D.5.2 ï Second Year Report

WP5 ï 3D Artefact Processing

Version3.5 - FINAL

29 November 2010

Grant Agreement number:	231809
Project acronym:	3D-COFORM
Project title:	Tools and Expertise for 3D Collection Formation
Funding Scheme:	FP7
Project co-ordinator name, Title and Organisation:	Prof David Arnold, University of Brighton
Tel:	+44 1273 642400
Fax:	+44 1273 642160
E-mail:	D.Arnold@brighton.ac.uk
Project website address:	www.3d -coform.eu

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 231809.

Authors: Roberto Scopigno

Consiglio Nazionale Delle Ricerche, ISTI (CNR-ISTI)

Contributing partner organizations: Katholieke Universiteit Leuven (KUL)

Eidgenossiche Technische Hochschule Zurich (ETHZ)

Fraunhofer Gesellschaft Zur Forderung Der Angewandten Forschung E.V. (FHG-IGD)

Technische Universitaet Graz (TU Graz)

Media Integration and Communication Centre, University of Florence (MICC)

Breuckmann

Spheron

Table of Contents

1 Executive Summary

This document presents the status of the work under Work Package 5 (WP5) “3D Artefact Processing” - at the end of the second year of activity of the 3D-COFORM project.

The activities follow smoothly the original plan drafted in the project Description of Work (DoW). All planned partners are now contributing to WP5 activities (a few of them have started to be active mostly since the start of Year 2). The end of Year 2 was one major milestone for WP5 and the 3D-COFORM project, since many beta releases of the tools were planned to be delivered on month M24. Major activities performed and results obtained in the second year are: several new versions of MeshLab and a progressive evolution and consolidation of the tool; delivery of the basic infrastructure for the shape analysis component; new algorithms for the completion of sampled models and for the generation of LOD representations / rendering from CityEngine models; a processing pipeline for the procedural modelling of landmark buildings, that has been designed and partially implemented and tested; finally, we have revised our research plan on fitting GML models over sampled datasets.

No major problems or major deviations arose during the second project year. The activities are going to continue in Year 3 according to the plan described in the project contract.

The overall organization of the document is as follows. Section 2 gives a brief presentation of the project structure, how WP5 activities and tools are located in the overall framework of the project, and relations of WP5 components with respect to the other components developed in 3D-COFORM. Sections 3, 4 and 5 present in details the work done in Year 2 and the results obtained in the three tasks of WP5. Section 6 reports on the milestones; some concluding remarks are presented in Section 7. Finally, the publications produced so far are listed in Section 8.

2 WP 5 ζ 3D Artefact Processing - Detailed description of work

The 3D-COFORM framework and its components have been divided into four clusters:

1. Acquiring and Processing (A&P), encompassing the developments in WP4/WP5
2. Integrated Viewer/Browser (IVB), encompassing the developments in WP6/WP7
3. Modelling and Presenting (M&P), encompassing the developments in WP8/WP9
4. Repository Infrastructure (RI), encompassing the developments in WP3

The central topic of WP5 is *shape processing* and *analysis*. On one side, transforming sampled raw data in *high quality digital representations* (i.e. all the geometric algorithms needed to process raw data and geometry-based representations); on the other side, developing a number of functionalities (segmentation, feature detection, component matching) which allow to structure the geometric data making it possible to implement more sophisticated *shape analysis* or detect *semantic correspondences* between different shapes or sections of a given model. In the latter case, an important contribution will be a methodology for turning 3D reconstructions into procedural models.

All tools will inter-operate with the repository (WP3): input data will be read from the repository (retrieved) and modified models will be uploaded back (ingested, which will include storing back both geometry and provenance data).

In most cases, input data for the tools designed in WP5 are:

- raw data coming from 3D scanning devices or from ARC 3D (production of raw 3D data from images), stored in the Repository Infrastructure;

- 3D meshes of whichever origin, also stored in the Repository Infrastructure.

The Repository Infrastructure is therefore the common data source for all the components and algorithms designed and implemented in WP5. It is also the sink used by all of the WP5 components for uploading the results produced after processing the input data, enriched by the related *provenance metadata* that will encode the specific processing action executed over the 3D data.

The *components* to be designed and implemented in WP5 have been described in deliverable D3.1 ς First Year Report on WP3 ς Repository Infrastructure, where the reader can find a detailed description of all the inter-components interactions; functional specifications were also presented in this report.

2.1 Task organization and planned work

The activity of WP5 is subdivided into three tasks. The activity planned in the second year for the three tasks have been defined in the Description of Work (DoW) document as follows:

Task 5.1 – Processing tools for mesh-based models

- delivery of MeshLab v. 1.3 (supporting capabilities for the management of very large dataset composed by many range maps).

Task 5.2 – Methods for shape analysis

- delivery of the beta release (first basic algorithms for user driven geometric segmentation and tagging)

Task 5.3 - Fitting procedural models to classify acquired 3D models

- delivery of the beta release of the fitting tools

2.2 Work performed

The work performed in the second year of activity is described in the following chapters, focusing on each single task.

Since the focus of this WP is both concerning the design of *new algorithms* and *new tools*, it is important to say here that the activity at the algorithmic level by its nature should follow different strategies and approaches than the system design. Not all the algorithms designed and implemented will perform at the same level of quality; some new ideas could result in being more successful than others. The decision of what solutions should then be integrated in the final system (e.g. MeshLab) will therefore follow the results of the preliminary assessment phase. It is common in the evolutionary approach endorsed that most of the algorithms proposed will find their way in the components to be delivered, but not all of them. Therefore, the activity at the algorithm level (design, implement, assess) is preliminary to a second phase (engineering, bug fix, porting in the final system component) and will be done only for the more successful algorithmic solutions.

3 Task 5.1 ƺ Processing tools for mesh-based models

3.1 Activities and results in Year 2

MeshLab

The major result in Task 5.1 is the design and implementation of further extensions and consolidation of the MeshLab tool. Large sections of the MeshLab architecture were redesigned, not only in order to improve tool stability and software maintenance, but also to add support towards new paradigms of use.

In particular, efforts were focussed to include in MeshLab ă ƚ Ÿ Ŝ Ž ƚ ǀ Ž Ŝ LJ ű ă ō ƚ NJNJ ƚ ǿ Ŝ Ř Ŧ ŀ ō ă ű NJ ŀ Ŏ ű ƛ ƚ ǀ Ф ! ǿ ƚ NJ ʅ Ŧ ƒ ƚ ǿ Ŏ ƚ ǀ ă ƛ ă ű ă which are useful to fulfil a ǀ Ŏ Ŝ ƚ Ŧ complex task. Following this new approach, a MeshLab filter should become a link of a functional chain defined in order to complete such a task. If the result of a step in the middle of the chain will change, then all the other following links will be dynamically updated. This approach should be useful also to direct the MeshLab Ŏ ƚ Ř Ŝ Ř Ŝ Ǿ Ŝ ƒ ƚ LJ Ÿ Ŝ ǀ ű ƛ ǀ ƚ NJ Ř Ŝ NJ ű ƚ Ŧ dz ƒ Ŧ ƛ ƒ ű K Ŝ N by the review committee (after Year 1 review).

On the other hand, following the usual improvement route, we have planned a set of major feature improvements of MeshLab to be implemented during the second year of the project:

- Testing and Benchmarking: since MeshLab version 1.4 (delivery planned in spring 2011, beta in January 2011) we started to include some form of automatic testing of all the filters and functionalities in order to improve significantly the overall robustness of the tool and to guarantee stability in performance.

- RI infrastructure: following the evolution of the deployed version of the RI we have prototyped again the RI communication features of MeshLab.

- Texture Alignment and processing: An integrated framework for the management of images and 3D models has been introduced in v.1.3, while MeshLab v.2.0 (release planned in early 2011) will include the functionalities required for the registration of photographic images over sampled 3D models (image-to-geometry alignment or registration) and tools for the integration of colour data contained in multiple images over 3D models (texture map synthesis or colour-per-vertex encoding).

The new versions of MeshLab delivered in Year 2 are:

- April 30, 2010 - v. 1.2.3
- September 2010 - v. 1.3 beta
- *November 2010 - v. 1.3*

New features added in MeshLab version 1.3 include:

- Totally restructured view/window mechanism. Now available:
  - multiple windows of the same mesh
  - standard orthographic viewing directions (up/down/left, etc)
  - copy/paste of current viewing parameters between different windows (you can even save them for later re-use...);
- A new single *shared* layer window replaces the old approach (that was forcing the existence of one layer for each document); the new shared layer is now relative only to the current mesh document (a mesh container that holds a set of meshes which are correlated according to the user's requirement).
- New behaviour for filter creating meshes. Empty mesh documents are now meaningful (for example, this is useful to create procedural meshes)
- New interface and behaviour for decoration plug-ins, now they can have dynamic parameters.
- The isoparametrization filter is now completed and fully debugged (this filter implements the new algorithm proposed in paper [T5.1.1] described in Task 4.5). It allows the production of a parametrized mesh with an associated texture map, offering ideal characteristics for the conversion of scanned 3D models with colour attribute to a simplified base mesh with texture-mapping encoding of the colour channel.
- New Radiance Scaling rendering mode (a new shader providing a better visualization of curvature variations on models).

A major redesign of the MeshLab architecture was needed to port in the tool the algorithmic solutions for colour management. The new solutions devised in WP4 (3D Artefact Acquisition) by CNR-ISTI to manage the colour data (set of photographs acquired either by a scanning device or, better, by a standard photographic camera, with no information on the inverse mapping to the digital 3D model) are being ported to the MeshLab platform. Since MeshLab was initially designed by focusing mainly on geometry-based processing, we had to extend the system by including a number of data structures for processing images as well, and for managing the relations between images and 3D meshes. This activity is currently ongoing: a number of internal instruments and revised data structures have been already introduced in version 1.3 beta. Porting of the following algorithms is in progress: (a) automatic image-to-geometry alignment; and (b) for the synthesis of blended texture maps. This porting is de facto a re-design and partial re-implementation of the new solutions designed by CNR-ISTI in WP4. Those features will be released with the new version of MeshLab (beginning 2011).

MeshLab validation / bug fixing

According with the comments and suggestions of reviewers (see the Year 1 project review results), efforts were made to follow the reviewers suggestions. See some details below.

Recommendation 2 (maturing): Consolidate the software tools that are planned to be used routinely in the centre of competence by setting up benchmarking suites and not just unitary tests (this may require additional resources).

Definitions of two categories of testing protocols have been started, one based on automatic thorough unit testing of all the filtering capabilities of MeshLab and another one based on a small number of selected standard workflows based on reference datasets.

For example, some default set of range maps are chosen from different acquisition devices (triangulation, time-of-flight, photogrammetric) that can be used to produce a final object following a precise pipeline of processing. The previous knowledge of the master-quality result of the processing will allow us to evaluate/detect possible inaccuracies or errors produced by tested MeshLab instance.

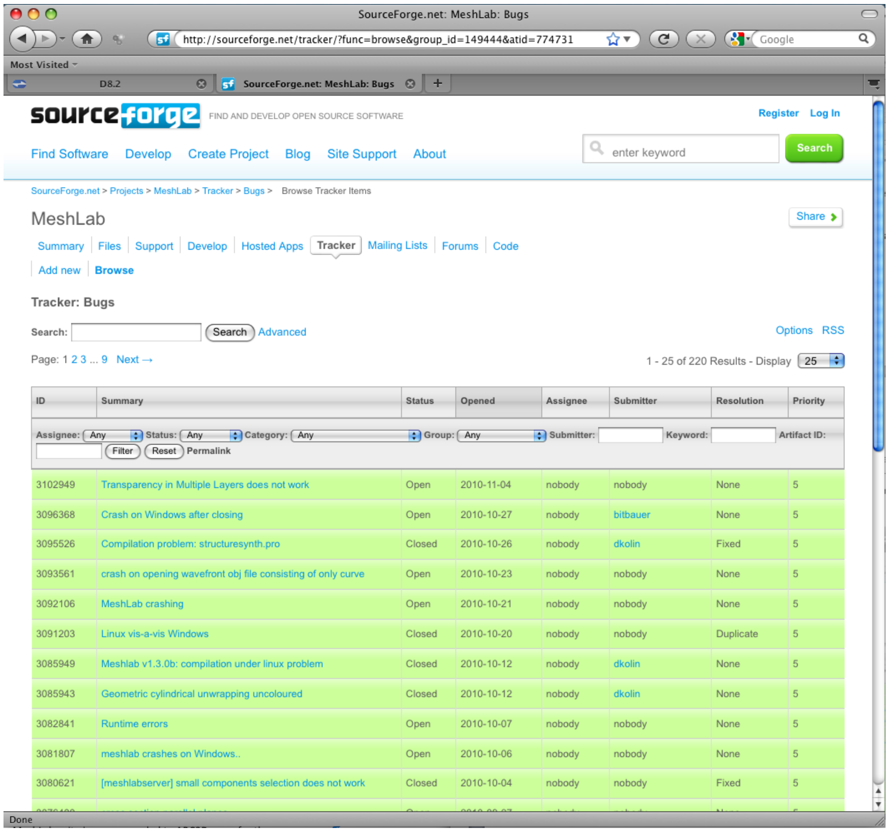
Design is in hand of the required extensions and modification of the system to provide the unit test feature, as requested by the reviewers.

Recommendation 3 (users): involve a variety of users in the training, testing and validation of the tools. For MeshLab, use an existing web-based bug tracking system and devote more efforts at treating the feedback (this may require additional resources).

MeshLab provides from the very beginning of its story an open web-based bug tracking system. We decided to use the instruments provided by sourceforge.com (see at:

https://sourceforge.net/tracker/?func=browse&group_id=149444&atid=774731)

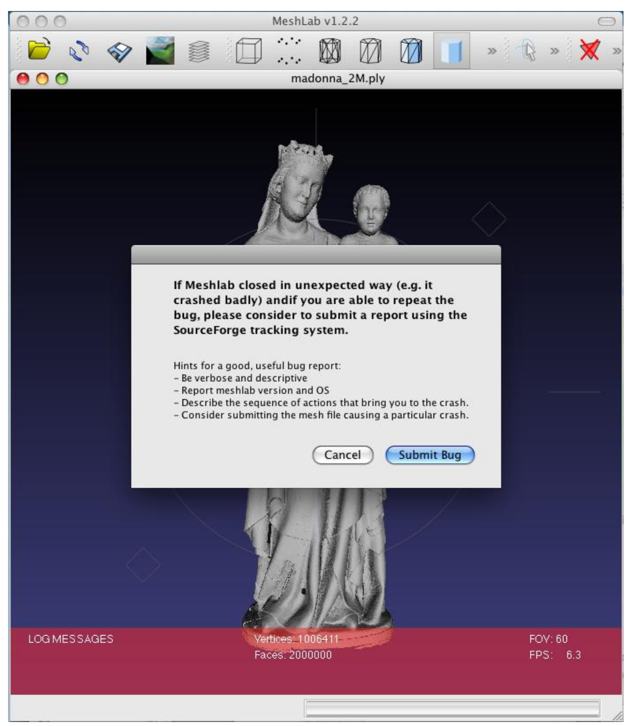
More than 220 bug reports have been opened so far by external users (and treated by ISTI-CNR staff). Probably we have not explicitly presented at the review meeting (where the presentation was very short indeed) the size and collaboration of the existing community of MeshLab users.



This resource is also accessible directly from MeshLab

Menu MeshLab-> help-> submit a bug

(see figure on right side).



Many notable bug fixes have been consolidated since the delivery of version 1.2.1.

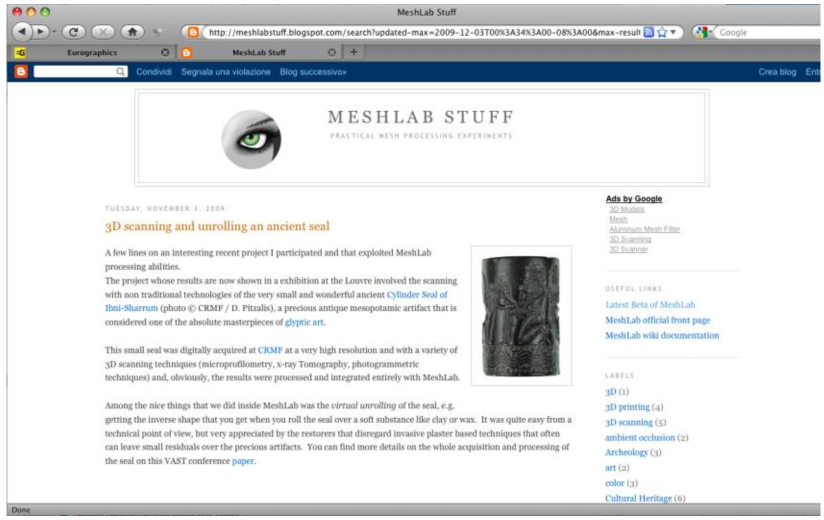
Another resource setup for MeshLab users is the MeshLab Blog:

http:// MeshLabstuff.blogspot.com

(see figure on right side).

It contains much useful information concerning new releases and *how-to-do* descriptions. It is obviously open to comments by the users.

It is currently visited by around 700 users per week.



New algorithmic results

The activity on mesh parameterization progressed. The activity related to the design and implementation of the isoparametrization algorithm (see paper [T5.1.1]) is now finalized, the algorithm has been ported to MeshLab and fully debugged.

We have worked also on algorithms for: (a) converting triangulated meshes into meshes based on quadrilaterals (already ported to MeshLab) and (b) simplification of quad-based meshes. The new quad-based simplification algorithm has been published (see paper [T5.1.3]) and we are currently porting this code to MeshLab.

Early assessment

MeshLab has been used in several test cases directly coordinated by ISTI-CNR (see for example the Madonna di Pietranico work described in Task 8.3: re-assembly of fragmented artefacts; the scanning of Michelangelo's Pietà done at Galleria dell'Accademia museum; scanning of architectural scene done at S. Gimignano, Pisa and Florence). These test cases allowed us to test the features and performances of the tool on highly complex application scenarios.

Training

MeshLab was extensively used in all 3D-COFORM training initiatives.

Watermarking of 3D meshes

The watermarking tool has been integrated within MeshLab as a plug-in and appears in the filter menu. This has been done according to MeshLab guidelines for software development. Such a processing tool is composed of two basic parts: an embedder and a detector. The embedder takes as input the to-be watermarked 3D mesh, the watermark parameters and other settings such as power, false alarm probability and so on, and gives as output the watermarked 3D mesh. On the other side, the detector takes the to-be-checked 3D object (supposed watermarked) and the watermark parameters, and gives as output a Boolean answer (e.g. the 3D mesh contains or not the searched watermark). The 3D input data formats are all those that MeshLab supports such as 3D-Studio, STL, Stanford Polygon, VRML 2.0, etc.

At the end of the second year of the project such a tool presents a sufficient degree of effectiveness with respect to the expected requirements like robustness, making watermark unperceivable and security.

The publications produced [T5.1.7, T5.1.8] are part of a general analysis carried out within this project in the scientific sector of data integrity and anti-falsification, which are fundamental in the field of multimedia for cultural heritage, multimedia forensics and security where watermarking algorithms are positioned.

Integration with 3D sampling devices

The integration of the sampling devices with MeshLab is progressing, but will be fully demonstrated and debugged when those tools will be delivered (planned delivery is in Year 3 and Year 4).

Concerning the ARC 3D platform, improved management of point-based raw dataset is being included in MeshLab. This is done by working on two different streams: (a) adding new specific point-based features in MeshLab and (b) by designing an algorithm for creating a complete model out of the ARC 3D raw data that could be used also directly on the ARC 3D server. In the second stream, the idea is to generate meshes at the ARC 3D server side. The algorithm aims to give an automatic implementation of the [illegible] ARC 3D [illegible] task 5.1. The algorithm takes a number [illegible] of depth-maps as input, cleans them and merges them into a single point cloud. This point cloud is then used as input to the Poisson reconstruction algorithm to create a watertight mesh. Finally, redundant [illegible] tions, [illegible] i.e. when the images have enough overlap and are not too noisy. If the ARC 3D image shooting instructions are followed, these constraints should be satisfied. An example of a reconstruction obtained by this method is shown in the following Figure 1.

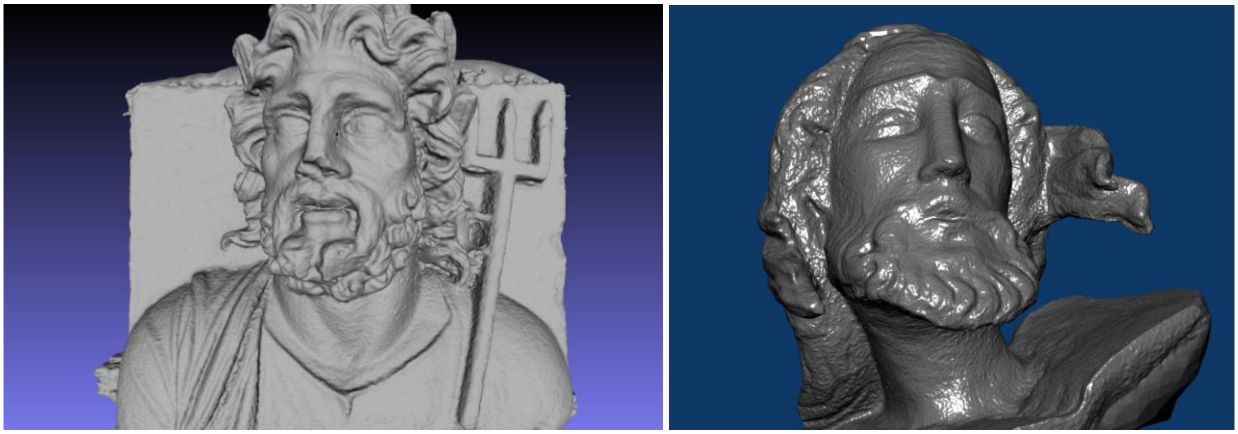


Figure 1. Some examples of meshes produced by the new reconstruction algorithm that can be added to the ARC 3D server.

3.2 Synthetic description of partners contributions

Contribution of partners to Task 5.1 activities and results has been as follows:

- CNR-ISTI: contributed to Task 5.1 with the design of extensions of the MeshLab tool and with maintenance and bug fixing; contribution to Task 5.2 was on surface completion tool and basic mesh curvature filters.
- MICC: porting of the 3D watermarking filter on the MeshLab platform
- KUL: implementation and evaluation of an improved reconstruction pipeline that could be added to the ARC 3D server.
- Spheron: integration of its forthcoming device with MeshLab. Integration is done via an intermediate file that transfers geometry, texture and provenance data.
- Breuckmann: design of the new scanning device and on its integration with MeshLab

3.3 Deviation from work plan

No deviations from the work plan have been registered for Task 5.1.

3.4 Plans for the next period (adaptations to the work plan of the next period)

The activity will proceed according to the original plan. Major results expected are:

- New versions of MeshLab will be delivered during the next year, with a new version delivered as soon as a sufficient number of new features and improvements (bug fixes, redesign of interface) will be finalized. We will try to keep the same frequency of new versions produced in 2009-2010 (approximately, an improved version every four months).
- Watermarking filter: fine-tuning and bug-fixing will be carried out and at the end of the third year a new version will be released as MeshLab plug-in
- Progress in the development of the integration of the sampling devices with MeshLab
- We will evaluate the possible integration of the automatic reconstruction feature in the ARC 3D server; but this will not replace the standard MeshLab plug-in, since in most acquisition projects manual cleaning of the raw data and sophisticated processing will be required to produce good quality results.

4 Task 5.2 – Methods for shape analysis

4.1 Activities and results in Year 2

Objectives of Task 5.2 are: to design user-driven analysis and geometry based segmentation; to design a user-driven tool for completing sampled representations; to design new methods for LOD encoding / rendering of CityEngine models. The activity of the first year of the project focused mostly on the second and third objectives, finalizing the design of a tool based on an active approach for closing holes on 3D scanned models. Conversely, the activity of the second year covered all three objectives.

Shape Analysis component

The objective of the Shape Analysis component for the second period of the project was to implement the functional specification defined during the first period (documented in the deliverables D3.1 – WP3 First Year Report and D5.1 – WP5 First Year Report). Therefore, the basic infrastructure should be implemented and the first level of hierarchical manual segmentation should be supported.

The basic hierarchical segmentation was implemented, based on manual segmentation, such that the user can define boundaries, which are automatically closed (following the shortest path or the main curvature) and then he can select the regions, which can further be segmented. The basic data structures for representing meshes and computing curvature (based on quadric fitting) as well as for querying and traversing the neighbouring information from different points of view were successfully developed and tested by the developers. Additionally, the manual segmentation capabilities were implemented as a plug-in, such as it can be employed in the Integrated Viewer/Browser (IVB) for redlining (see Figure 2) and selecting areas of interest, which could afterward be annotated, seeking always for a very intuitive and simple user interaction. This plug-in was implemented according to the plug-in system of the IVB (please see Deliverable D7.2 – WP7 Second Year Report) and supporting OpenSG, in order to ease the integration within other 3D-COFORM tools. In order to explore the capabilities of our tool, a testing phase was conducted with artefacts coming from an acquisition process, based on Lens-Shifted Structured Light [T5.2.4] for reconstructing fine details of CH-artefacts (resolution down to 55 micron), as the coin displayed in the following image.

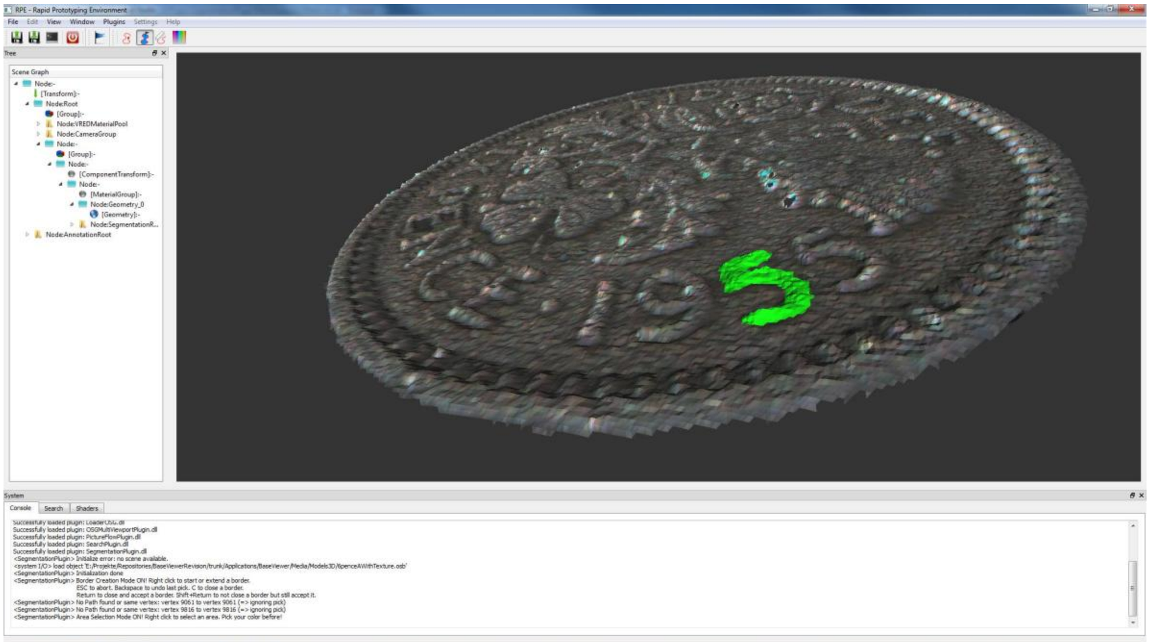


Figure 2. Example of a redlining operation on a 3D mesh.

The segmentation process consists of two steps:

- Generate borders
- Select regions

These two functions can be used in any arbitrary order. Thus, after selecting some regions additional borders can be added to select different regions, or the other way around. Overlapping borders are tracked as well, in order to be able to create intersections between different regions.

Generating borders the user can pick points on the mesh and the points are interconnected through existing mesh edges to generate a border. The connection of two picked points is calculated by finding the shortest path among mesh edges. Special attention has been drawn to usability issues (navigation is possible while defining a contour), in order to achieve an intuitive generation of borders. Figure 3 shows the process of defining some borders on a 3D mesh.

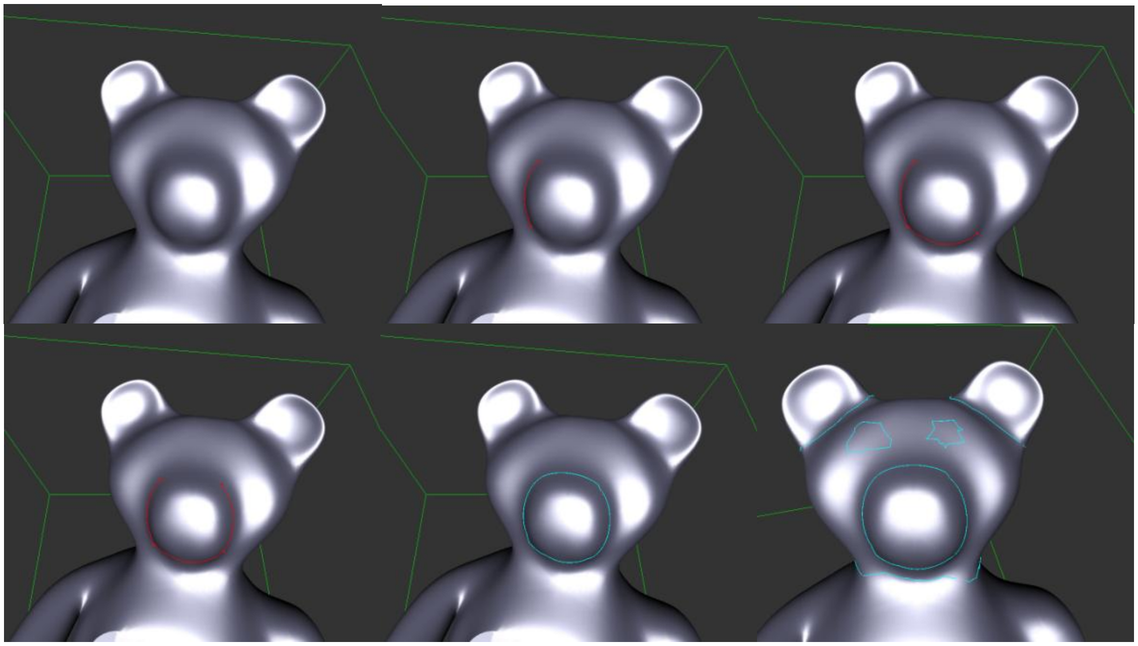


Figure 3. Example of generating borders on a teddy within seconds and a few picks only. The current border is visualized in red while other existing borders are blue.

Selecting regions: the user can select a colour from a colour table and then pick a point on the mesh. A region growing algorithm is applied using as a seed the picked point, which then gathers neighbouring triangles until a border is reached. The reached border is visualized with a thick line in a slightly darker colour.

Since several borders can be defined, arbitrary regions with several independent borders can be selected, in other words a region can be defined by a set of borders and borders can be shared by many regions. For example, the region on the arm of the teddy mesh (Figure 4) consists of three separate borders. Overlapping borders are tracked too, in order to allow the user to generate regions of intersections. Regions can be coloured, and selected or de-selected according to the user's needs.

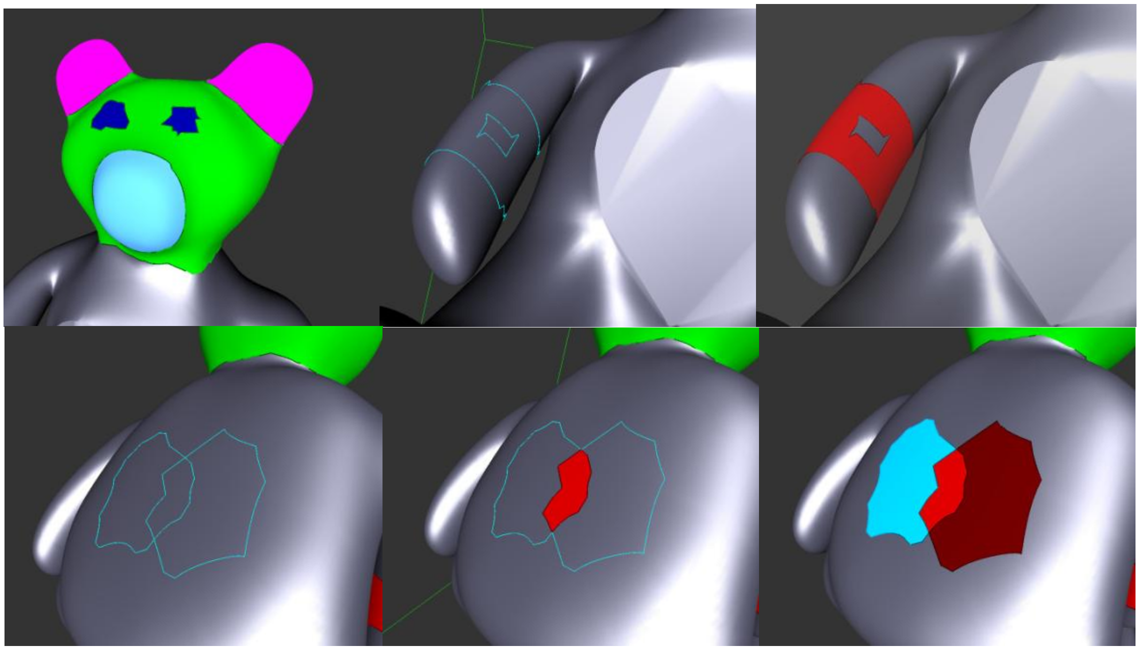


Figure 4. Examples of different regions, created according to different combinations of the selected borders.

Some activity concerned also the extension of the instruments included in MeshLab and in the underlying graphics library (VCGlib) to support shape analysis computations. We have extended the basic features of MeshLab concerning curvature estimation adding new techniques for both estimating curvature in a more geometrically robust way and interactive techniques to drive selection of portion of mesh according to a function defined over these values according to user needs (see Figure 5). In this way the user can define in a very flexible way how to detect a given portion of the mesh according to multiple combined values (curvature, visibility, colour, etc).

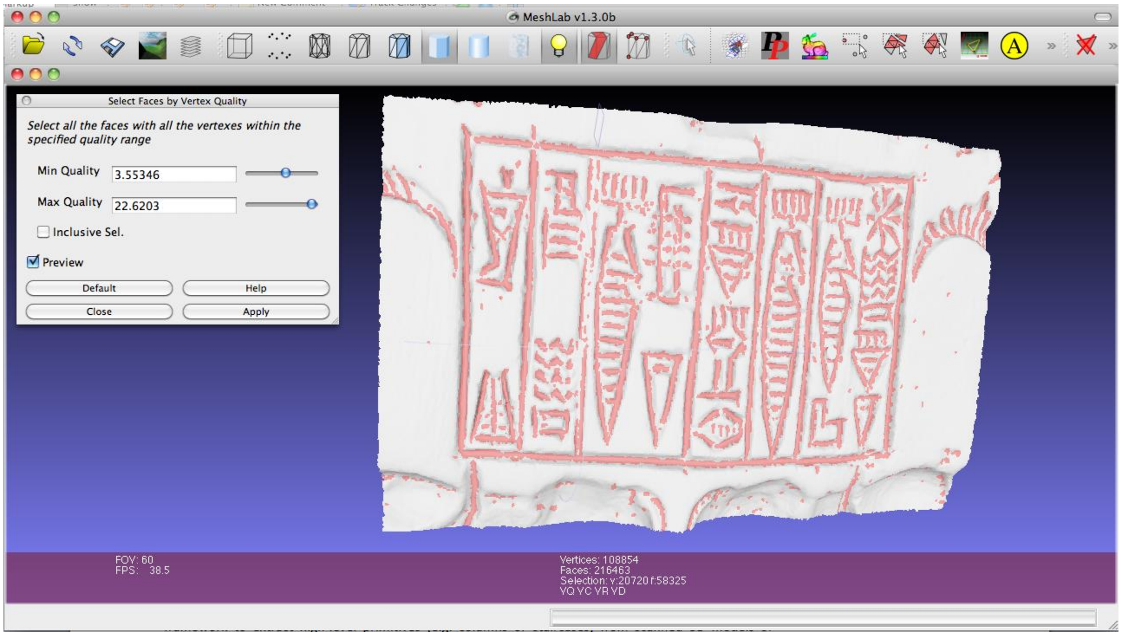


Figure 5. Mean curvature-controlled interactive selection of the inscription. The user can interactiv change the curvature threshold used to segment the pit portions of the inscription. Standard morphological operators (erosion/dilation) can be further applied to better clean the segmented outpu

Tools for completing sampled representations

The second objective of Task 5.2, completing sampled representation, has some apparent similarity with the work done in Task 5.3. It is therefore important to underline here that the goal of this activity in Task 5.2 is to select structured regions of a sampled model that need to be completed (since 3D scanning usually produces incomplete sampling of complex surfaces) and, after selection, to propose geometry to complete plausibly those unsampled regions. Therefore, the activity in Task 5.2 has some similarity with the activity in Task 5.3 (for example, a similar user-driven sketch-based approach is adopted and implemented for the selection of proper surface regions) but the purpose and the geometry processing performed is completely different.

The work has progressed in this sub-task with the development of a user-assisted tool that allows the user to detect elementary architectural elements contained in the sampled data (e.g. columns on the facade of a building). A user-assisted sketch-based framework was designed (see Figure 6) to extract high-level primitives (e.g. columns or staircases) from scanned 3D models of structured artworks (e.g. architectures). The framework offers a unified level of representation of the high-level primitives, so that new types of primitives can be easily added as plug-ins to the main engine. Primitives are fitted with a user-assisted procedure: the user suggests the approximate location of the primitive by means of simple mouse gestures, sketched over a rendering of the model. The viewpoint that was selected prior to the sketching is also taken into consideration as hints on the orientation and size of the primitive. The engine performs a GPU assisted fitting and the result is shown in real time to the user. Ad-hoc gestures cause the system to add and fit groups of primitive in one go (e.g. a column complex, or a sequence of windows). This tool is described in detail in paper [T5.2.3].

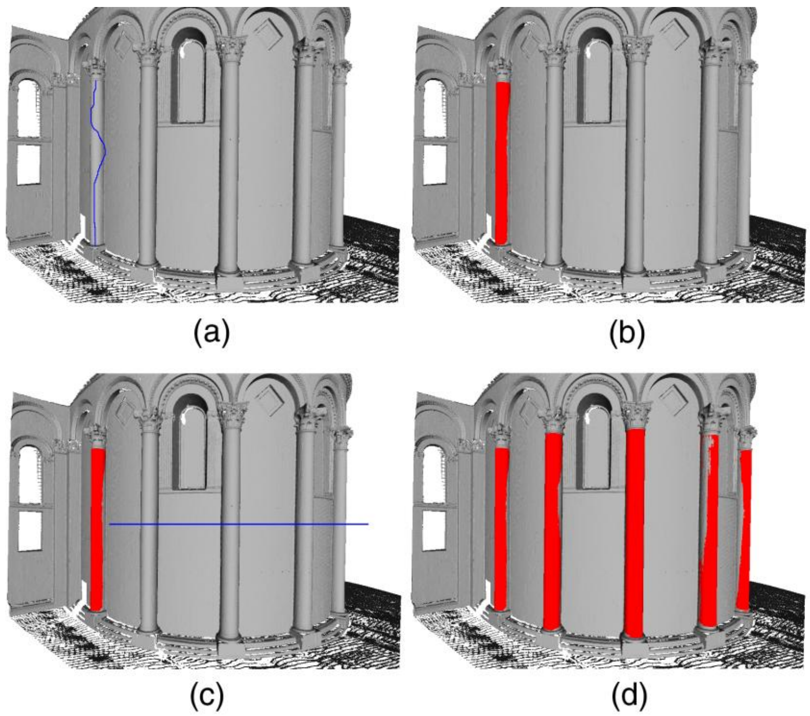


Figure 6. An example of sketch-based selection of a single column, fitting and selection of the other similar components.

CityEngine - Semantic LoD Rendering

In general, complex 3D models of cities and buildings (e.g. ancient Pompeii or Rome in the context of 3D-COFORM) need to be simplified in order to enable efficient rendering. The basic idea is to cull away details which are too small to be visible (below pixel size) or which are hidden and slow down transmission and rendering without contributing to the perceived images. Established mesh decimation techniques (see Figure 7) are usually not suitable for the building models relevant to 3D-COFORM as they (1) do not have the necessary topology (these models consist of many disconnected meshes) and (2) tend to destroy architecturally relevant building parts.

Figure7. Traditional mesh decimation algorithms tend to destroy architectural models. This example has been produced with a standard vertex clustering technique.

In Year 2 we have worked on two paradigms to tackle the level-of-detail challenge for procedurally generated city models:

1. Explicit creation of low-res polygonal models from hi-res models based on geometry, mesh hierarchy and semantic information.
2. Direct rendering of grammar-based model descriptions, generating as much detail as is visually needed.

<u>*Approach 1: Creation of low-res polygonal models:*</u>

The idea of the first approach is to use the geometrical, hierarchical and semantic information (Figure 8) produced by a grammar-based modelling system to segment the models and replace expensive parts with simpler primitives. This approach is currently described in a Master Thesis [T5.2.1]. While achieving a good segmentation and simplification based on meshes only turned out to be very hard, we successfully used the scene-graph information produced by the grammar to reorganize it into volume façade sub-trees. These sub-trees are then replaced by simple primitives used an image-based shape-matching with alignment optimization.