

INFORMATION LACK LOCALIZATION IN ARCHITECTURAL LASER SCANNING SURVEYS: THE VENICE TEST SITE

G. Salemi, I. D'Urso, V. Achilli, A. Bedin

Dipartimento di Architettura, Urbanistica e Rilevamento – Facoltà di Ingegneria, Università di Padova
Via Marzolo, 9 – 35131 Padova – Italy
giuseppe.salemi@unipd.it

Commission V, WG 4

KEY WORDS: Laser scanner, point cloud, Nurbs, mesh

ABSTRACT:

The laser scanning survey is one of the most updated and growing technique to acquire digital data for the architectural archives, including the cultural heritage applications and the archeological sites documentation. In the last years, many approaches have been proposed integrating digital photogrammetry and image processing algorithms. The laser scanner, sometimes, has been used only to produce something like “beautiful and good photos”, where the most prominent aspect is the big amount of digital data acquired at a big rate of spatial resolution. In this paper, the authors propose an approach based on the interpretation and the data mining applied to the laser scanning surveys performed at some historical buildings in Venice. This approach integrates the digital image filtering and the study about the lack of information among different digital data format.

1. INTRODUCTION

In the past years laser scanning has become more and more important to perform 3D scanning of regular and irregular surfaces, working at long-range, medium-range and close-range scale.

This paper will give an overview about some data export formats also in terms of memory storage and cpu-time. Problems which appeared during data translation are described and some interesting results about lack of information are presented using the Riegl RiScan software. This is able to export data using different formats: 3DD, VRML, VTK, DXF, PIF, OBJ, STL, TXT. In this paper, the export efficiency is analysed in order to evaluate after the data translation:

- the preserved quantity of data information;
- the file dimension;
- the access file times.

Four data formats have been studied, choosing them among the more useful and used for the three-dimensional objects visualization: VRML, DXF, STL, OBJ.

The text format was not taken in account, because it is a simple list of the points coordinates from the raw file and therefore a direct comparison was not possible with the other interchange formats. To perform this task the SolidView and the AutoCAD software have been used.

In these cases, the object is visualised as polygonal mesh (triangular mesh), that is the points of the points cloud with their position coordinates (x, y, z) are linked each other with segments to form plane faces. These approximate the original object form.

A short description of each examined format is reported, in order to understand the construction methodology of the 3D object.

2. DATA EXPORT FORMATS DESCRIPTION

2.1 VRML

VRML means “*Virtual Reality Modeling Language*” and it is a format to describe 3D objects, especially suited for Internet. It is a scenes description language and not a programming language. The VRML files are not compiled, because they are simple ASCII text files which are interpreted by the so called VRML browsers.

A VRML file is recognized by the interpreter program using a text string at the file bottom up:

```
#VRML V1.0 ascii
```

In the default reference system, the Y axis is on the top, the X axis is on right and the Z axis is on the user direction. The objects on the scene and the scene itself can be rotated using the axes orientation. When the object position is determined in the axes origin framework, it can be oriented using:

- yaw (rotation on the Y axis)
- pitch (rotation on the X axis)
- roll (rotation on the Z axis).

The three axes (X, Y, Z) and the three rotations (yaw, pitch, roll) are the six freedom degrees of the system: the position and the orientation of the objects in the space are determined by these six parameters.

In order to represent a 3D object, the computer uses a points data set to plot the form: because, is quite impossible to distinguish the form itself, this data set is called *points cloud*. Afterwards, the points are linked together using lines to form a *mesh*, which is composed by a polygons set (typically are triangles, because are the most simple to generate).

The polygons have only one face, therefore are visible from the object outside and invisible from object inside. After the polygon generation phase, the computer is able to create some shades to better render the three-dimensional aspect.

In a VRML file, the objects are called *nodes*; when they are hierarchically ordered they set up the scene representation.

Each VRML environment, even if it is very simple, is a scene representation.

There are three nodes types:

- form node, if it describes the real geometry;
- propriety node, if it modifies the geometry;
- group node, if it is able to modify objects groups like a single object.

The cumulative effect of the nodes on the scene representation is called *status*.

VRML uses the following measurements units:

- meter for the distances;
- radian for the angles;
- percent (as unit fraction) for the others quantities.

In the VRML language, there are some a priori defined forms, which are called *primitives*. The Cube, the Sphere, the Cone and the Cylinder are the form nodes which define the primitives.

On the surface of each form, a texture can be applied, coming from an external image file or from a VRML file definition like a set of coordinates.

As default, the objects of a VRML scene have the centre and the orientation coincident with those ones of the scene. To change the default values, the *transformation nodes* must be used, in order to have the translation, the rotation and the scale variation of the object. Furthermore, the default light source and the scene sight point can be moved.

2.2 DXF

DXF means “*Drawing Exchange Format*” and was created by the Autodesk in order to exchange drawing files information using a text language which could be controlled by a common user.

The DXF format is a representation of all the information encapsulated in a AutoCAD drawing file using *tagged data*.

Tagged data means that each data element is preceded by an integer number which is called *group code*, to indicate the data type which follows. It also indicates the data meaning for a specific object type or record.

A file coming from a CAD program, generally speaking, is a database. On the top level there are some information which aren't database records; they are in the *header* section of the file and collect more data about the file version.

After, there some information about the framework variables and all the “symbols” which will be used in the principal section of the file; for example, the definition of “text style”, “elevation style”, “lines types” and “blocks”.

The most heavy section is that one related to the objects records. Here, there are the graphic elements of the drawing, with all the related information; for example:

- line: start point, end point, line type, colour, ...;
- block: insert point, rotation angle and the x, y, z scales.

DXF is the interchange format developed by the Autodesk with the copyright, but the specifies are public domain; at the present is one of the most secure approach to convert data among different formats.

2.3 WaveFront (OBJ)

The OBJ WaveFront format can describe three-dimensional objects in text files using a list of polygonal mesh, where each one has one face and one related index.

This format is often used to transfer 3D objects among different programs, like Alias or 3Dstudio.

The polygons belonging to the same objects are exported as single groups and the group name will coincide with the name of the primary object. In the file, the name of the group is characterized by the letter *g*.

It is impossible to maintain the information about the hierarchy among the objects; the tip is the exportation in the same group.

The object file format is ASCII. It supports both polygonal objects and more complex forms (*free form*).

The polygonal geometry uses points, lines and faces to define the objects, whereas the *free form geometry* uses curves and surfaces.

The vertex related data are representable using four modes, one for each type of vertex coordinates:

- geometric vertex: *v*;
- texture vertex: *vt*;
- normal to vertex: *vn*;
- vertex which indicate the spatial parameters (only for *free-form* objects): *vp*.

The following example shows a small part of an OBJ file, with the four type of information about the vertices:

```

v -6.000000 6.000000 0.000000
v -6.000000 -6.000000 0.000000
v 6.000000 -6.000000 0.000000
v 6.000000 6.000000 0.000000
vt -6.000000 6.000000 0.000000
vt -6.000000 -6.000000 0.000000
vt 6.000000 -6.000000 0.000000
vt 6.000000 6.000000 0.000000
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
vp 0.210000 3.590000
vp 0.000000 0.000000
vp 1.000000 0.000000
vp 0.500000 0.500000

```

2.4 StereoLithography (STL)

The STL or StereoLithography format is an ASCII or binary file: it is a list of triangular surfaces which describe a computer generated 3D model.

The STL text file must be open and closed with the key words *solid* and *endsolid*. Between these two words there is a list of all the triangles that define the 3D model faces.

The definition of each triangle includes one normal vector in the direction of the outside of the object surface and the x, y, z components for each three vertex. Therefore, twelve parameters are need to define one triangle.

These values are related to a Cartesian reference system and are expressed by a rational number. The values related to triangles must be positive.

The following example shows the definition in ASCII format of one single triangle belonging to an STL file:

```

solid
...
facet normal 0.000001 0.0
outer loop
vertex 2.00 2.00 0.00
vertex -1.00 1.00 0.00
vertex 0.00 -1.00 0.00
endloop
endfacet
...
endsolid
    
```

The most common error in a STL file is related the rule vertex-vertex: this rule assumes that all the adjacent triangles must have two common vertices. In the Figure 1 (at left), one triangle with four vertices is shown. The external vertices are not shared with other triangles, whereas each lower triangle has one non shared vertex. Using the above rule, the upper triangle must be divided as in figure (right triangle).

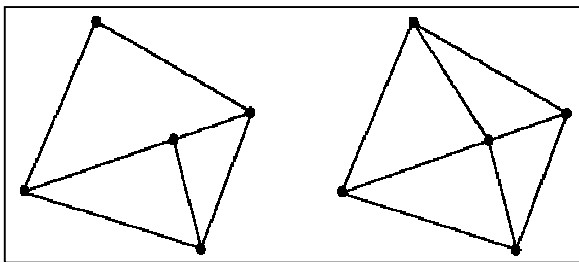


Figure 1. Example of rule vertex-vertex validation (see text)

3. EXPORT DATA BENCHMARK

Venice has been a good test site to set up the laser scanner instrumentation. More and more images were acquired changing the operative modes, as the scanning speed and the angular deflection (both for the “line scan” as for the “frame scan”).

Two files acquired with the Riegl LMS-Z210 laser scanner are presented; using the 3D RiSCAN software, the files have been converted from the original 3DD format to the four previous described 3D formats.

3.1 Surian Bellotto Building: Scan09

This file in real colours, represents the central balcony of the Surian Bellotto Building front (Figure 2); the file is 20 kB long in the original 3DD format.



Figure 2. Front of the Surian Bellotto Building, Venice

The image definition it isn't so good; in fact, it is possible to distinguish the pixels. Each of them has four information: the position in the framework of the instrument reference system, the distance from the acquisition instrument, the reflectance and the colour.

The files coming from the exportation phase, have been visualized with the SolidView software with these options:

- the light is on front to determine different shading of the mesh faces which have different slopes;
- the observer is on a plane that is parallel to that one where the object is lying;
- the triangular meshes have the contours evidenced.

In the Figure 3, the results are showed, using VRML, DXF, OBJ and STL formats.

3.1.1 VRML file

The VRML file needs 147 kB of storage memory and it is composed by 2942 triangles; in this format the colour and reflectance information are not conserved.

From 3D-RiScan the data are exported as points, lines or faces; SolidView is able to read the exported file only as faces file .A not negligible percentage of position data are lost: the 8.7% of the image surface isn't represented in this format.

The lack of information is more evident in the upper part of the image: nearly an entire horizontal line is not acquired.

This a feature of this format and it isn't present in the other translations.

Other lacks are in the lower part of the image, near the balcony and the windows.

The light is quite homogenous, except in some zones where the shade seems to indicate the form of some windows.

3.1.2 DXF file

The DXF file needs 347 kB of storage memory and also it has 2942 triangles. The memory storage is increased; must be noted also that the raw data (3DD format) needs only 20 kB of memory. To read the data as points or lines, another kind of software, like AutoCAD, must be used.

The percentage of surface without information is about 3.9%; this is the lowest among the four exportation files.

The lack of information is on the lower part of the image, near the balcony and the windows (see also the OBJ and STL files).

The shadows indicate the arches of two big central windows and of two small lateral windows.

3.1.3 OBJ file

The OBJ WaveFront file uses 95 kB of storage memory and it also has 2942 triangles. The percentage of surface without information is consistent: 8.2%, which is lower than the VRML format.

Some small vertical areas are showed as lack of information in the lower part of the image; this is a feature of this format and of STL too. A vertical line of no-information is also present as depicted in the Figure 4. The object colours are not conserved in this format. The windows profiles are clearly visible, using also a darker colour which is related to different lighting conditions.

3.1.4 STL file

The STL Stereolithography file uses 524 kB of storage memory and it also has 2942 triangles. This format uses more memory space in respect to the other ones and therefore needs of more time to access the file itself (read and write operations). The percentage of information lack is 8.2%, like for the OBJ file. The visual aspect of these two files is quite similar; for this image the two files are perfectly coincident.

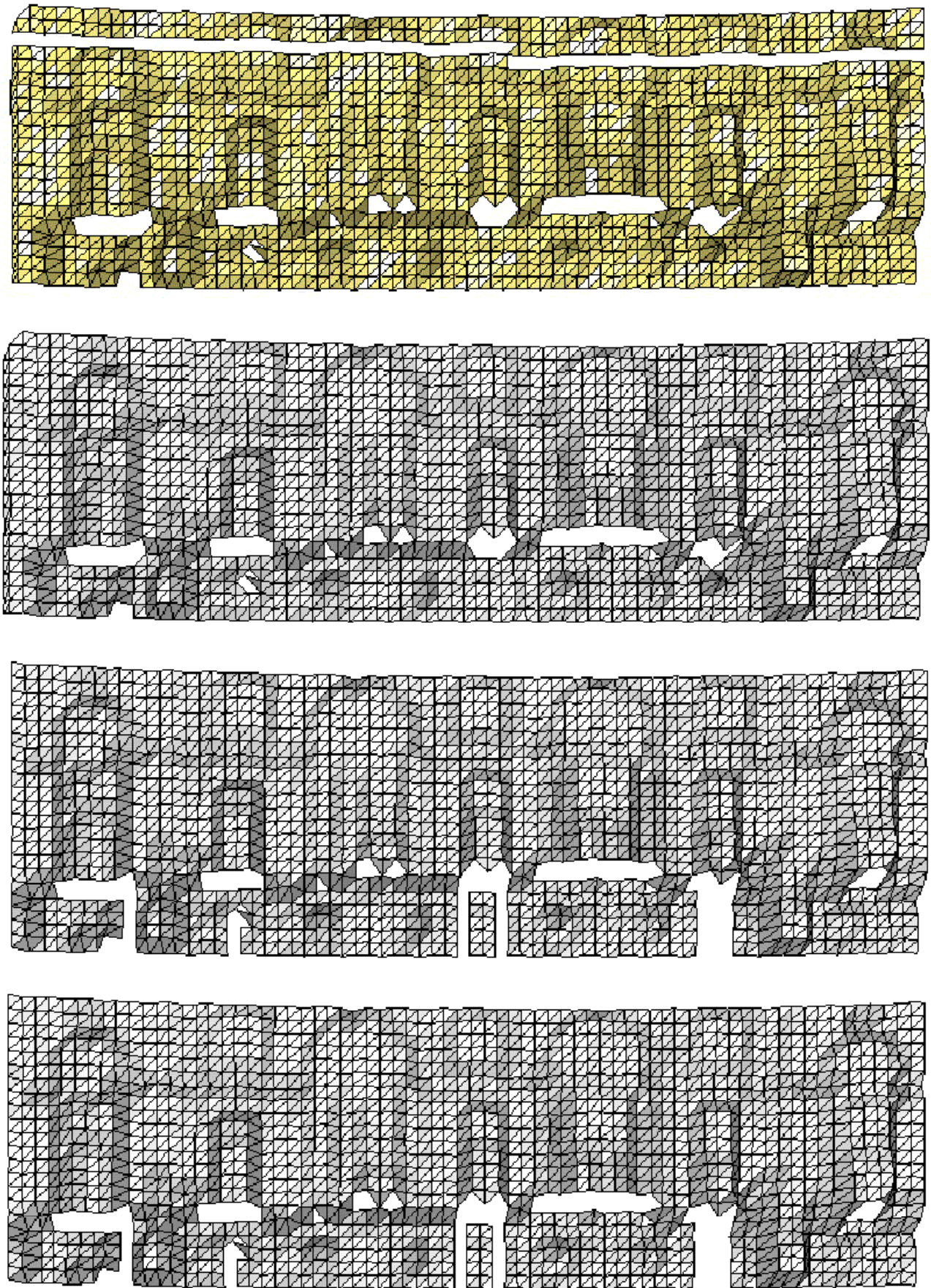


Figure 3. The image from Surian Bellotto Building in the VRML, DXF, OBJ and STL export formats (from the top to the bottom)

3.1.5 Difference file: Scan 09

The Figure 4 shows the difference file for the image Scan09. Using four colours are depicted the “information holes” for each export data format.

The red colour localizes the areas which are, in terms of no data representations, the same in the four formats. They are in the lower part of the image near the windows.

The green colour shows the areas with no data in the OBJ and STL formats; a long vertical line (at the left of the image) must be observed.

The yellow color represents the zones with no data in the VRML format, which has the most prominent percentage of information lack. This format is characterized by an horizontal trend in the “information holes”.

The blue colour is common at OBJ, STL and WRL formats; it is localized at the cross point of the two major lines of no-data: the green vertical one and the yellow horizontal one.



Figure 4. Difference file for Scan09 in false colours

3.2 Detail of a Venetian building front: Scan12

The second test is related to a small part of a Venetian building front. The 3DD raw data file uses 24 kB of storage memory. The surfaces trend is quite simple, because there aren't balconies or concave details as in the previous image.

On this file, two filters have been applied to better appreciate small distance and reflectance variations:

- the first one is an intensity filter, which imposes a reflectance range between 70 and 138 (the previous values were 13 and 138);
- the second one is a range filter, which imposes a distance range between 28 m and 30 m (the previous values were 7.7 m and 29.7 m).

This approach was very useful to filter one object localized at about 8 m from the instrument during the acquisition phase vs a medium distance of about 30 m.

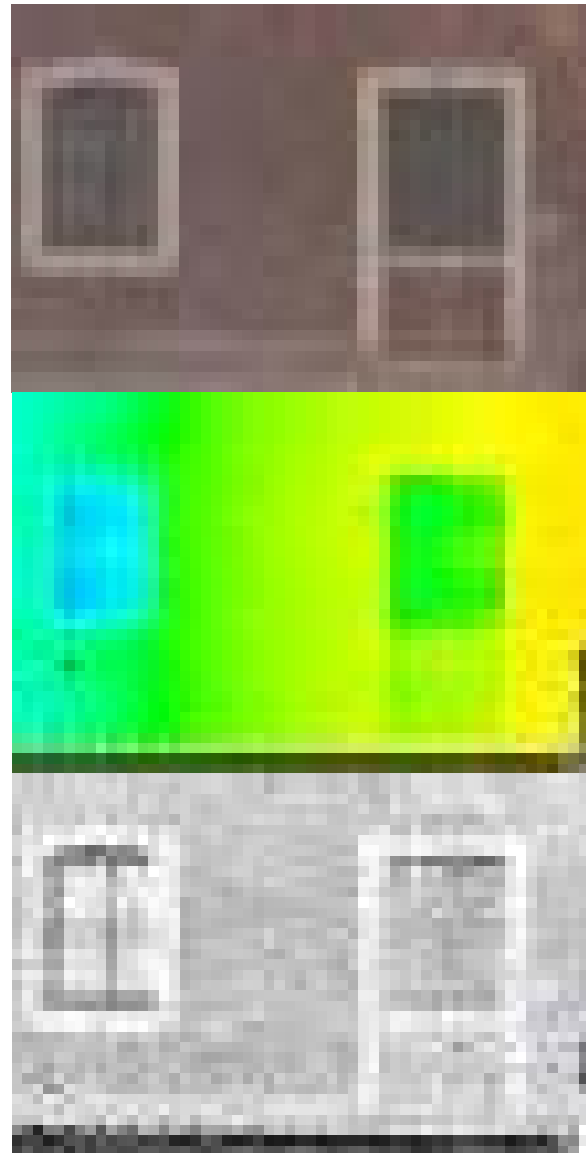


Figure 5. Detail of a Venetian building front: 3DD image in real colours, after the intensity filter

The VRML file uses 182 kB of memory and has 3649 triangles. The translation was made using the option faces, as requested by the SolidView program (Figure 6).

The percentage of surface without data is 3.8%, the highest in this test among the different formats.

Also in this case, there is an horizontal line without data; there is also a small “hole” in the lower part of the image. The profiles of the windows are clearly visible.

The DXF file needs of 555kB of memory and uses 3649 triangles. The percentage of “no data” is very low: 0.1% and it is the lowest among the examined formats. Only one “hole” it is present under the window profile at right in the image.

Also for this image, the OBJ and STL files are coincident; both have as a 0.9% of “no data” and both have 3649 triangles with the vertical trend of information lack under the window profile (Figure 7).

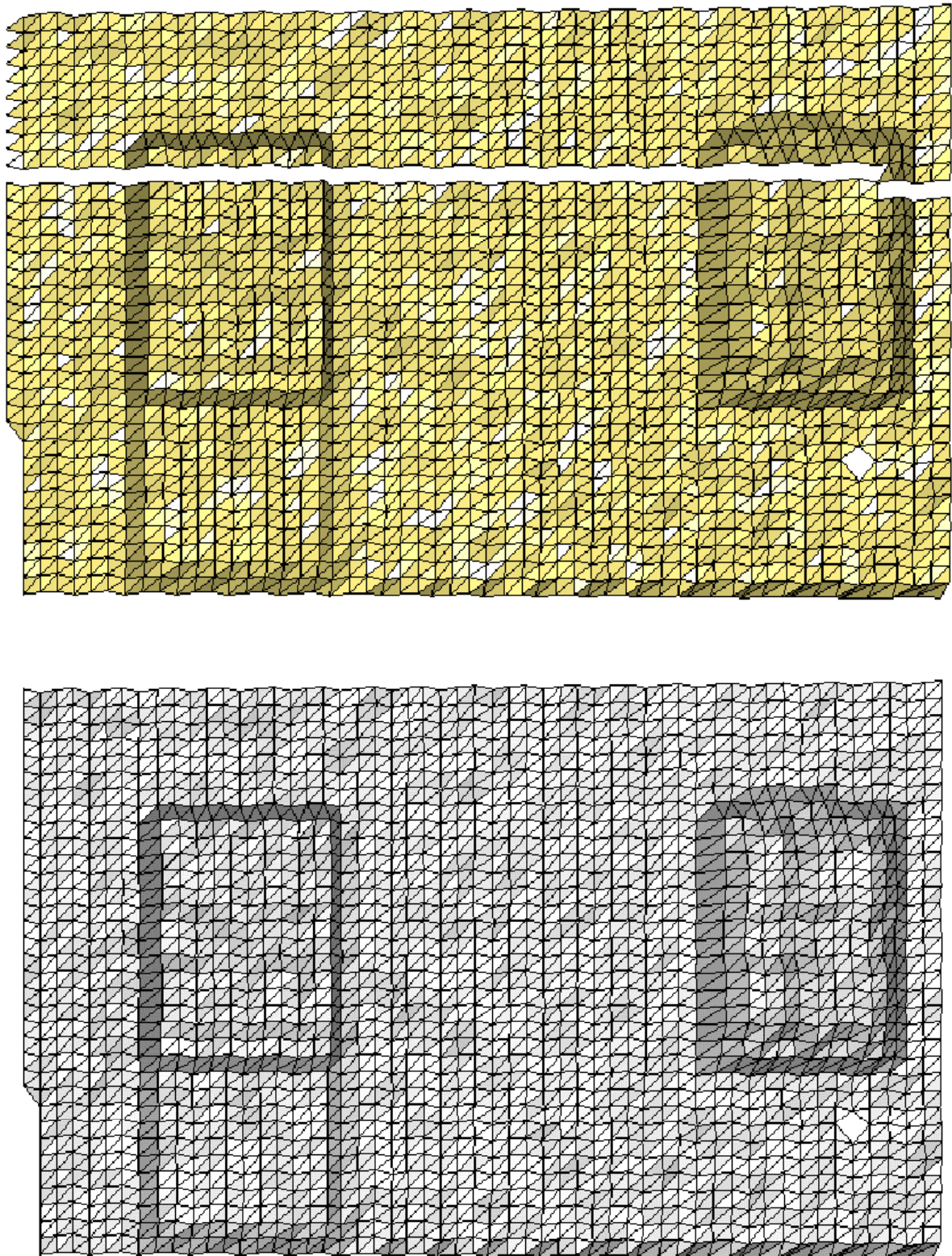


Figure 6. The image of Scan12 translated in VRML (top) and DXF (bottom)

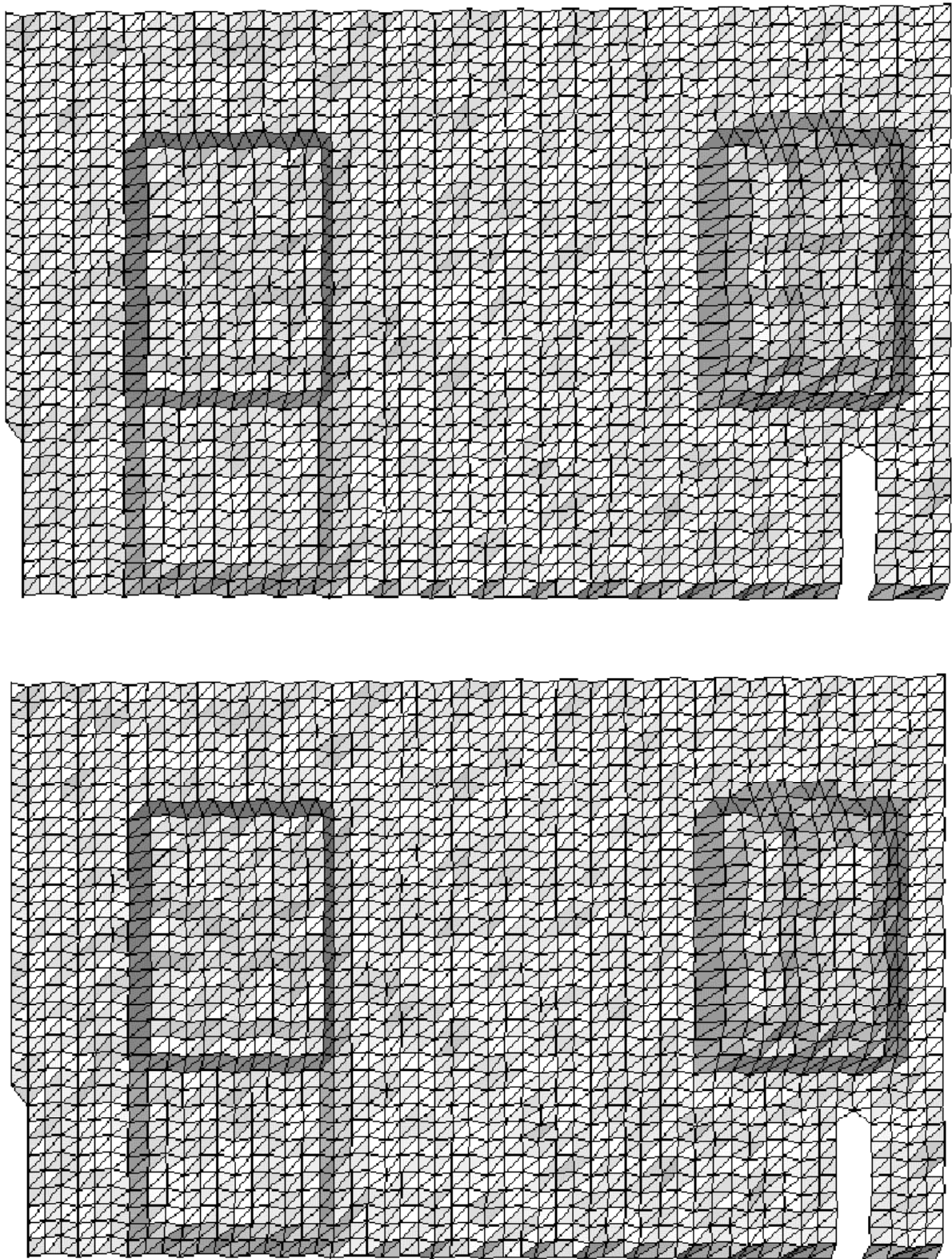


Figure 7. The image of Scan12 translated in OBJ (top) and STL (bottom)

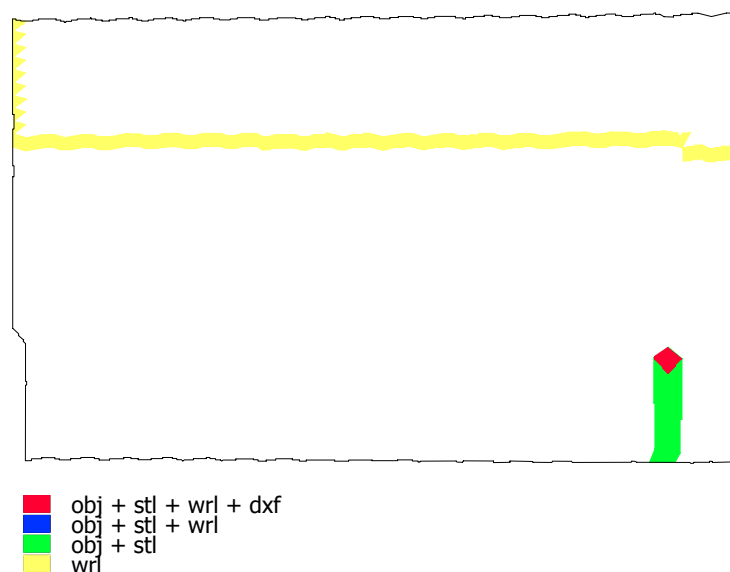


Figure 8. Difference map for Scan 12

4. CONCLUSIONS

Some facts must be noted about this experiment related to some exportation data formats.

The topics are the surface coverage and the files dimensions. About the first one, a lack of information is present after the translation phase from the original 3DD format to the new ones. The DXF format is the best one from this point of view with a lack of 3.9% and 0.1%; the VRML is the worse one (8.7% and 3.8%) and the OBJ and STL present a larger number of "information holes" with a minor percentage (8.2% and 0.9%). Another interesting fact is about the spatial trend in the lack of information: horizontal for DXF and VRML, vertical for OBJ and STL (Figures 4 and 8).

For the second topic, the increment in file dimensions must be stressed out: the variation (for the first example) is from 20 kb in the 3DD raw data format up to 524 kB for STL format. OBJ is the best one: it requires an increment by a factor 5 vs. VRML (factor 7) and DXF (factor 22).

In this sense, the increment in the files dimensions must be taken in account for very large images; for example, an image of 2500 kB was acquired in Venice producing long times for accessing the data in STL and DXF formats.

In the next future, the authors will investigate others data formats in order to establish the good compromise among surface coverage, access times and files dimensions. Of course an effort must be taken in account to avoid the lack of information as the colour and the reflectance and nevertheless more performant 3D viewer are needed, also for the requirements of the Internet browsers.

5. SELECTED BIBLIOGRAPHY

- Bern, M., and Eppstein D., 1992. Mesh Generation and Optimal Triangulation. Technical Report P9200047, Xerox Palo Alto Research Center.
- Boissonnat, J.D., 1984. Geometric Structures for Three-Dimensional Shape Representation. ACM Transactions on Graphics, Vol. 3, No. 4, pp. 266--286.
- Chen, Y., and Medioni, G., 1992. Object Modelling by Registration of Multiple Range Images. Image and Vision Computing, Vol. 10, No. 3, pp. 145--155.
- Edelsbrunner, H., and Mücke, E. P., 1992. Threedimensional Alpha Shapes. Proceedings of the 1992 Workshop on Volume Visualization, Boston, pp. 75--82.
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W., 1992. Surface Reconstruction from Unorganized Points. Computer Graphics, Vol. 26, No. 2, pp. 71--78.
- Schroeder, W.J., Zarge, J.A., and Lorensen, W.E., 1992. Decimation of Triangle Meshes. Computer Graphics, Vol. 26, No. 2, pp. 65--70.
- Turk, G., 1992. ReTiling Polygonal Surfaces. Computer Graphics, Vol. 26, No. 2, pp. 55--64.