Parametric Architectural Design with Point-clouds

Volvox

Mateusz Zwierzycki¹, Henrik Leander Evers², Martin Tamke³
¹,³ The Royal Danish Academy of Fine Arts, Schools of Architecture, Design and Conservation ² CITA
¹,²,³ {mzwi|heve|martin.tamke}@kadk.dk

This paper investigates the efforts and benefits of the implementation of point clouds into architectural design processes and tools. Based on a study on the principal work processes of designers with point clouds the prototypical plugin/library - Volvox - was developed for the parametric modelling environment Grasshopper in Rhino. The prototype allows us to discuss the necessary technical layer for the task, benchmark the tool, and finally to evaluate the benefits, that this approach has for architectural practice, through a series of use cases.

Keywords: point-clouds, Architectural Design Tools, 3d Scan, multithreading, .net

INTRODUCTION: POINT CLOUDS IN ARCHITECTURAL DESIGN

The last years has seen a massive proliferation of 3D scanners in all building related disciplines[1][2]. Affordable scanners are today available for all sorts of applications from airborne scans with drones, over high-resolution indoor scans to handheld scanners, which provide real-time data. 3D capturing technology is becoming increasingly faster and mobile (see Figure 1) and provides a new link between the build environment and its representation in architectural design.

Point clouds as a geometrical class

The output of all 3d capturing devices are point clouds. Their sheer size poses a challenge to the current computational infrastructure and processes of design. Point clouds are computationally heavy and the processing of them is demanding in terms of hard- and software. Their proliferation is hence directly linked to the advancements in computational power. Point Clouds are a relatively new geometrical class, which was only introduced in the 2000s (Rusinkiewicz and Levoy 2000). Since then only specialist software was able to handle point clouds with reasonable speed. The current market of point cloud software is oriented towards professionals from engineering and land surveying industries. The leading software as those from 3D laser manufacturers, such as FARO scene [3] or Leica Cyclone [4], from CAD producers, such as Bentley Pointtools [5] or Autodesk Recap [6]. Design and engineering processes can currently not take place in Point Clouds, which necessitates a transformation of these into CAD formats by means of software, such as FARO Point Sense [7], Clearedge Edge Wise[8], Trimble Real Works[9], Scalypso [10], PointCap [11]. The mentioned tools are expensive: a Faro Scene licence costs e.g. around
11.000€ [12]. The functionality of these tools is mature and they have e.g. their own advanced graphic pipelines. All of this software is however conceived as processor of point cloud datasets before their ingestion into stakeholders design software. For this task a set of generic workflows is offered, which allows specialists to clean, subsample and transform and to some extend analyse point clouds and their quality (Tamke et al. 2016). The outcome of these processes is either a digital model, in a format that is directly editable in a CAD package (e.g. polygon, Nurbs or BIM formats) or a processed point cloud, which can be read by some architectural modelling programs, such as Recap for Autodesk Revit or E57 and XYZ point cloud formats for Archicad. These design environments do not foresee a further interaction with the point clouds. Users have currently hardly any means to manipulate the point clouds during the design process. This results for instance in retrofitting projects in a situation, where the core of the design, the representation of the building to be designed, is in fact not accessible by the architectural designer. This results in an overly complex workflow, where even simple tasks, such as the transformation of point clouds.
ARCHITECTURAL WORKFLOWS WITH POINT CLOUDS

An investigation and collaboration with stakeholders, who are experienced in the use of point clouds, and the review of nearly 500GB of point cloud and derived BIM data in the European DURAARK[13] project, revealed a general pattern of activities, in which data from a raw point cloud is transformed into a simplified dataset that is considered useful for a design task (Tamke et al. 2015). We could identify 5 steps, see figure 2:

1. Retrieve data - the collection and import of different datasets, which provide insights into the existing structure, such as point clouds and BIM files, but as well photos, 3D legacy plans, 3D models, geoinformation.

2. Allocation - Subsampling of datasets which exceeds the resolution capable by the user's hard- and software. Locating of different datasets to each other and transformation of these, so that they fit to each other geometrically.

3. Analysis and Verification - Generation of a qualitative understanding of the dataset. Assessment of the precision of the dataset, identification of the difference between an idealised design model and the scanned reality, such as non planar walls or floors or deviations.

4. Focus & Abstraction - Interaction with the dataset through filtering of noise, focus on parts and elements in the dataset, which are relevant for the task, addition of new elements and transformation of the existing. Often transfer of the point cloud into the data format of the design environment (Polygons, Nurbs).

5. Design - Refinement and visualisation of the data and integration into the further planning process.

As previously mentioned the investigations into the state of the art of point cloud related processes revealed that current point cloud software restricts users to a very limited set of workflows, analysis and representations, and is restrictive in terms of input geometry. This implies for instance, that design entities cannot be linked to point clouds and that the analysis of point clouds cannot be adapted to the design cases or language of the architects.

DESIGN INTEGRATED PARAMETRIC EDITING TOOL FOR POINT CLOUDS - VOLVOX

The five recurring steps: Retrieve data, Allocation, Analysis & Verification, Focus & Abstraction and Design, form the base functionality of Volvox, a plugin developed by the authors for the highly popular Rhino/Grasshopper visual programming environment. It provides basic and advanced point cloud manipulation and analysis tools such as: reading/writing various file formats (including industry standard E57[14]), transformation, merging of multiple point clouds, various subsampling techniques, cropping based on geometry or scalar field, comparison of point cloud with other point clouds or meshes and various manipulation tools based on scalar fields. It took one month to develop and release the Volvox (version 0.1.0.0), with a basic point cloud manipulation toolset. After four more months of development, a grand update was released (version 0.2.0.0), which brought multiple changes and new tools to Volvox. One of the biggest problems encountered during the development was a lack of .NET libraries for reading and writing E57 file format. The widely used library (libE57) is written in C++ which is not easy to inte-
Implementing point clouds in a parametric modelling workflow - Techniques and Technologies

Grasshopper is a node-based, algorithmic modeling application, utilizing the RhinoCommon library geometry classes (e.g. Meshes, Nurbs geometry, various types of curves). Any type of geometry sourced from RhinoCommon has to be wrapped in a separate class to be used in Grasshopper's visual workflow (tasks include here: describing how to draw the geometry, apply transformation etc.). This wrapping class enables the developers to seamlessly add new types of geometry, which can be modified with the already existing Grasshopper nodes. The development of such a class for the RhinoCommon PointCloud class was the first step in the creation of Volvox. It was quickly realized, that the node-based nature of Grasshopper is not optimal for handling memory expensive data types such as point clouds - each node creates most of the times a local copy of the data to work on. An algorithm, which loads for example the point cloud from a file, subsamples it and then moves it, creates 3 separate copies of the data. Our solution is to develop a programmable node (called "Engine") which can copy the point cloud only once and which executes a set of parametric instructions on it, see figure 3. The parametric nature of the instruction set for the Engine enables new ways of working with point clouds. The user can work on the Grasshopper solution with a highly subsampled cloud. And once it's ready the same stack of operations can be executed on a full resolution cloud - simply by changing the subsampling parameter.

Volvox consists of the Engine, Instruction components and further components, which operate in linear Grasshopper fashion. These are components which don't require a local copy of the point cloud, and are mostly meant for analysis (average point, closest point etc.) The plugin utilizes multithreading on multiple levels. The Engine itself works in the background thread. Some of the Instructions, as well as regular components, are also multithreaded. In case of the Engine, the thread separation has the advantage, that users can use the main Grasshopper thread simultaneously. The use of multithreading in other areas is meant to speed up the overall workflow, giving a better, more responsive user experi-
Table 1
Benchmark of standard operations in Volvox and Cloud Compare. We measure the time required for similar tasks and acknowledge that the functionality of the algorithms of the two applications can not be 1:1 translated.

<table>
<thead>
<tr>
<th>Task</th>
<th>Evaluation process</th>
<th>Volvox</th>
<th>CloudCompare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import</td>
<td>64 Million Points point cloud</td>
<td>92s</td>
<td>120s</td>
</tr>
<tr>
<td>Random Subsample</td>
<td>Random Subsample to 50% of points for a 64 million point cloud</td>
<td>7s</td>
<td>4s</td>
</tr>
<tr>
<td>Spatial Subsample</td>
<td>Spatial Subsample to 0.1 unit distance between neighbouring points for a 64 million point cloud</td>
<td>10s</td>
<td>24s</td>
</tr>
</tbody>
</table>

**Evaluation of technological approach**

The competitive speed of Volvox is revealed in a benchmark of Volvox against one of the most popular and versatile point cloud editors - CloudCompare[16] - a freeware tool for Windows, Mac and Linux developed by (Girardeau-Montau et al. 2005), see Table 1.

**Using Point Clouds in Architectural Workflows**

The evaluation of the tool took place in a series of use cases with stakeholders. These use cases were explored in common projects and workshops between researchers and practitioners. They investigated both existing and novel ways to address point clouds. The question was to which extent the design processes with point clouds can be improved or even charter into new fields, when a customizable computational environment for point clouds is used.

**Dynamic creation of Plans and Sections.** Sections can be dynamically created within point clouds.

Architects work traditionally with plans and sections. This approach can be easily facilitated with the Volvox Plugin. We find, that within renovation tasks, plans and sections of the existing building stock are either missing or deviate from the as-built. The section component in Volvox creates dynamic sections on the fly. These can thereafter either be used as the base drawing, to adjust existing drawings or as underlay for new drawings. Figure 4 shows how Volvox is used to analyse the Nikolaj Kunsthalle in Copenhagen, Denmark and display it in scale in plans and sections.

**Multiple crops from a point cloud.** Relevant details for design can be extracted by means of multiple crops based on euclidean and freeform shapes.

The utility of the filtering of point cloud content for architectural design was explored in a student workshop at CITA, KADK [17]. The Volvox Plugin enabled architectural design students to manipulate point clouds in Gigabyte size and extract from full building scans within minutes the information, which was useful to their design tasks, see Figure 5. Cropping spaces from point clouds enables architects to embed their design in a realistic looking context or to extract as-built information in an easy and aesthetic way, see Figure 6.

**Interactive comparison of models.** Analysis of the deviation of point cloud in relation to points, planes, meshes and other point clouds.

The monitoring of the progress on building sites, the control of geometric deviations between planned and as-built geometry and the check of planarity of building elements during construction planning are essential tasks in AEC (Tamke et al. 2015). The Volvox plugin gives the user the ability to calculate and store e.g. deviations between point cloud and a mesh of the same object. The deviation from each point to the mesh is measured and stored in the point cloud as UserData for each point. This UserData can then be retrieved and used to colorize points in the point cloud according to a user defined color range. Figure 7 shows the comparison, which the Danish land surveying company LE34[18] did using Volvox. They found deviations between an as-built point cloud scan and the modelled representation...
of a tunnel entrance to Nørreport Subway in Copenhagen, Denmark. The company used Volvox, as their standard software tool could not handle the mesh-based representation of the tunnel, which the company received from their collaborators. The versatility of Rhino/Grasshopper allowed here a fast conversion and user defined display of deviations.

Another common need for a deviation analysis is the deviation from a plane. In energy renovation tasks of existing building stock it is important to know about the deviation of an exterior wall to the optimal plane. With Volvox the best-fit plane of a facade can be found and the distance from each point to that plane can be stored as UserData, see Figure 8.

**Automated creation of architectural geometries.**

*External point cloud engines can be directly accessed from Volvox and enable e.g. automatically generated architectural geometries from indoor 3d scans (e.g. DURAARK IFC reconstruction DURAARK, CloudCompare [16]).*

The creation of architectural as-built geometry can be a time consuming task. The open structure of the Volvox plugin enables the user to extend the core components of Volvox with other libraries and/or point cloud engines. IFC reconstruction and association algorithms from the EU funded DURAARK project were implemented in Volvox. This enables the Volvox Plugin to overlay and associate point clouds and BIM models parametrically (see Figure 9), and/or to automatically reconstruct a BIM model from indoor point cloud scans. This process takes place in a fraction of the time it would take to manually model the reconstruction (Ochmann et al. 2016). One example from the DURAARK project shows a decrease in time spend of approx. 70%, see Figure 10 (Tamke et al. 2015).

**Interfacing on Point Level.** *Volvox grants access to the point level within a point cloud.*

Users can program expressions to attach additional data as Rhino UserData to point clouds, e.g.
Figure 8
Deviation from plane of a facade of the Nyborg Strand Hotel and Conference Center. Red < -100mm, green: 0mm, Blue > 100mm.

Figure 9
Parametrically overlay and association of point cloud subsets to BIM elements through a link between external tools and Volvox.

Figure 10
Automatic reconstruction of BIM geometry using Volvox extended with the IFC reconstruction algorithm from DURAARK (Ochmann et al. 2016). The time needed for algorithm to reconstruct model was approx. 10 min per building/floor. (Point cloud from Symetri[19]).
the distance per point to target or the height of points above a plane. These values can be visualised through interactive coloration of the clouds. Figure 11 shows from the left the original colored point cloud, the colorization of the height of the points with a custom color range and the point cloud culled by Height UserData above a certain threshold. The point cloud used here is a scan of an installation designed and built at the AA_AArhus Visiting School 2015.

The expression can as well operate using simple mathematical formulas or functions available in .NET (trigonometry, conditionals, operators), see Figure 12.

3D print of point cloud. Voxelization allows to 3d print any point cloud.

With the Volvox Plugin it is possible to mesh a point cloud using the voxelization component. The approach was first implemented by (Shepherd and Treddinick 2015). Our component creates a colored watertight mesh from the point cloud. The output mesh can be directly 3D printed. Figure 13 shows the voxelization of the installation from the AA_AArhus Visiting School 2015. From left to right it shows the original point cloud, a coarse voxelization and a fine voxelization.

Evaluation of use in design cases

The evaluation showed the benefits of the integration of point cloud functionality into design environments. Stakeholders fed back, that already the ability to focus and refine in a context specific way provides new insights for architects, as they can at any time overlay a model with the physical reality without disturbing information clutter from the rest of the point cloud. The simultaneous representation of CAD, parametric environment and large point cloud data was as well seen as beneficial, see Figure 14.

Users with a modest knowledge of visual scripting are furthermore able to quickly create customised analysis and design evaluation tools. This allows them to engage in their design in novel ways with point clouds and create novel design driven tools and workflows in this field.
CONCLUSION
The integration of a point cloud editing functionality into architectural design environments has proved to be beneficial for designers, as this frees the users of point clouds to be authors and create the data, rather than sole consumers of data - the role that current point cloud software reserves for them. The integration of point cloud functionality into a parametric environment was evaluated positively, because users could adapt existing workflows and create new ones in a project specific and personal way. To use a parametric modeling environment as front end for the point cloud integration proved to be technically challenging. Our approach turned however out to be quite performative - being as good or even better in a comparison with the state of the art Open Source tools for point cloud editing. The invention of a single node point cloud engine enables efficient workflows, which allow for preliminary experimentation and speculation with point clouds and a later high quality output of professional results. The Volvox plugin has been downloaded more than 800 times within the first four months after its release from its repository on food4rhino [21]. The source code for the plugin and the custom developed .NET E57 read/write library are furthermore published as open-source on GitHub. As the Volvox plugin is split into separate libraries, it’s core can be used in custom scripts and other components - taking advantage of a seamless integration of point clouds in Grasshopper.

ACKNOWLEDGMENTS
The research, that led to Volvox took part in the frame of the EU funded DURAARK project (FP7 - ICT - Digital Preservation, Grant agreement No.: 600908). The project wouldn't have been possible without the strong support of industrial stakeholders, such as LE34 Denmark, Symetri, Sweden & Denmark and Plan3D Germany.

REFERENCES
http://www.autodesk.com/products/recap-360/overview
http://faro-3d-software.com/CAD/Products/PointSense/index.php
http://www.clearedge3d.com/
[12] http://www.3dscannerstore.co.uk/software/
[18] http://www.le34.dk/