



DURAARK
DURABLE
ARCHITECTURAL
KNOWLEDGE

D4.4.1 Documenting the Changing State of Built Architecture

Software prototype v1

DURAARK

FP7 – ICT – Digital Preservation
Grant agreement No.: 600908

Date: 2014-01-31
Version 1.0
Document id. : duraark/2014/D.4.4.1/v1.0



Grant agreement number	:	600908
Project acronym	:	DURAARK
Project full title	:	Durable Architectural Knowledge
Project's website	:	www.duraark.eu
Partners	:	UBO – Rheinische Friedrich-Wilhelms-Universitaet Bonn [DE]
Project instrument	:	EU FP7 Collaborative Project
Project thematic priority	:	Information and Communication Technologies (ICT) Digital Preservation
Project start date	:	2013-02-01
Project duration	:	36 months
Document number	:	duraark/2014/D.4.4.1/v1.0
Title of document	:	Documenting the Changing State of Built Architecture – Software prototype v1
Deliverable type	:	Software prototype
Contractual date of delivery	:	2014-01-31
Actual date of delivery	:	2014-01-31
Lead beneficiary	:	UBO
Author(s)	:	Sebastian Ochmann <ochmann@cs.uni-bonn.de> (UBO) Richard Vock <voek@cs.uni-bonn.de> (UBO), Raoul Wessel <wesselr@cs.uni-bonn.de> (UBO) Henrik Leander Evers <heve@kadk.dk> (CITA) Henrik Nergård <Henrik.Nergard@ltu.se> (LTU) Peter Törlind <Peter.Torlind@ltu.se> (LTU)
Responsible editor(s)	:	Sebastian Ochmann <ochmann@cs.uni-bonn.de> (UBO) Richard Vock <voek@cs.uni-bonn.de> (UBO)
Quality assessor(s)	:	René Berndt <rene.berndt@fraunhofer.at> (FhA) Stefan Dietze <dietze@L3S.de> (LUH)
Approval of this deliverable	:	Stefan Dietze <dietze@L3S.de> (LUH) – Project Coordinator Marco Fisichella <fisichella@L3S.de> (LUH) – Project manager
Distribution	:	Public
Keywords list	:	Curation, Documentation of Building State, Registration, Alignment, Difference Detection

Executive Summary

This document presents deliverable D4.4.1, the software prototype v1 for (semi-)automated registration of models which follow the Building Information Modeling (BIM) paradigm and point clouds. This deliverable is part of WP4, “Documenting the Changing State of Built Architecture”. The developed methods and tools support several curation tasks in the Long-Term Digital Preservation (LDP) framework by providing means to spatially align different digital, three-dimensional building representations. Furthermore, this alignment is a crucial prerequisite for further processing steps which will be developed in the course of WP4.

Table of Contents

1	Introduction	5
2	Problem Description and Requirements	9
2.1	Usage Scenarios	9
2.2	Relation to the DURAARK System Architecture	10
2.3	Requirements	10
3	Related Work	14
4	Workflow and Implementation	16
4.1	Workflow	16
4.2	Implementation	17
5	Evaluation of Suitable Host Applications	20
5.1	Features to evaluate	20
5.2	Evaluation of Revit	22
5.3	Evaluation of Leica Cloudworx plugin for Revit	24
5.4	Evaluation of CloudCompare	24
5.5	Evaluation of Graphisoft ArchiCAD	26
5.6	Evaluation of Nemetschek Vectorworks Architect	27
5.7	Evaluation of Nemetschek Allplan	28
5.8	Discussion	29
5.9	Conclusions	31

6	Summary	32
	Glossary	33
	References	35
A	Prototype Software Usage Manual	36
	A.1 Prerequisites and Installation	36
	A.2 First Steps and Camera Navigation	36
	A.3 Spatial Alignment	37
	A.4 Export Registration Results	38

1 Introduction

Most buildings undergo a more or less complex lifecycle during which steady change to their appearance occurs. Usually erected for a specific purpose, a building's interior and potentially its exterior are adapted as soon as this purpose changes. Retrofitting, renovation, and environmental influences are additional sources for change in structure and appearance over the years. Due to the increasing use of 3D techniques for prescriptive and descriptive building documentation, a number of *Digital Models* are created throughout the building's entire lifecycle. These digital objects may include descriptions of the complete building, but they may also reflect only a part of it which recently underwent a change. Accordingly, the digital objects are usually out of spatial alignment. Furthermore, the documentations may consist of high-level Building Information Modeling (BIM) objects but also of low-level point clouds. These two kinds of representations are out-of-sync regarding the quantitative and qualitative amount of adherent metadata. For DURAARK's *Designated Community* that comprises architects, engineers, municipalities, and building/facility managers, the documentation of change together with the synchronization of representations in different formats or obtained at different points in time constitute a fundamental and crucial aspect regarding the lifecycle of a building. One or several of these varying representations of the same building may be stored in an OAIS compliant archive.

The goal of WP4 is to develop tools for the successful spatial alignment of two representations (Task 4.1), which is commonly known as *registration*, the detection and storage of information about geometric differences (Task 4.2), and the synchronization of adherent metadata (Task 4.3). In this first WP4 deliverable we present software prototype v1 for registration (Task 4.1) given as a standalone Graphical User Interface (GUI) application¹. The need for registration in the context of DURAARK arises when two distinct measurements of the same scene (e.g. a building) are taken. In this case both measurements have their own local coordinate system and need to be aligned in order to obtain a representation in a common, global coordinate system. For the purpose of documenting the changing state of architecture this is a fundamental requirement in order to compare two representations constructed/measured at different points in time. Figure 2 shows a

¹The software prototype may be obtained from the FTP site <ftp://ftp.cg.cs.uni-bonn.de> (anonymous login, file path /pub/outgoing/duraark/Software_Prototype_v1_D441_D551.zip) which has been tested on Windows 7 (64-bit). For downloading example datasets, user credentials are required which will be provided to reviewers separately.

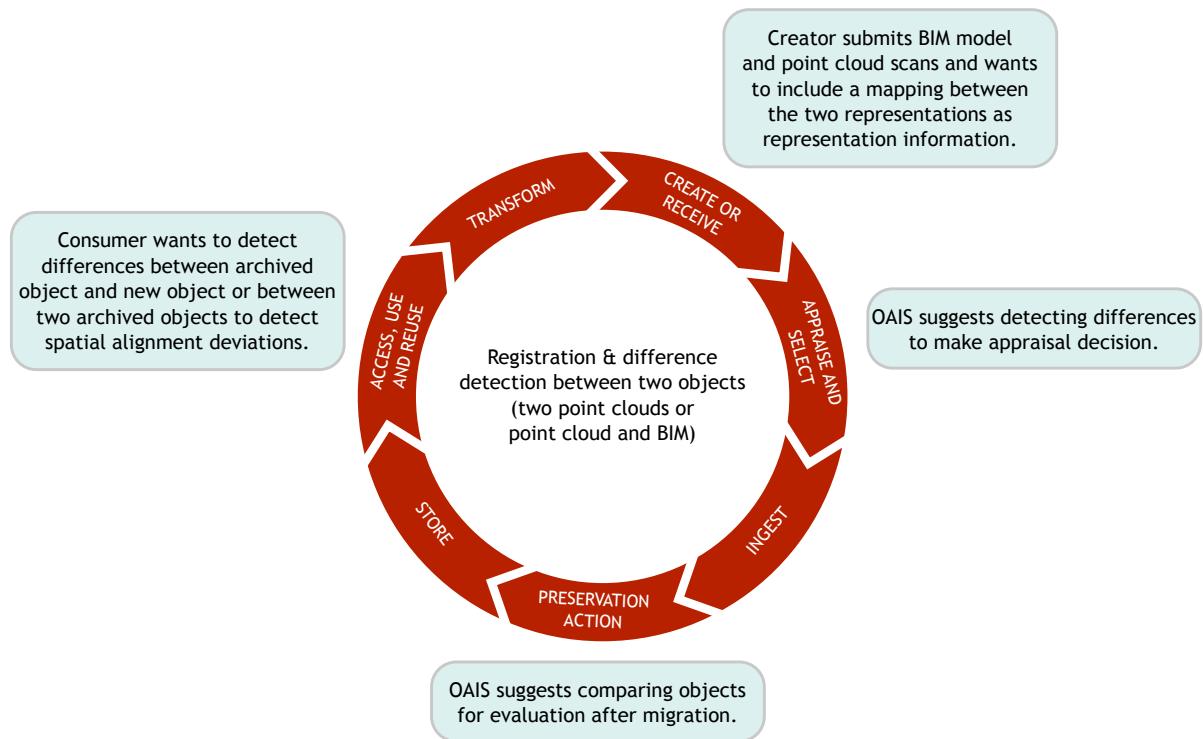


Figure 1: Usage scenarios tackled in WP4 associated with several *sequential actions* in the DCC Lifecycle Model. The Figure uses a part of the model described in [12].

typical example for the registration of two different representations of the same building. Figure 1 shows the developed methods as part of the Digital Curation Centre (DCC) Lifecycle Model [12] together with an assignment of several usage scenarios to *sequential actions* taken throughout the lifecycle of an archived digital asset. These usage scenarios as well as requirements posed to the developed software in order to fulfill these tasks will be further discussed in Section 2.

In the remainder of this deliverable, we will provide more in-depth insights to the requirements of the DURAARK registration component and discuss related work. We will then present a suitable interface and demonstrate the workflow inherent to our registration component, which was developed in close cooperation with architects to ensure that the requirements of the particular target audience are met. In this first prototype the registration component was implemented as a stand-alone program. However, as many people in DURAARK's *Designated Community* are used to working with (commercial) off-the-shelf 3D Computer-Aided Design (CAD) software, we also examined the feasibility of integrating the registration component and other software modules originating from

WP4 and WP5 into plug-ins. Finally, we will summarize our efforts, comment on the results, and give a brief outlook on the work to be conducted in WP4 throughout the second year of DURAARK.

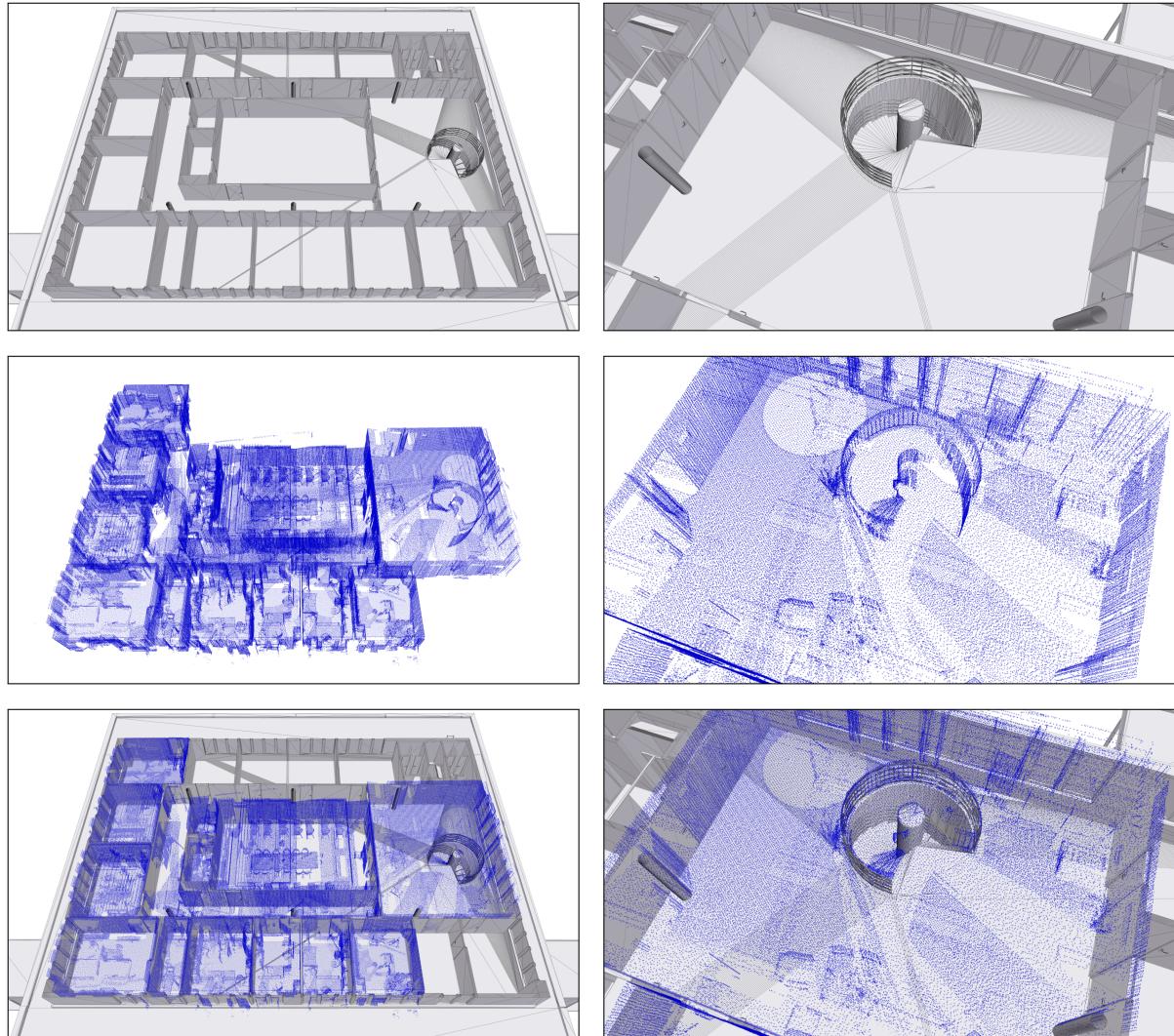


Figure 2: Registration of a BIM model and point clouds of the *Risløkka Trafikkstasjon* (Norway). Top: A mesh model generated from an Industry Foundation Classes (IFC) file of the building (overview of one storey and a detail view). The ceiling structure has been removed for visualization. Middle: Corresponding scans taken in the real building. Bottom: The two representations after registration using the developed tools.

2 Problem Description and Requirements

Archival of building representations includes the management of multiple different representations of the same building. For instance, in addition to an as-planned BIM model, one or more point cloud scans acquired at different locations within the building or at different points in time may be stored. In order to keep the different three-dimensional representations synchronized and usable for tasks like performing comparisons between them, they need to be spatially aligned in a common coordinate system. The software component developed in the course of WP4 supports this curation task. Also, the availability of registered representations is a crucial prerequisite for the tools which will be developed in upcoming WP4 deliverables for performing comparisons and transferring structure and semantics from high-level BIM models to point clouds.

2.1 Usage Scenarios

The developed software component supports the registration of a BIM model against a point cloud scan, as well as the alignment of two point clouds². These two modes of operation support the user with tasks in different usage scenarios throughout the digital preservation lifecycle (see also Figure 1).

- During the *Create or Receive* phase of the lifecycle model, a content creator who wishes to ingest multiple representations of the same building into the archive may use the developed tools to spatially align the representations in order to improve comprehensibility and enable further processing (for instance, comparing different representations of the same building).
- During the *Appraise and Select* phase, the detection and highlighting of differences between multiple (concurrent) representations of the same building or part of building may be used to guide appraisal decisions. For instance, comparing a newly acquired point cloud scan with a BIM model from the archive may help with an analysis whether the building has been scanned completely (or which parts have been scanned).
- The comparison of objects after a migration (for instance, into another data format) helps with validation of migration results implemented as a *Preservation Action* in

²The individual scans *within* a point cloud are assumed to be already registered to each other. This step is usually done by the scanner software.

the lifecycle by analyzing whether any unwanted or unexpected changes occurred due to the migration.

- An archived representation may be analyzed regarding spatial deviations with either a newly acquired dataset or another archived version in the scope of *Access, Use and Reuse*. For instance, if multiple point cloud scans taken over a long period of time are available in the archive, they may be compared in order to detect either planned changes to the building (e.g., torn down walls) or defects which may slowly impair the building's structural integrity.

In regard to the Requirement Document D2.2.1, the use cases specifically addressed in this deliverable are:

- Use Case 4 – Detect differences between as-planned and as-built state,
- Use Case 5 – Monitor the evolution of a structure over time, and
- Use Case 7 – Plan, document and verify retrofitting/energy renovation of buildings.

2.2 Relation to the DURAARK System Architecture

The software components developed in the course of WP4 and WP5 are designed to be modular software tools that can be used on the client computer at any stage when enrichment, analysis or comparison of different geometric representations of building data (either acquired by the user or fetched from the Long-Term Digital Preservation (LDP) system) is requested. In this context the most efficient and generic way to interface with the rest of the DURAARK system architecture is by means of file input/output. For an overview of the system architecture and how it relates/interfaces to the software components developed in WP4/WP5, see Figure 3.

2.3 Requirements

This section gives an overview of the requirements of the developed software component from a usability and technical perspective. Section 4 describes how these requirements were met in the first software prototype.

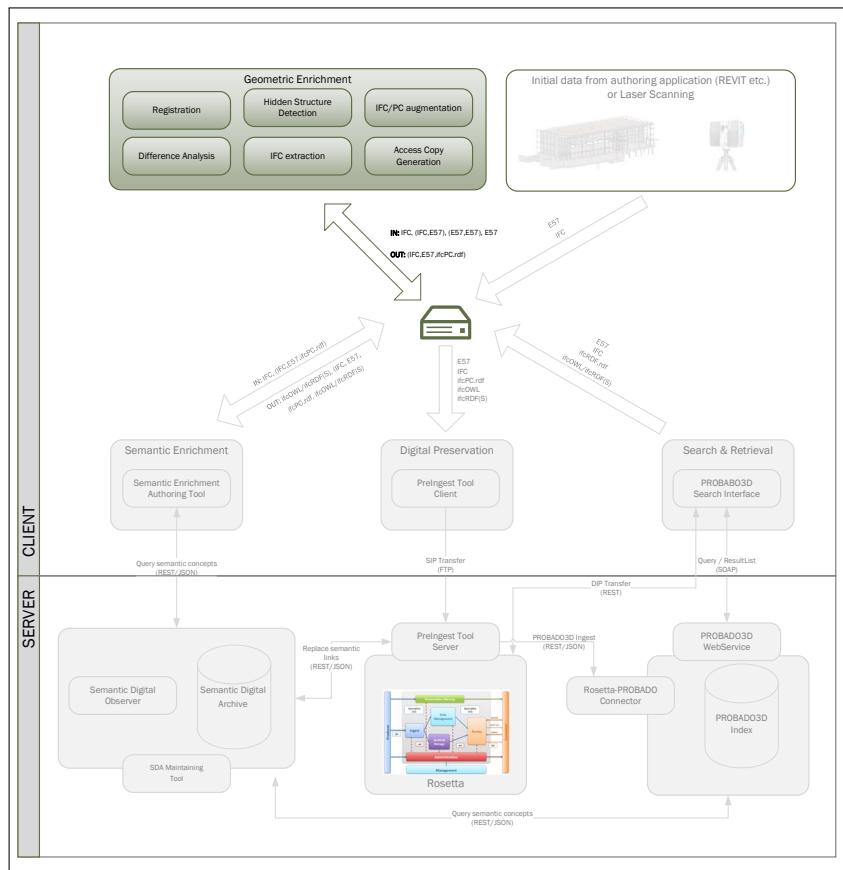


Figure 3: Overview of the current system architecture design for the DURAARK project. The Geometric Enrichment section in the upper left corner gives an overview of the WP4/WP5 software components and their interfaces/relations to other DURAARK framework components.

2.3.1 Usability Requirements

The tasks pointed out in Section 2.1 pose several requirements to the registration component. Firstly, it needs to include means to select the representations to be aligned and load them into the software component. Furthermore, the registration method used requires a coarse pre-alignment. Thus, the user needs to be able to position the representations relatively to each other in an intuitive manner which is accessible also for non-expert users. This is realized by allowing the user to mouse-drag one representation onto the other. The automatic fine alignment step shall be controllable via parameters such as the desired alignment precision in order to control the trade-off between computation time and alignment precision. For this purpose the GUI designed for the software

prototype includes interface elements commonly used for numeric data input. Finally, means for storing the computed results must be provided which is done in the prototype using conventional output file selection tools.

Although the developed interface methods are intuitive to use, the process of pre-aligning large datasets of buildings containing many rooms may be cumbersome. We are evaluating methods for assisting the user by exploiting information about building structures and topology extracted using algorithms developed in the course of WP5 (Task 5.1). By using this derived information about rooms together with suitable attributes (such as room area), suggestions for the initial alignment of the datasets may be generated in an automated manner. These methods will be further evaluated and developed for inclusion in the next version of the software prototype.

2.3.2 Technical Requirements

The first requirement is the ability to load the data formats which are used for storage of building representations. In the scope of the DURAARK project, E57 [1] was chosen as a versatile format for storing point cloud data and associated metadata such as scanner positions and the Industry Foundation Classes (IFC) standard was selected for storage of BIM models which is widely adopted by major software vendors of architectural software packages. The import of both of these formats into the prototype application is realized using suitable import libraries/functions for these formats³. In addition to parsing the higher-level entities stored in an IFC file, means for converting BIM models to mesh geometry is needed. The reason for this is that the registration method used requires explicit three-dimensional geometry to be given. However, entities in the IFC format (e.g. walls, doors) are usually specified in an implicit manner (for instance, a wall may be specified by its outline on the ground plus an elevation instead of an explicit representation in form of connected vertices in three-dimensional space). Once the data is loaded into memory, efficient means to process point cloud data and mesh geometry are required. Another requirement is that the software needs to be able to cope with the large amounts of stored data. Especially point cloud scans can easily reach sizes of tens or hundreds of gigabytes which usually makes it impossible to load the complete scan data of a building into main memory for further processing. Suitable methods for

³However in the *delivered* application which has been ported from GNU/Linux to Windows time constraints forbid porting these libraries to windows as well, so only .obj (mesh geometry) and .pcd (point clouds) files are supported in the delivered version.

overcoming this difficulty must be implemented to allow processing and viewing of the datasets. In the prototype developed, uniformly subsampled parts of the point clouds are used as a basis for visualization and computation to overcome size constraints. Lastly, the computed registration result (i.e. the alignment transformation) needs to be exported in a format which is also suitable for storage in the LDP system alongside with the data objects. This is done using a file format in one of several common Resource Description Format (RDF) formats (a concrete format depends on choices to be made in WP2 and WP3, in the prototype implementation the Turtle format was chosen).

The detection of shapes developed in WP5 (Task 5.2) may provide useful information for increasing the robustness of the registration process by filtering out elements which do not belong to the building's structure (e.g. furniture present in the scans). This will be further evaluated during development of the next version of the registration software prototype.

3 Related Work

Methods for aligning multiple representations (e.g. point clouds, mesh models) of the same object have become an important field of research with the increasing availability of scanning devices for capturing real-world objects. This section gives an overview of approaches for the alignment of two or more, possibly partial, views of the same object. Although these methods are often used for the alignment of point clouds, most of them are also applicable to different kinds of object representations, e.g. mesh geometry.

One of the most popular method among registration algorithms is the Iterative Closest Point (ICP) as proposed by Besl and McKay [6]. The basic approach works as follows. Corresponding points in two given representations of the same object are determined (in the original version, nearest pairs of points are selected) and a rigid transformation is computed which aligns the point sets optimally in a least squares sense. The computed transformation is applied to one of the point sets and the process is iterated until convergence. The approach possesses many advantages which made the basic approach popular and widely-used. For instance, it is applicable independently of the kind of shape representations used, it has a simple structure, and a reasonably fast convergence behaviour. The main disadvantages are its high sensitivity to outliers in the data, the fact that the basic method is restricted to rigid transformations, and that it requires a sufficiently good initial alignment.

Many variations and extensions have been proposed which make the ICP approach one of the most widely-used to this day; Rusinkiewicz and Levoy [17] give an overview of criteria which distinguish ICP variants and evaluates the performance of several variants. Gelfand et al. [10] propose criteria for selecting specific point subsets for usage in the ICP algorithm which help to constrain the computed transformations, improving the geometric stability of the basic approach. The work of Chetverikov et al. [9] aims at improving the robustness of the ICP approach in regard to outliers by restricting the distance of points which are considered correspondences to a certain limit. This way, outliers in the data are mostly ignored when estimating the transformations during the alignment process. Zinßer et al. [20] present a variant called “Picky ICP” which combines several extensions to the basic algorithm regarding point selection, outlier elimination, motion parameter extrapolation and a new stopping criterion. Other variants propose alternatives for finding correspondences in the datasets such as point-to-line [8], point-to-plane [19] or plane-to-plane [18] approaches.

In regard to the initial pre-alignment which is necessary in the basic ICP approach, several proposals have been made to automatically estimate an initial alignment. Gelfand et al. [11] uses local feature descriptors detected on the surface of the objects to be aligned and uses the information about corresponding feature descriptors for computing an initial rigid alignment. After that, ICP may be used for fine alignment. In the work of Pandey et al. [15], information about co-registered LIDAR and camera imagery is exploited in order to compute an initial alignment by using image descriptors. Huber et al. [13] estimate an initial alignment between multiple views of the same object by converting the input laser scans to mesh surfaces and performing a two-phase registration on these representations. First, the generated surfaces are registered pair-wise using a surface matching system. Subsequently, a global registration is performed by collecting the pair-wise registration of overlapping surfaces in a graph encoding the connectivity between the individual views and a search for a connected sub-graph containing only correct matches is performed. The approach developed by Makadia et al. [14] computes a coarse alignment between multiple scans by correlating orientation histograms created from normal fields before applying ICP for fine alignment. In addition to scans of small objects, this method was also evaluated on indoor scans of a room.

Pottmann et al. [16] pursue an alternative approach to ICP. They propose the usage of “instantaneous kinematics” using approximated squared distance functions of object surfaces. The authors argue that ICP does not converge well in the vicinity of the object surface which is a problem the proposed method strives to overcome. A software for registration specifically designed for terrestrial laser scanning for architectural and archaeological purposes has been presented by Bornaz et al. [7]. It proposes the usage of reflective targets which are strategically placed at the scanning site such that their locations may be detected in the scans and used for determining transformations between the individual views. While similar methods have been adopted by popular software packages (e.g. FARO SCENE) it has the drawback that it requires the presence of specifically placed target objects during the scanning phase.

4 Workflow and Implementation

This section describes the developed software component from a user's perspective as well as implementation details. In particular, it describes how the requirements stated in Section 2 were met in the first software prototype. For detailed instructions on how the software is used the user is referred to the manual section in Appendix A.

4.1 Workflow

Because the implemented registration method is semi-automatic in nature and requires a manual, coarse pre-alignment of the building representations, and because the registration results shall be visually verifiable, a graphical user interface has been developed in close cooperation with architects to support the user throughout the alignment process.

The workflow starts with the selection of two representations of the same building which may be either a BIM model and a point cloud, or two point clouds. After the representations have been loaded, they are visualized in an interactive, three-dimensional viewport. Figure 4 shows an example for a BIM model and a point cloud.

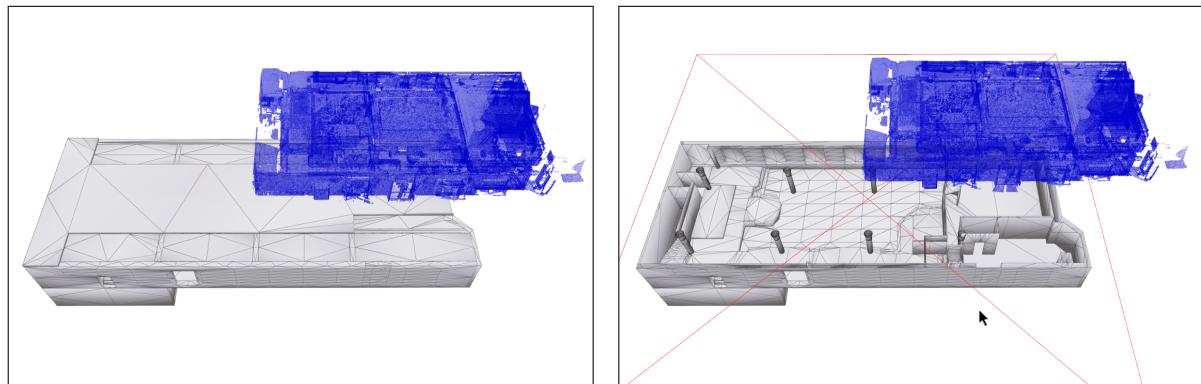


Figure 4: Left: A BIM model and a point cloud have been loaded. Right: In our stand-alone implementation, the mesh geometry may be clipped interactively to get a better view of the building's interior.

The user may then reposition and rotate either of the representations in order to provide the registration algorithm with the required initial, coarse pre-alignment. This repositioning is done by dragging the visual representations of the building in a similar manner as in most 3D modeling software packages, see Figure 5.

After the initial alignment is given, the user has the opportunity to specify parameters

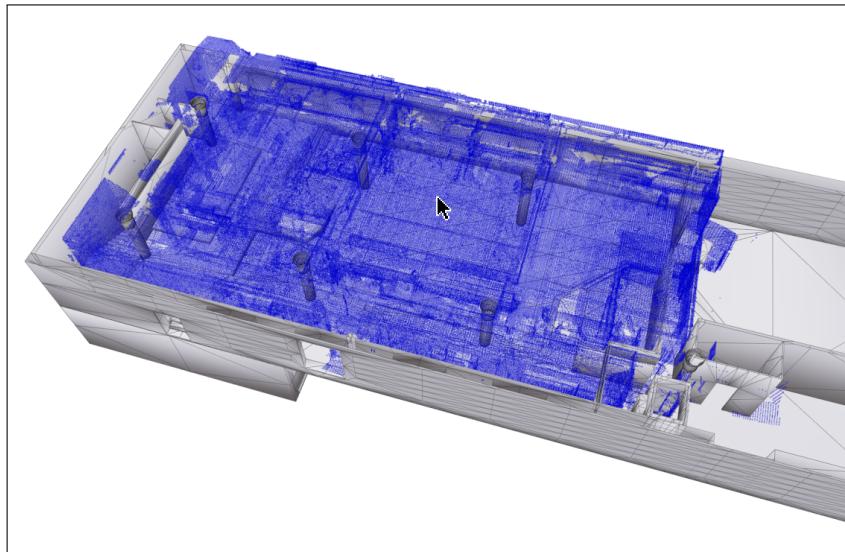


Figure 5: The representations after coarse user alignment through dragging.

which control the subsequent automatic fine alignment process. These parameters include a desired, approximate target precision of the alignment (which controls the trade-off between computation time and precision) and the number of sample points which are used (a higher number of points may increase precision at the cost of higher computational costs).

Subsequently, the fine alignment process is initiated and its result (i.e. the improved alignment) is displayed in the viewport for verification. If necessary or desired, any of the previous steps may be repeated to further tweak the registration results. Figure 6 shows a close-up of a part of the building before and after fine alignment. Finally, the computed transformations for the models may be exported for storage as *representation information* accompanying the datasets. Figure 7 shows a diagram of the workflow of the developed software component.

4.2 Implementation

For the automatic fine alignment between the different building representations, a variant of ICP is used. The two major modifications to the basic algorithm are

- sampling of a subset of points used for registration, and
- capping the maximum distance between corresponding points to a given threshold.

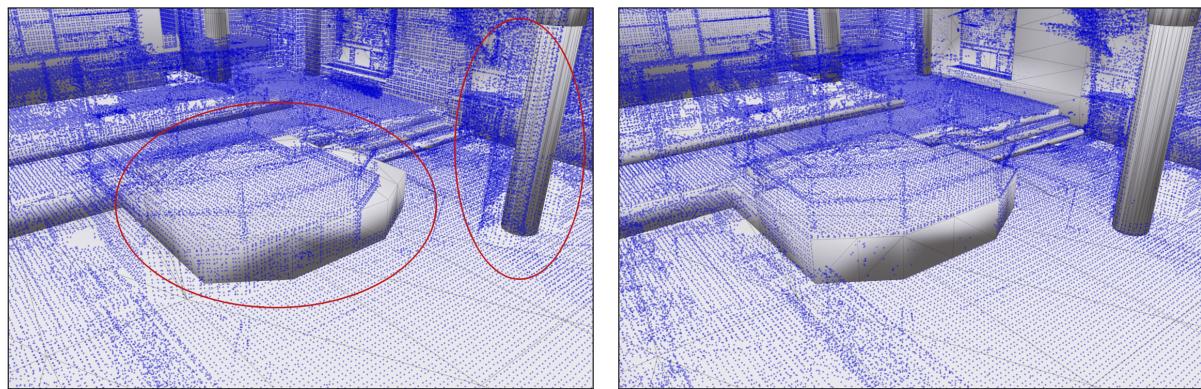


Figure 6: Left: A close-up showing the representations after manual pre-alignment. The alignment is quite coarse as can be seen for the platform and the column in the marked regions. Right: The same view after automatic fine registration.

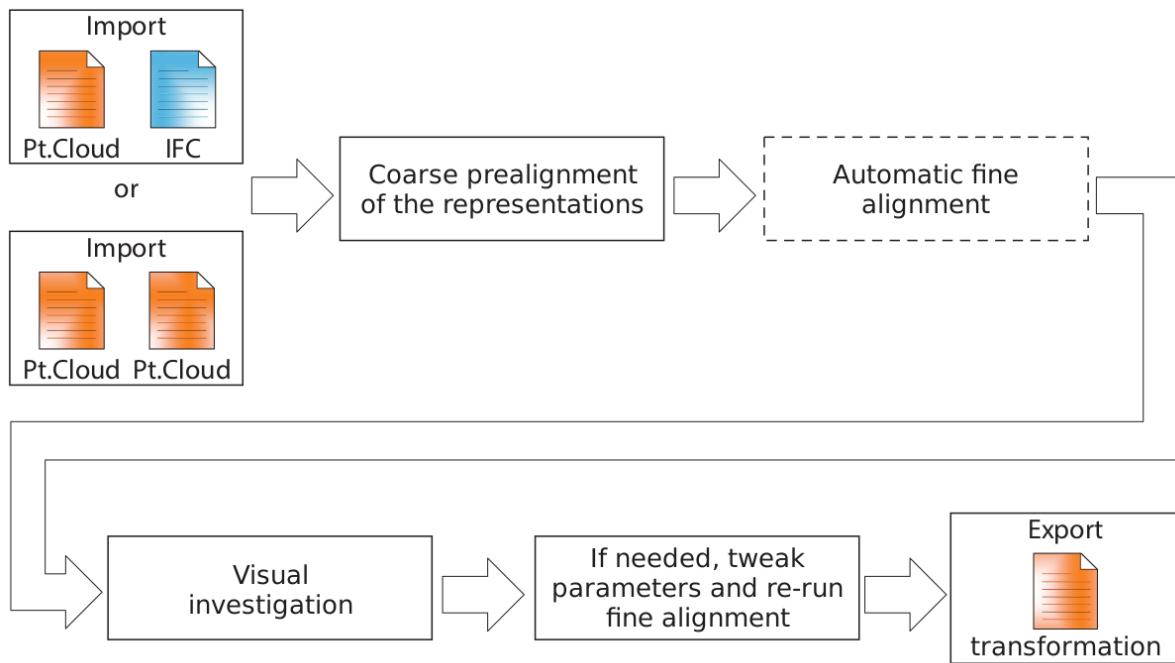


Figure 7: Flow diagram showing the steps of the registration workflow.

The first modification is crucial for the feasibility of the approach when handling *large datasets* which is often the case when dealing with point cloud scans. The representations are subsampled prior to performing the registration such that a suitable trade-off between computation time and precision is achieved. The second modification addresses robustness problems of the basic ICP approach in the presence of only *small overlaps* between

the representations (see [9]). For instance, point cloud scans seldom cover all parts of a building that are included in a corresponding BIM model. Likewise, two different point clouds may include different parts of a building such that only a small overlapping part may be used for registration.

The ICP approach requires information about corresponding point pairs in two given (subsampled) representations A and B . In case of two point clouds, this correspondence is obtained by searching for nearest points in representation B from all points in representation A (or vice-versa). In case of registration against a BIM model, the surfaces of the model are first sampled point-wise in order to generate a point cloud representation internally. Subsequently, the method for aligning two point clouds is applied.

In regard to the file formats used (E57, IFC), open-source libraries have been chosen to parse and access the stored data. For E57, the libE57 library [3] has been selected. It provides programming interfaces to access the data part of an E57 file (e.g. point coordinates, images) and associated metadata (e.g. number of scans, scanner positions). BIM data in the IFC format is accessed using the IFCOpenShell library and toolset [2]. In addition to making the high-level entities stored in an IFC file available through suitable programming interfaces, this library allows the generation of mesh geometry from the BIM model which is required for registration against a point cloud (as described above). For internal processing of mesh and point cloud data (i.e. performing operations on the data once it is loaded into memory), further software libraries are used. For mesh processing, the OpenMesh [4] library is used which provides efficient data structures for working with mesh data as well as algorithms for performing several operations on meshes. For working with point cloud data, the Point Cloud Library (PCL) [5] is used which provides data structures and algorithms for manipulating point clouds.

The computed transformations are stored in a format which follows the RDF family of specifications. This standard has been chosen as it provides universal and versatile means for attaching information to datasets and interlinking them. Please see document D2.2.3 for further information and a concrete example.

5 Evaluation of Suitable Host Applications

The first software prototype for registration is implemented as a stand-alone application using a framework which was developed internally for testing and demonstration purposes. Although the current version is already useable and functional, it is planned to also provide the components developed in the course of WP4 and WP5 as plugins for software packages widely-used by architects. This section provides an evaluation of possible host applications in order to choose a suitable target software package for which a plugin will be implemented.

There exist several commercial programs that support BIM data. Well-known software with a big user group within AEC (architecture, engineering and construction) industry includes Autodesk Revit, Graphisoft ArchiCAD, Nemetschek Allplan, and Vectorworks. These software solutions include features required within the project (working with BIM and point cloud data, support for the E57 format for point clouds and IFC formats for BIM data).

The buildingSMART⁴ organization has been assigned to certify different software solutions for IFC read/write compatibility. According to the tables available on the buildingSMART website⁵ a number of different applications have been certified for IFC read/write compatibility following the IFC2x3 CV v2.0 standard. At this point in time only a few products (Graphisoft Archicad⁶, Nemetschek Scia Engineer⁷, Vectorworks⁸, and Tekla Structures⁹) have been certified for both read and write compatibility. Most have been certified for “Export/Write”-mode and are currently under review for IFC import/read-mode.

5.1 Features to evaluate

Within the DURAARK project certain features that a host application would need to provide in order to be suitable have been discussed and defined.

⁴<http://www.buildingsmart.org/>

⁵<http://www.buildingsmart-tech.org/certification/ifc-certification-2.0/ifc2x3-cv-v2.0-certification/participants>

⁶<http://www.graphisoft.com/archicad/>

⁷<http://nemetschek-scia.com/en/software/product-selection/scia-engineer>

⁸<http://www.vectorworks.net/>

⁹<http://www.tekla.com/products/tekla-structures>

The requirements to the software for usage in the project up until now includes:

1. Application Programming Interface (API) functionality
 - (a) The host applications need to give the user access to software functionality through a suitable API. This is needed since developed software modules need integration with the host application as a plug-in.
 - (b) Means for integrating new GUI elements to the native GUI
2. Support for IFC
 - (a) Import/export capabilities for the IFC2x3 CV v2.0 standard¹⁰
 - (b) Extraction of geometric data from loaded BIM models (some operations such as the registration of BIM models and point clouds require an explicit geometric representation of the model but entities in BIM models may be, at least partly, defined implicitly in the IFC format)
3. Support for working with point clouds
 - (a) Import/Export functionality
 - (b) Support for the E57 file format¹¹
 - (c) Coarse (manual) pre-alignment of point clouds
 - (d) Means to remove or add points from/to point clouds or individual scans
 - (e) Extraction of point cloud data from loaded point clouds
 - i. Point coordinates
 - ii. Scanner positions
 - iii. Information about multiple scans
 - (f) Provide means to change coordinate base and rescale functionality (for instance, mm ↔ inch, mm ↔ m)
 - (g) Coloring functionality, highlighting parts of a point cloud
 - (h) Efficient rendering of point clouds
4. Transforming models and point clouds (translation, rotation of some or all parts)
5. Functions for creating new geometry

¹⁰<http://www.buildingsmart-tech.org/specifications>

¹¹<http://www.astm.org/Standards/E2807.htm>

5.2 Evaluation of Revit

Autodesk Revit is a CAD software developed for Building Information Models that has some unique features such as parametric components, bi-directional associativity, work-sharing, and construction modelling¹².

5.2.1 API

Autodesk Revit has a .NET API which means any of the .NET compliant programming languages (C#, VB.NET, F#, etc.) may be used to develop a plug-in¹³. Autodesk Revit 2012 and higher supports the use of .NET Framework 4.0.

The point cloud API

The point cloud API in Revit provides two ways to work with point clouds.

1. The Client API allows to create new point cloud instances, read and filter points, select subsets of points, and select points to be highlighted or isolated.
2. The Engine API allows the design and integration of new point cloud engines and processing of unsupported file formats by directly providing point cloud data to Revit.

The point cloud engine provides means to render point clouds efficiently by subsampling the visualized point set when necessary. Using a custom point cloud engine also enables the development of features which are otherwise not supported by the default engine provided by Revit.

5.2.2 Support for IFC

Revit can export IFC using two different exporters:

1. the official one included in Revit
2. an open source exporter (<http://sourceforge.net/projects/ifcexporter/>)

¹²<http://www.autodesk.com/products/autodesk-revit-family/overview>

¹³<http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=16777469>

The internal one is currently certified for export of IFC; import is in the progress of certification. Autodesk Revit has support for loading and modifying IFC mapping files so that internal Revit Containers represent a specific IFC container. Also, Revit has functionality to further edit and work with the imported files.

5.2.3 Working with point clouds

Revit has a limited functionality to import point clouds built-in. The imported point clouds are converted to an internal working format during the import process.

A scan including several scan positions is converted to one .rcp file and several .rcs files (one for each scan position) when imported. The imported point cloud representation is selectable in Revit only as a single point cloud, however the separate scan positions can be highlighted in different colors.

Subdivision or sectioning of point clouds needs to be done before entering Revit (for instance in Autodesk ReCAP or FARO Scene), if such functionality is desired. Separating walls, floors, doors, windows, stairs etc. from the initial point cloud into several regions can be done in ReCap or Scene. The separate regions can then be exported into separate point cloud files for import into Revit.

Each imported point cloud is represented as an instance in Revit. There is quite limited functionality to interact with parts of the point clouds. There is no possibility to filter, hide or select parts of the point cloud. Advanced manipulation has to be done using the PointCloud API. There is some possibility to snap to features in the point clouds where Revit will detect planes that are perpendicular to the current work plane and very close to the cursor.

Alignment, translation and scaling of point clouds

When importing the point clouds, the user may choose different positioning options. Manual movement and rotation of point clouds or other instances can be done within the software (selecting object and using the Move command). Scaling and changing between units is possible.

Grouping, changing colour

The different point clouds imported into Revit may be differentiated from each other by overriding the imported RGB-visibility settings.

5.2.4 Working with objects and geometry

Revit has good functionality to work with and edit different types of objects. New geometry may be conducted in many ways using built-in Revit family objects as well as creating customized objects.

5.3 Evaluation of Leica Cloudworx plugin for Revit

Leica Cloudworx¹⁴ is a point cloud processing plugin for Revit (as well as other software) created by Leica Geosystems to increase functionality when working with point clouds. It is mentioned within this report since it might be an alternative to Autodesk ReCAP that has rather limited functionality for editing point clouds. Cloudworx has direct import functionality of projects created in e.g. Cyclone¹⁵ as well as point clouds from other applications. When importing, the user has the ability to select user defined coordinate systems set in other software (Cyclone). One can also choose to set distance units.

Several clipping modes that are made in the Cloudworx software can be managed and modified (name, size, location, thickness etc.) via a clipping manager. The tools available make working with the point clouds much easier than exporting them to an external point cloud editor since you only work within Autodesk Revit.

5.4 Evaluation of CloudCompare

CloudCompare¹⁶ is a 3D point cloud (and triangular mesh) processing software used for 3D deviation analysis and point cloud registration. It provides means for matching and comparing multiple point clouds as well as point clouds and mesh models. It is an open source project and totally free for use. It is mentioned here since it can be used to edit the point clouds if needed before importing them into BIM software. It includes features/algorithms¹⁷ such as registration, projections, distance computations, statistics computations, segmentation, and estimation of geometric features (density, curvature, roughness, etc.).

¹⁴http://hds.leica-geosystems.com/en/Leica-CloudWorx_60696.htm

¹⁵http://hds.leica-geosystems.com/en/Leica-Cyclone_6515.htm

¹⁶<http://www.danielgm.net/cc/>

¹⁷<http://en.wikipedia.org/wiki/CloudCompare>

5.4.1 API functionality

Since it is open source there are means to develop own code as well as to create plugins to extend the current software. It runs on several platforms (Windows, Linux and Mac OS). Two kinds of plugins are available:

- Standard plugins for algorithms from external libraries (PCL¹⁸) or hardware devices
- OpenGL plugins for alternative shaders

5.4.2 Support for IFC

CloudCompare is not a BIM-style CAD software and as such it does not have functionality to open and edit BIM and IFC classes which makes it unsuitable for direct usage in the scope of this project.

5.4.3 Support for working with point clouds

CloudCompare supports opening and saving point clouds in the E57 file format as well as other formats¹⁹.

CloudCompare has both automatic alignment functionalities (best fit) as well as manual operation. Manual operation includes translation and rotation with an input device as well as changing values within a transformation matrix.

CloudCompare has the functionality to edit and manipulate the point clouds. It is possible to select and cut out subsets of points; this function is called segmentation. The user can also fuse points back into the scan or move point subsets within the model. This feature enables exporting datasets either as a single or multiple point clouds. It has not been investigated if it is possible to add single points in the cloud. CloudCompare supports switching between scalar and RGB colours of the point cloud. Using OpenGL plugins, it is possible to modify the rendering behaviour.

No information has been found if it is possible to extract point coordinates or scanner positions. It is possible to add multiple scans, however no information has been found about what information can be viewed from a scan except for the actual points. Also there is no means to switch between metric or inch units.

¹⁸<http://www.pointclouds.org/>

¹⁹<http://en.wikipedia.org/wiki/CloudCompare>

5.5 Evaluation of Graphisoft ArchiCAD

Graphisoft ArchiCAD²⁰ is a BIM CAD software for architecture. It supports 2D and 3D drafting, visualisation and other advanced features.

5.5.1 API functionality

Graphisoft ArchiCAD includes support for Geometric Description Language (GDL) to create new components. The user can gain access to API and ODBC database connections to further extend the software. Graphisoft runs an Independent Developer Program, which enables access to different development tools. Developers need to apply for user accounts to get access to developer tools²¹. Adding customized GUI and functionality is possible.

5.5.2 Support for IFC

ArchiCAD is certified for the IFC2x3 CV V2.0 standard for both importing and exporting IFC classes²². There is an extensive support for different file formats for both import and export²³.

5.5.3 Working with point clouds

ArchiCAD has little support for working with point clouds as standalone software. However there exist plugins (PointCab²⁴, Cigraph ArchiTools²⁵, Cadimage Pointclouds²⁶, and FlexiJet²⁷) that enable point cloud features. Cadimage Pointclouds enables import of a point cloud in XYZ format into the ArchiCAD model and also seems to be the most up-to date software regarding point clouds for ArchiCAD. It needs to be mentioned that this software has not been tested in practice in the course of this evaluation.

ArchiCAD does not currently support E57 import/export.

²⁰<http://www.graphisoft.com/archicad/>

²¹<http://www.graphisoft.com/support/developer/downloads/>

²²http://www.graphisoft.com/archicad/archicad_17/faq/

²³<http://www.archicadwiki.com/FileFormats>

²⁴<http://www.pointcab-software.com/en/>

²⁵<http://www.cigraph.it/en/node/422>

²⁶<http://www.cadimage.com/pointclouds>

²⁷<http://www.flexijet.info/en/>

Other features for working with point clouds such as pre-alignment, getting and setting a transformation matrix and extracting information from point clouds have not been investigated in ArchiCAD since it lacks an internal point cloud engine.

5.5.4 Working with objects

ArchiCAD has extensive features that enable the user to create different objects, geometry and complex shapes in various design stages.

5.6 Evaluation of Nemetschek Vectorworks Architect

Vectorworks Architect²⁸ is a BIM CAD software for architecture. It runs on windows and Macintosh computers. It supports 2D and 3D drafting, visualisation and other advanced features.

5.6.1 API functionality

Vectorworks Architect includes built-in features for customizing the application. Three different ways exist²⁹: VectorScript, Vectorworks Software Development Kit (SDK) and Python³⁰. VectorScript is a scripting language integrated into the core of Vectorworks. The Vectorworks SDK gives the user the ability to develop plugins in C and C++. Since Vectorworks 2014, the user may also choose to use Python as a scripting language.

5.6.2 Support for IFC

Vectorworks Architect is certified according to the IFC2x3 CV V2.0 (IFC Coordination View 2.0 Import and Export certification) standard for both importing and exporting IFC files^{31,32}. There is support for different file formats for both import and export^{33,34}.

²⁸<http://www.vectorworks.net/architect/>

²⁹<http://www.vectorworks.net/support/custom/index.php>

³⁰<http://www.python.org/>

³¹http://www.graphisoft.com/archicad/archicad_17/faq/

³²<http://planet.vectorworks.net/2013/11/vectorworks-2014-achieves-ifc-coordination-view-2-0-import-certification/>

³³<http://kbase.vectorworks.net/questions/567/Importing+Files>

³⁴<http://kbase.vectorworks.net/questions/568/Exporting+Files>

5.6.3 Working with point clouds

Vectorworks Architect has little support for working with point clouds built-in. However there exist plugins, e.g. PointCab³⁵, that enable point cloud features.

Vectorworks does not currently support E57 import/export.

Other features for working with point clouds such as pre-alignment, getting and setting of a transformation matrix and extraction of information from point clouds have not been investigated in Vectorworks Architect since it lacks an internal point cloud engine.

5.7 Evaluation of Nemetschek Allplan

Nemetschek Allplan³⁶ is a BIM CAD software that targets architects, engineers, cost management, facility management and software+service. The user can design in both 2D and 3D. The user can also conduct structural analysis on construction parts.

5.7.1 API functionality

No information has been found whether there is a way to access Allplan functions via an API or SDK. However, Nemetschek also has a cloud-based service for managing BIM information called BIM+³⁷ which is IFC compliant and includes an open API for development of applications both on PC and Mobile devices. It has not been evaluated if any editing can be done to the models within this environment or if it can only be used for sharing, viewing and commenting the files uploaded to the cloud service. BIM+ aims to support Open BIM and thus also software by other vendors. Also Vectorworks, which is a part of the Nemetschek group, has support for API and SDK.

5.7.2 Support for IFC

Allplan is a member of the BuildSmart foundation and is certified for IFC export. Import certification is currently in progress. Allplan uses a distributed way of working with files which limits the smart functionalities between applications³⁸. As of Allplan 2011 a number of file formats can be imported and exported³⁹.

³⁵<http://www.pointcab-software.com/en/>

³⁶<http://www.nemetschek-allplan.eu/>

³⁷<https://www.bimplus.net/>

³⁸<http://www.aecbytes.com/review/2013/Allplan2013.html>

³⁹http://www.nemetschek-allplan.ch/fileadmin/user_upload/switzerland-german/newsletter/02_2011/DataExchangeFormate.pdf

Exporting the entire BIM model in IFC format is possible⁴⁰. Geometric data can be extracted from the BIM models.

5.7.3 Support for working with point clouds

Allplan does not have built-in support to open files in the E57 format. However, there exist plugins that are very tightly integrated with the workflow of Allplan that might be used for this purpose. An example of this is Scalypso⁴¹ which is a software suite for analysis and reconstruction tasks using terrestrial 3D laser scanning data⁴². It supports real-time interfaces with Allplan⁴³. Scalypso also has support for transferring 3D elements to other CAD applications (AutoCAD, Microstation, and ZWCAD+). Among other formats, Scalypso supports import and export of E57 files.

The software supports multiple scans that can be aligned with each other. It is possible to export selected areas of a point cloud. Furthermore, it is possible to change colors of point clouds. However, whether selected parts of the point cloud can be highlighted needs further investigation.

5.7.4 Working with objects and creating new geometry

Allplan has advanced features for creating new geometry such as 3D solids and surface modelling. It also includes functionality to create parametric objects called SmartPart⁴⁴ that can be designed both in 2D and 3D as well to give them little bit of “intelligence” on how they can behave in the work environment. Allplan does not support full associativity between objects if the model is edited; this requires manual work to fix. An example of this is e.g. if a wall is moved the connecting walls does not “follow” the change but has to be manually edited into position.

5.8 Discussion

Within the DURAARK project there are several demands on file compatibility as well as manual/automatic operation to adjust and synchronize different architectural objects with point clouds.

⁴⁰<http://www.aecbytes.com/review/2013/Allplan2013.html>

⁴¹<http://www.scalypso.com/>

⁴²http://www.scalypso.com/en_ueber-scalypso-3d-software.phtml

⁴³<http://www.scalypso.com/scalypso-modeler-2.phtml>

⁴⁴<http://www.aecbytes.com/review/2013/Allplan2013.html>

5.8.1 IFC support

Since the standpoint has been that the IFC file format should be supported and that usage of BIM paradigms is crucial for the building and architect industry, support of the IFC standard is mandatory. Building Information Models contain various concepts that enhance the amount of information surrounding a building. Many applications support certain areas within the BIM area but very few (if any) support *all* of them. Therefore things that are important within this project have been stated in section 5.1. A crucial feature is support for import and export of BIM models. Many software packages, whereas the biggest of them already have both support and import functionality, aim to adapt this standard. All BIM software packages that have been mentioned in this report support that (with the exception of CloudCompare which does not aim to be a complete BIM software package). Also, all tools are good editors for creation of new geometry and graphical representations (rendering, fly-through etc.), which makes them all suitable for this project.

5.8.2 Point cloud support

Other important features for this project are import and editing capabilities for point clouds. This is where it becomes a bit trickier. Many of the BIM software applications do not support import and editing of point clouds internally but rely on plug-ins or add-ons to view and register the point clouds prior to import. The only software that has this feature built-in is Autodesk Revit. Revit also supports import of the E57 format (via the included ReCap). Although the functionality within ReCAP and Revit to edit the point clouds is rather limited, the possibility to develop own point cloud engines allows incorporation of custom functionality. In all plugins for point clouds there exists functionality to import multiple scans that can be integrated into one large point cloud. Full support for automatic alignment is not fully investigated. Access to the exact location of the scanner position or single points is not known for all software packages. It can be obtained in Revit as a read-only feature. To summarize, the point cloud support for almost all applications relies on third party plugins and only Revit supports E57 out of the box.

5.8.3 API support

The third thing that is important for the project is the means to customize the software to work for the project. This means access to an API and/or SDK or other means to program plug-ins, macros or applications that enhance functionality in manual or batch mode. Autodesk Revit, ArchiCAD and Vectorworks have support for API and SDK to customize or enhance the software.

5.9 Conclusions

To conclude, as far as has been investigated within this report only one BIM software tool supports most of the required features directly, namely Autodesk Revit. It supports point clouds, the E57 format, it is compatible with IFC, and offers an API/SDK. It is therefore concluded that Autodesk Revit is the best (out of the box) solution to use within this project at this point in time.

If additional point cloud functionality is needed that is not available in Autodesk Revit and ReCap, custom applications may be developed and used in CloudCompare before transferring the point cloud to Autodesk Revit.

It also has to be mentioned that the BIM+ cloud service and bimOS has to be further investigated for suitability as a viewer within this project since it has a fully open API and supports IFC from the start. Point cloud functionality may be added via CloudCompare as a plugin/app/software extension towards BIM+.

6 Summary

The first registration software prototype provides means to spatially align two different representations of the same building (Task 4.1). Using a graphical interface, the user is supported with this task in an intuitive manner. The result of the registration process helps with synchronizing multiple representations present in the long-term archival system and is an important prerequisite for upcoming tasks in WP4 such as comparison of representations (Task 4.2) and transfer of structure and semantics (Task 4.3). The resultant transformations may be stored alongside the datasets in the long-term archive for later use as a comparison mechanism for different states of a building.

For the next version of the software prototype, a plugin variant of the registration component for usage with popular software packages is planned which will increase usability for users with an architectural background. Furthermore, methods for assisting the user during the pre-alignment step will be developed using methods for the extraction of building structures and room topology from WP5 (Task 5.1). Methods for the detection of shapes developed in WP5 (Task 5.2) may help to increase robustness of the fine registration process by filtering out elements which do not belong to the building's structure.

Glossary

BIM Building Information Modeling

DCC Digital Curation Centre

CAD Computer-Aided Design

IFC Industry Foundation Classes

LDP Long-Term Digital Preservation

API Application Programming Interface

SDK Software Development Kit

RDF Resource Description Format

GUI Graphical User Interface

References

- [1] *E57 standard.* <http://www.astm.org/Standards/E2807.htm>.
- [2] *IfcOpenShell.* <http://ifcopenshell.org/>.
- [3] *libE57.* <http://www.libe57.org/>.
- [4] *OpenMesh.* <http://www.openmesh.org/>.
- [5] *Point Cloud Library (PCL).* <http://www.pointclouds.org/>.
- [6] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.
- [7] L. Bornaz, A. Lingua, and F. Rinaudo. A new software for the automatic registration of 3d digital models acquired using laser scanner devices. In *CIPA WG 6 International Workshop Scanning for Cultural Heritage Recording, Corfu, September*, pages 1–2, 2002.
- [8] A. Censi. An icp variant using a point-to-line metric. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 19–25. IEEE, 2008.
- [9] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 545–548. IEEE, 2002.
- [10] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy. Geometrically stable sampling for the icp algorithm. In *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, pages 260–267. IEEE, 2003.

- [11] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Proceedings of the Third Eurographics Symposium on Geometry Processing*, SGP '05, Aire-la-Ville, Switzerland, 2005. Eurographics Association.
- [12] S. Higgins. The DCC Curation Lifecycle Model. *International Journal of Digital Curation*, 3(1):134–140, 2008.
- [13] D. F. Huber and M. Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21(7):637–650, 2003.
- [14] A. Makadia, A. Patterson, and K. Daniilidis. Fully automatic registration of 3d point clouds. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1297–1304. IEEE, 2006.
- [15] G. Pandey, J. McBride, S. Savarese, and R. M. Eustice. Visually bootstrapped generalized icp. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2660–2667. IEEE, 2011.
- [16] H. Pottmann, S. Leopoldseder, and M. Hofer. Registration without icp. *Computer Vision and Image Understanding*, 95(1):54–71, 2004.
- [17] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.
- [18] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Robotics: Science and Systems*, volume 2, page 4, 2009.
- [19] C. Yang and G. Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.
- [20] T. Zinßer, J. Schmidt, and H. Niemann. A refined icp algorithm for robust 3-d correspondence estimation. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–695. IEEE, 2003.

A Prototype Software Usage Manual

A.1 Prerequisites and Installation

Before using the prototype software it is recommended to update the graphics card driver(s) to the newest stable version. Apart from this there are no steps necessary to use the software - it is sufficient to simply copy the software folder to a local folder and execute **Registration.bat** with user privileges. This automatically starts the modular visualization software in a setting tailored to the task of spatial object representation alignment.

A.2 First Steps and Camera Navigation

Once started, the prototype software will greet the user with an initial settings dialog (see example screenshot in Figure 8). The user must choose two files as dataset representations to align for the software - supported formats are *.obj* and *.pcd* files⁴⁵; example files are included in the software prototype package. With two datasets chosen, clicking **Start** will launch the actual registration interface.

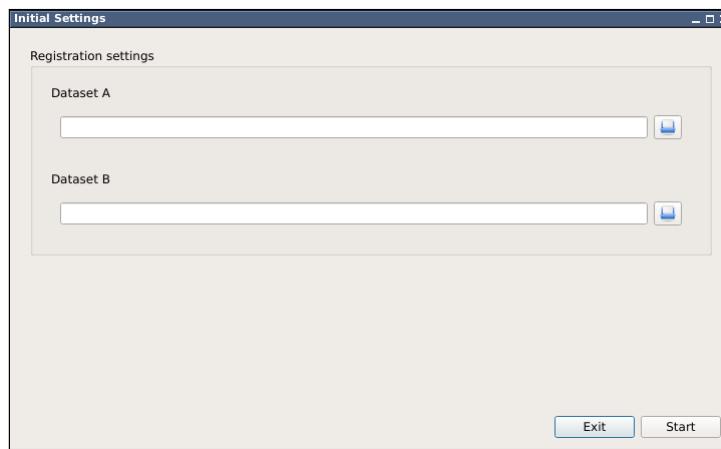


Figure 8: Initial settings dialog. Two datasets (as *.obj* or *.pcd* files) must be chosen.

After starting the actual registration interface, the software looks like Figure 9. In order to navigate the data, the software supports common ways of navigating the virtual camera.

⁴⁵For this deliverable the prototype has been ported from a *GNU/Linux* version to a *Windows* version - the former can also import *.ifc* and *.e57* files, however due to time constraints the latter (and therefore the delivered prototype) does not yet support IFC/E57.

Dragging with the *Right Mouse Button* rotates the camera around the center of the scene. *Right Dragging while holding Shift* allows to *zoom* in and out of the scene and *Right Dragging while holding Ctrl* pans the camera up/down or left/right relative to the viewing angle.

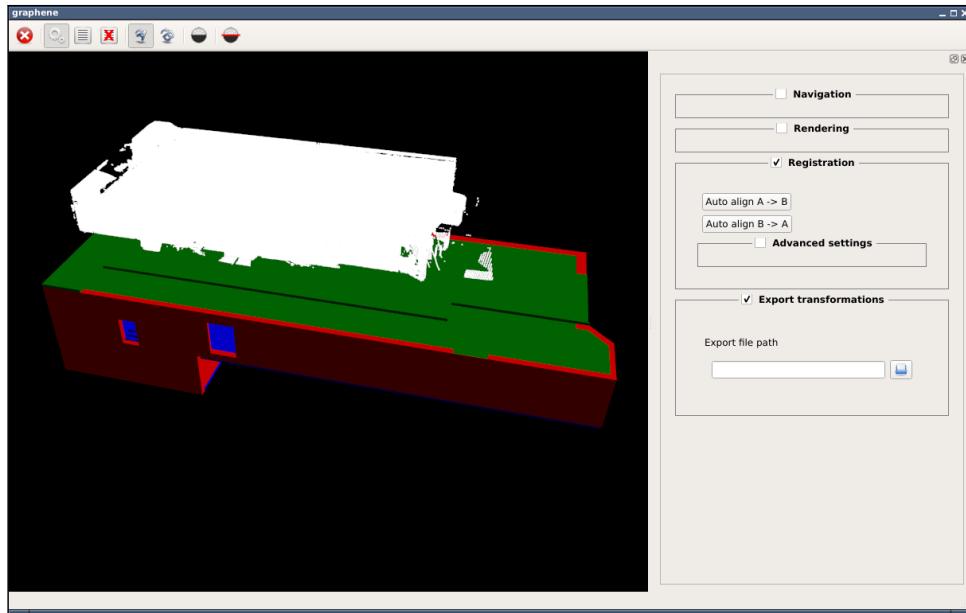


Figure 9: Software interface after loading two corresponding datasets.

Since mesh data often occludes important parts of itself, the software includes a *Vertical Plane Cutoff Mode* where all parts above a virtual plane are cut off (see Figure 10). To enable Cut-Off mode, two buttons in the upper toolbar were added. If is active, clipping mode is enabled and mesh data is cut off at the height of the virtual plane. Activating enables the user to change the virtual plane height by *Dragging* with the *Left Mouse Button*. For a brief overview of all shortcuts see Table 1.

A.3 Spatial Alignment

In order to perform the actual registration (i.e. spatial alignment) two steps are necessary. Before using the automatic alignment the two representations must be manually aligned in a coarse way. This can be done by dragging either representation while being in the respective *Transformation Mode* which can be enabled using the corresponding buttons in the upper toolbar. If is active the *first* representation can be dragged using the *Left Mouse Button*; conversely activating allows for left-dragging the *second* representation.

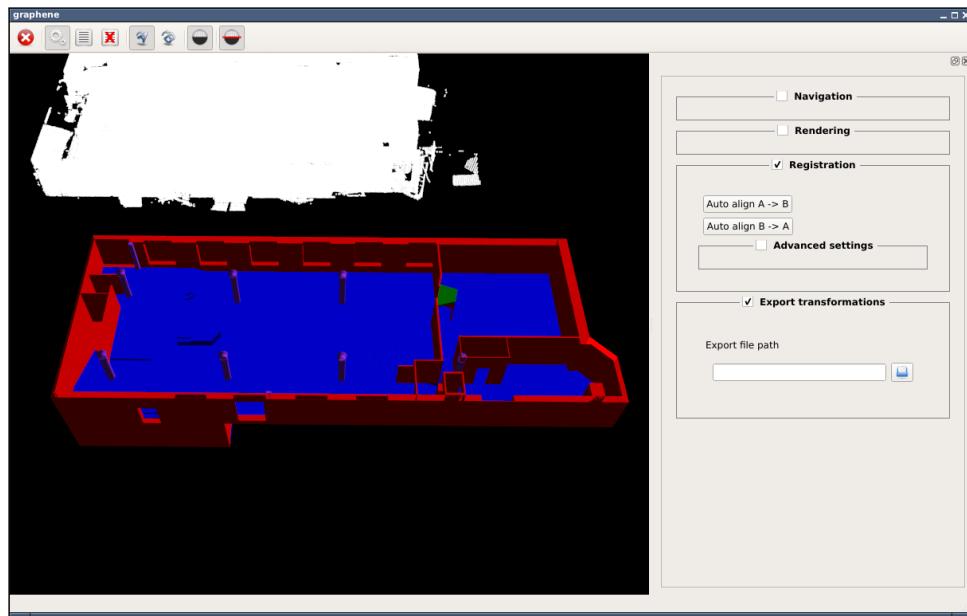


Figure 10: The interface in Vertical Plane Cutoff Mode.

After coarse alignment (see for example Figure 11) the fine adjustment of the transformation is done by the automatic spatial alignment algorithm which can be triggered using the corresponding buttons **Auto align A -> B** (aligning the first representation to the second) and **Auto align B -> A** (respectively align second to first).

A.4 Export Registration Results

For the purpose of demonstrating the prototype software this step is not necessary. However for the sake of completeness the software also supports exporting the computed *Registration Transformation* as a *.n3* file in *Turtle RDF* format. In order to do this, after completing the automatic alignment computation, choose an output file and save to it using the file chooser dialog accessible in section “Export transformations” in the settings pane on the right hand side of the GUI.

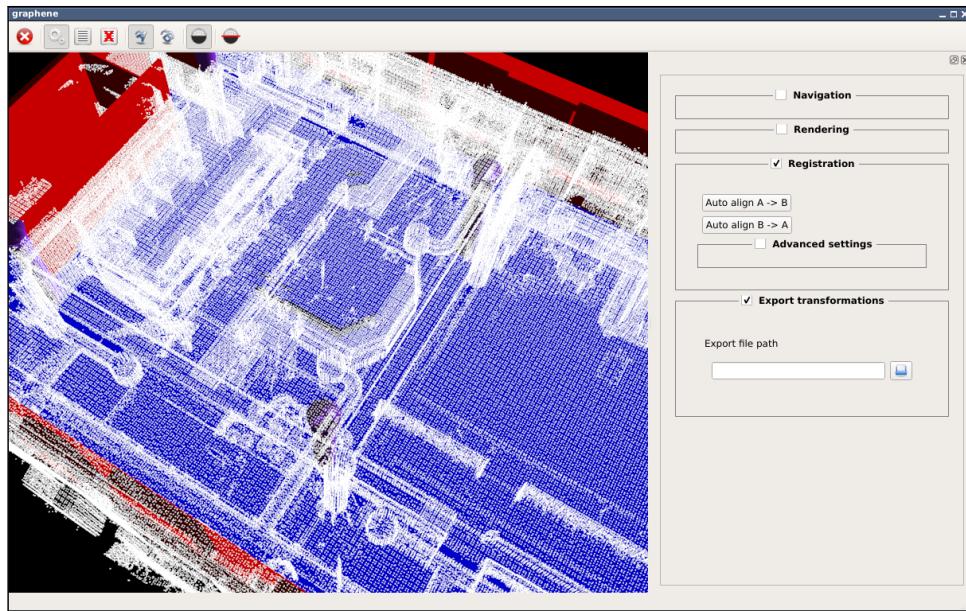


Figure 11: Software interface after roughly pre-aligning both representations using *Transform Mode*.

Function	Shortcut	Available in Mode
Rotate Camera	Right Mouse Drag	Any
Zoom Camera	Shift + Right Mouse Drag	Any
Pan Camera	Ctrl + Right Mouse Drag	Any
Modify Cut-Off Plane	Left Mouse Drag	Plane Edit Mode 

Table 1: Available shortcuts for navigation/cut-off mode