours to modify the input prior to processing and output after completion that would not be considered as real preprocessing or postprocessing, respectively. Also, within the diagram the boundary describes an atomicity of the three contained processes as a collection. This design choice aims to offer better support for parallelisation and further hybridisation of different processes. This is aided by the side effects being separated from other behaviours; an external processing black box may contain multiple instances of the three internal processes. As any analytical or recognition based findings do not need to be directly comparable between other point cloud processing jobs different approaches can be used and combined in order to attempt to produce the highest quality output rather than maintaining the conditions of a fair test required for cross analysis.

V. SYSTEM OVERVIEW

The first complete iteration of the pcl_toolkit is a targeted implementation of the functionality made available by the Point Cloud Library. Created is a black box solution of selected PCL features. From an external perspective, the pcl_toolkit offers a key reduction in complexity. It is no longer necessary to expose the Point Cloud Library to those who wish to use already provided functionality but do not have a need to implement new behaviours. The pcl_toolkit aims to streamline the workflow commonly associated with point cloud preprocessing and analytical tasks from writing code to higher level scripting. Like implementation hiding this further reduces task complexity as input can produce output via commands rather than manually written code.

The creation of a robust PCL implementation allows for a collection of operations to be performed more quickly. This reduces the current difficulty of entry associated with working with point cloud data as there is no longer a need to have a command of a general purpose programming language. Instead, as the initial pcl_toolkit provides a command line interface (CLI) a user only needs to have basic experience working in a terminal environment. This centres focus around approach and optimisation by actively reducing associated development overheads.

The functionality of PCL depends on a set of metaparameters. These must be well tailored to not only reflect the characteristics of the point cloud data but also the capabilities of the host system and is something that currently cannot be set in advance. Operation configuration has a strong influence on system performance and so this must remain attached to the operator setup. As a result pcl_toolkit supports such values to be included with datasets. This provides the same level of intractability with functional behaviour that is found when code is written manually; the black box solution reduces complexity without reducing functionality.

The Point Cloud Library is able to leverage the benefits of low level access to system hardware from the C++ language in comparison to other, similar languages such as Java. This results in potentially state of the art algorithmic performance. When such high performance tasks are made more easily available by the pcl_toolkit and the desire is to script the behaviour then other language choices may become more suited. The inclusion of a Python wrapper works to offer an improvement to usability. Operations can be run in the native interpretative mode allowing for in terminal data processing as well as supporting running through a Python file. The Python paradigm is at a higher level versus C++, and adding this layer to the pcl_toolkit complements the core goals of the project.

From an implementation perspective, the use of a Python wrapper provides an excellent means to create a layer to handle compound operations. If left solely as compiled C++ a predefined logic would have to be written to handle multiple combinations of operations. This would then have to be reflected at a functional level with extra internal interoperability added. The inclusion of another language level means outputs can be held in memory with the C++ beneath passed to and returning. As less excess is performed at a code level this design choice works to maintain the elegance of the implementation by utilising the already available functionality of the Python interpreter.

The self-contained design of pcl_toolkit means that it can be run from any host system that meets the dependency

requirements. Further, pcl_toolkit is lightweight and focuses on performing operations on supplied input data in order to produce a desired output. This allows an installer the support to run pcl_toolkit on a single instance as well as in parallel by installing multiple instances across a potentially decentralised architecture. Point cloud data may be segmented in an appropriate fashion and each subset processed at the same time.

Initially the pcl_toolkit operates on the CPU as a single threaded application. However, future work may lead to support for a GPU implementation in order to offer better performance where such hardware is available to an installer. Point Cloud Library has native support for GPU utilisation and so adding this improved processing paradigm will allow for a powerful expansion to a wider range of system architectures. As the project progresses it may become possible to expand to a distributed processing methodology where each pcl_toolkit instance has access to GPU resources. This would lead to a very high performance approach to computational point cloud processing where the dataset supports being split.

Further future work will look to examine the applicability of various machine learning techniques in order to offer a deeper automation of the full process a user develops. If the developed system has a regular use pattern where similarities can be drawn between datasets then there may be an application for classification. The inclusion of this technique aims to test the applicability of classification tools to this type of work. If a system can learn the characteristic and tie this knowledge to configuration used by the operator then it could lead to a viable route to prediction on unseen cases. The pcl_toolkit is nave in that it disregards input each time. Instead of a total reliance on a user specifying configuration for each dataset they could provide a classification model for a batch of datasets. This may be a route to provide support for batch processing without the need for hard coded configuration, a system could instead select from a value pool of previous runs.

Machine learning could be used as a method to work towards a basic level of total automation of this task. Initially, the pcl_toolkit is designed to simplify the objective of performing common processing and analytical tasks on point cloud data. However if this is successful then it changes the process in a way that may more easily support automation and this should be explored. A route to providing a compatibility with a learning model format may be to support exports from a data mining tool such as Weka. Weka is one of the most popular of its kind and it is likely that if a classification model is produced for user operations then it could well be undertaken in Weka. If pcl_toolkit and Weka were successfully integrated then the overall intelligence of the system could be further increased which is beneficial when potential workloads are considered.

Python provides for a higher level of operations that naturally support compound tasks by holding returns in memory and passing them to the next function. This is not the only choice for a language wrapper that may suit this type of functionality. It could be argued that a functional language has potential to add a highly accurate representation of the behaviour of the pcl_toolkit. A functional language describes the output of what it is passed with no real exposure to any machine level operations. If such a wrapper were implemented then the language would focus on what is to be performed on the point cloud data, the technology of the language would further abstract away from the fine grained operations involved.

Another strength of functional languages is their ability to describe functionality at an abstract level without any need to handle type. This means that code written for point clouds using the pcl_toolkit may not function as expected if used outside of this context but it also works to reduce the complexity of working with point cloud types. In a C++ implementation there is the need for many namespaces so the compiler may be made aware of how to handle an extension on the core library of types. As variables are very different in a functional context this complexity is removed and rather the focus remains on the main part of what the pcl_toolkit aims to simplify, the point cloud operations.

The expressiveness of a functional language is slightly higher than that of Python as it is a specialisation in this philosophy. One of the key points for supporting the development of the pcl_toolkit was to reduce the complexity and difficulty associated with using the Point Cloud Library for point cloud data processing. It would have not made sense to try and simplify the process by developing only a functional solution. This is due to the functional programming skills being considered a niche within the development community and so it may even increase the overall difficulty if this was the only provided option. With this considered, if the benefits of providing different paradigms of language wrappers can be quantified and are beneficial then its development will likely be undertaken.

As the pcl_toolkit is in its first iteration the approach to delivering the functionality currently reflects this factor. The main body of code consists of a collection of functions that represent the different operations offered by the interface. However, there is still potential for further optimisations to be found through gaining a deeper understanding of the inter-relationships between the functions in the code. As the C++ includes work at the macro level there are no optimisations provided by the language design, the code lines are simply copied in to where the corresponding include statement is prior to compilation. Therefore, this is an expense that can be reduced when the code is cut down appropriately.

Syntactically, there is no real repetition in the code but there may be at an operational level, where sub functions could be considered repeated. This should be broken down into atomic functionality and this then centralised in relation to that which uses it. The resulting code may make more frequent function calls in order to offer the same behaviour but there will be less code and a further reduction in what can be considered repetition. Versus the previous inclusion there will be a shorter body of code to be inserted. This reduces compilation time, file size, and from a wrapper perspective the lifting weight and resulting resources required.

VI. SYSTEM DESIGN

Within the systems design shown in Fig. 3, the C++ manager works as a behaviour hub allowing the bridge between Python and C++ and also handling passing the supported operational structure, making the transition to internal function calls. This core will allow for further expansion of library based functionality in later development cycles. The addition of other point cloud related functionality can be achieved by creating a new standalone module, similar to those already used by the C++ manager. In order for the module to join non static methods need to be included in the central operation manager.

The simplicity of module addition is part of an agile, forward thinking design that will allow the pcl_toolkit to expand rapidly as to include more suitable functionality. At the early point in the life cycle the aim is to offer a strong basic platform that offers some of the core functionality at a high standard of usability. This is in hope of creating an ethos for the project of rapid further development and quick functionality deployment. Potential areas of expansion in this manor are the inclusion of a wider selection of image processing and machine learning tools. This will allow for abstraction beyond point cloud specific approaches and permit this type of data to have access to a wider range of processing options.

The aim of the pcl_toolkit is to offer powerful point cloud processing and analytical tools at a higher level of abstraction than a standalone C++ implementation of the Point Cloud Library. An example use case of the pcl_toolkit would be in a cluster processing configuration. Each node could have an instance of pcl_toolkit made available to it and therefore subsections of the dataset could be sent to each node for processing tasks. The pcl_toolkit could also play a part in the preprocessing prior to hitting the cluster, allowing the full point cloud to be broken down. Rather than this being a full implementation task pcl_toolkit reduces it to configuration settings and a short Python script.

The current installation on the suggested operating system (Ubuntu 16.04.2 LTS) encounters some complexity due to the dependency to Point Cloud Library and its required modules. As functionality expands beyond just PCL more dependencies will be passed on to the operator to manage prior to installation. A streamlining of this process offered by the pcl_toolkit will mean that it can be more quickly deployed onto a system. This will be an important feature when considering installations on clustered computing environments. In order to provide an appropriate setup on a complex architecture pcl_toolkit must be installable in a minimal amount of commands, handle core external dependency setup automatically, and also offer support for image cloning for ease of duplication.

A targeted platform would be a specialised installation on the recommended operating system. The resulting cleaner instance produces a more predictable base platform with only operational tools installed. Dependencies can be more easily managed at installation of the pcl_toolkit if the environment is configured in this way. If pcl_toolkit is installed on a general purpose, working system rather than a for purpose virtual server the compatibility difficulties may increase and can be less easily foreseen. This assumption of platform allows to further define the purpose of the system and maintain a focused direction of development.

VII. SYSTEM FEATURES

Of the modules described in Fig. 3 the filters play a key part in the preprocessing capabilities of pcl_toolkit. They allow for input to be reduced so that it can be more efficiently analysed using other features of the software. One of the benefits of applying well selected filters to point cloud data is that the physical task of processing is reduced in scale whilst the contextual knowledge can be mostly protected in comparison. Some of the reduction filters made available via the pcl_toolkit are the basics such as voxel grid and radius outlier. At the early stages of the development cycle these provide an excellent means to quickly implement robust and highly versatile filters that actively and effectively optimise data processing tasks.

Filters are commonly one of the initial processes and in pcl_toolkit they are represented by a standalone module. A filter can take an input and can be used to generate output meaning it can be used on its own for rapid dataset reductions. Alternatively, a filter can be added to a script and return to the next part of a processing pipeline. This functional approach to all of the modules aims to boost the flexibility of the application whilst also increasing the functionality.

The segmentation module provides a group of processes to create subsets of the input point cloud data. Approaches such as extract indices allow an operator to partition the dataset based on configured logic. In the context of this algorithm and its provision via the pcl_toolkit it offers a simplified means to break a point cloud into planes. Where appropriate this created subsection of the point cloud may be considered a disjoint entity. If assumed, this attribute may provide a basis for parallel processing in the future.

In a similar manor to that of the functionality of the filter module the segmentation algorithms can be supplied with direct from operator input and also returns from other compatible processes within pcl_toolkit. This key internal compatibility manifests as functionality to the operator by allowing the option to increase the complexity of the mass processing application on demand.

The I/O and visualisation modules locate the basic functionality needed to support the interface layers. Currently, file type is limited to that of Point Cloud Library .pcd which was a first choice early on in the project in order to quickly be able to provide functional implementations from PCL. This can be expanded to other common types when it becomes appropriate to offer such support.

Visualisation is handled by the pcl_visualization module and provides a user a basic level of feedback via this means. This tool is meant to provide for an extra method of debugging during development and may not progress much further in this capacity. This is due to the targeted use case being more inclined towards a server environment without a graphical user

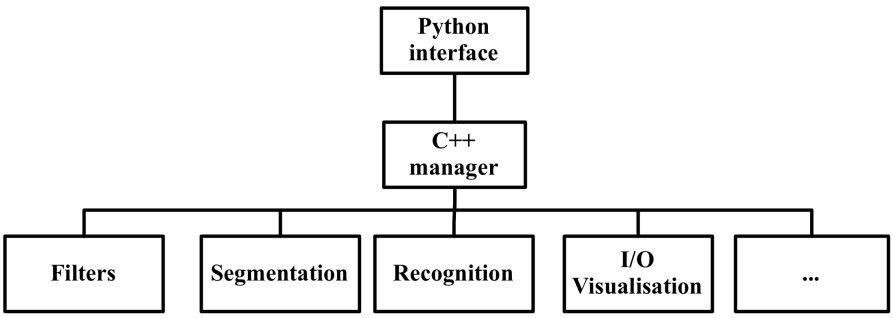

Fig. 3. Module Interaction

interface. Visual output could be generated and forwarded via a host. However, this approach becomes more complex in a distributed environment and such development overheads quickly outweigh potential return based on the current use case.

VIII. FUTURE WORK

This section will reflect the near future and discuss concepts and design choices likely to soon shape the pcl_toolkit. Of the current modules recognition is likely to become a key future element in order to generate knowledge applicable to point cloud semantisation tasks. The pcl_toolkit will continue to be developed so that it provides a platform for experimentation into the applicability and performance of various approaches. This is an essential functionality for the project as successful techniques can be finalised and then quickly become supported for deployment. This consistency aims to speed up development in this area. Offering a recognition platform from the start is a representation of the focus and future of this part of the project.

There may also be potential to gain further semantic knowledge at the filtering stage of a processing pipeline. Condition based filters could play a recognition role useful to preprocessing and so relevant validation will take place in the near future. Adding semantic information at this initial stage could benefit future processes in a developed pcl_toolkit script.

A current perceived weakness of pcl_toolkit is how an exposed dependency to the Point Cloud Library significantly increases the complexity of installation. It would be beneficial even in early iterations to improve this aspect. The pcl_toolkit is planned to be a robust black box solution that can be easily added to a system and quickly become operational with little requirement for setup. In the current environment demands are already in place for this attribute to be improved so it is likely to take a high precedent in order to simplify future usage.

The pcl_toolkit is planned to support distributed computational environments as well as a standalone single node processing case. It is therefore essential to add a cluster management facility in order to make adjusting settings easier for the operator. If each node needs to be configured manually for all settings (the case at present) pcl_toolkit becomes less useful quickly when a large scale cluster is considered. On such a platform with the current level of support offered pcl_toolkit is not suitable and so this will change in future iterations to leverage the potential benefits from larger scale architectures. The proposed methodology will find an application in the energy retrofitting domain where access to a BIM [22] for delivering a holistic energy retrofitting capability is essential. The scan-to-BIM approach will facilitate the development of an energy model which will play a pivotal role in the energy retrofitting exercise [23]. The latter will be documented in follow-on publications.

System validation will be a constant, ongoing process closely running in tandem with the iterative development approach of the platform. The current, near future goal of the pcl_toolkit is to support large scale preprocessing tasks and so dataset size will appropriately act as an initial threshold metric. To achieve this artificial inputs will be used in order to better control benchmarking results. Desirable observations from this early testing will be a proportional increase in runtime; representing only the increased workload. Optimisations in later development cycles will attempt to weaken this coupling to improve performance.

The main heuristics for high-level validation will be human work effort and perceived task complexity. These will be derived from the aforementioned Ebbw Vale and REACH projects along with involved operator testimonies. Both ventures generated large datasets that are targeted use cases for pcl_toolkit. Modelling operations will have already been completed using common to industry practices so a base line control will be known; supporting benchmarking of these two attributes. Also, this quantity of data will support progression from artificial stress testing to validating performance under actual load.

ACKNOWLEDGEMENT

The authors would like to acknowledge the financial support of the BRE (Building Research Establishment).

REFERENCES

[1] B. Okorn, X. Xiong, B. Akinci, and D. Huber, "Toward automated modeling of floor plans," in *Proceedings of the symposium on 3D data processing, visualization and transmission*, vol. 2.

[2] S. Leme and N. Zaimovi-Uzunovi, "Study of ambient light influence on laser 3d scanning," in *Proceedings of the 7th International Conference on Industrial Tools and Material Processing Technologies*, pp. 327–330.

[3] D. Huber, B. Akinci, A. A. Oliver, E. Anil, B. E. Okorn, and X. Xiong, "Methods for automatically modeling and representing as-built building information models," in *Proceedings of the NSF CMMI Research Innovation Conference*.

[4] A. Dimitrov and M. Golparvar-Fard, "Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections," vol. 28, no. 1, pp. 37–49.

[5] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and automation (ICRA), 2011 IEEE International Conference on*. IEEE, pp. 1–4.

[6] R. Hulik, M. Spanel, P. Smrz, and Z. Materna, "Continuous plane detection in point-cloud data based on 3d hough transform," vol. 25, no. 1, pp. 86–97.

[7] T. L. Cottom, "Using SWIG to bind c++ to python," vol. 5, no. 2, pp. 88–97.

[8] J. Jung, S. Hong, S. Jeong, S. Kim, H. Cho, S. Hong, and J. Heo, "Productive modeling for development of as-built BIM of existing indoor structures," vol. 42, pp. 68–77.

[9] 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," vol. 19, no. 7, pp. 829–843.

[10] C. Frhlich, M. Mettenleiter, and others, "Terrestrial laser scanningnew perspectives in 3d surveying," vol. 36, p. W2.

[11] G. Vosselman, B. G. Gorte, G. Sithole, and T. Rabbani, "Recognising structure in laser scanner point clouds," vol. 46, no. 8, pp. 33–38.

[12] N. Hichri, C. Stefani, L. De Luca, and P. Veron, "Review of the as-buit BIM approaches." Proceedings of the 3D-ARCH International Conference.

[13] M. Golparvar-Fard, F. Pea-Mora, and S. Savarese, "Automated progress monitoring using unordered daily construction photographs and IFC-based building information models," vol. 29, no. 1, p. 04014025.

[14] L. Barazzetti, "Parametric as-built model generation of complex shapes from point clouds," vol. 30, no. 3, pp. 298–311.

[15] X. Xiong, A. Adan, B. Akinci, and D. Huber, "Automatic creation of semantically rich 3d building models from laser scanner data," vol. 31, pp. 325–337.

[16] E. Valero, A. Adan, and C. Cerrada, "Automatic construction of 3d basic-semantic models of inhabited interiors using laser scanners and RFID sensors," vol. 12, no. 5, pp. 5705–5724.

[17] European Commision, "Energy-efficient buildings PPP: Multi-annual roadmap and long term strategy."

[18] M. Economidou, "Europes buildings under the microscope."

[19] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," vol. 275, pp. 314–347.

[20] J. Steel, R. Drogemuller, and B. Toth, "Model interoperability in building information modelling," vol. 11, no. 1, pp. 99–109.

[21] T. Mill, A. Alt, and R. Liias, "Combined 3d building surveying techniquesterrestrial laser scanning (TLS) and total station surveying for BIM data management purposes," vol. 19, pp. S23–S32.

[22] Y. Rezgui, T. Beach, and O. Rana, "A governance approach for BIM management across lifecycle and supply chains using mixed-modes of information delivery," vol. 19, no. 2, pp. 239–258.

[23] B. Yuce, H. Li, Y. Rezgui, I. Petri, B. Jayan, and C. Yang, "Utilizing artificial neural network to predict energy consumption and thermal comfort level: An indoor swimming pool case study," vol. 80, pp. 45–56.