

DISTANCE-VALUE-ADDED PANORAMIC IMAGES AS THE BASE DATA MODEL FOR 3D-GIS

Panoramic Photogrammetry Workshop - Berlin Germany, 24-25 February 2005

Edward Verbree¹⁾, Sisi Zlatanova¹⁾ and Sander Dijkman²⁾

¹⁾ Delft University of Technology, OTB Research Institute for Housing, Urban and Mobility Studies, Section GIS Technology
Jaffalaan 9, 2628BX Delft, The Netherlands
e.verbree@otb.tudelft.nl; s.zlatanova@otb.tudelft.nl

²⁾ TerraImaging
Koningslaan 35, 1075 AB Amsterdam, The Netherlands
s.dijkman@terraimaging.nl

KEY WORDS: Panoramic Images, Laser Altimetry, Data Processing, and Visualization

ABSTRACT:

Panoramic images portray a surround view of the real world in one image. This kind of presentation however does, alike regular pictures, not give any real depth information. Depth related relationships are only to be detected (within one image) by psychological cues like relative size, linear perspective and shadow. The relative distance between two features can only be retrieved by 1) stereoscopic measurement within two pictures or 2) by the integration of terrestrial laser scanning systems. In the second approach the photo is enhanced with information about the distance between each pixel and the location of image recording. If we visualize this cloud of 'distance pixels' from a point of view chosen at the recording place through a panoramic perspective it will give us the impression of the original panoramic picture, but now with the added value of depth-related queries. This kind of 'distance-value-added' panoramic pictures can be used as a base data model for 3D-GIS visualizations. This paper concentrates on the database organization of RGB laser scan point clouds, the creation of virtual 'distance-value-added' panoramic images from these clouds, and the visualization and spatial analysis based on this kind of perception rich images.

1. INTRODUCTION

Panoramic images are defined to be 'distance-value-added' when each pixel holds the distance to the recording location. This distance can be obtained in two ways. The first one is by attaching the distance to each pixel of the image. This distance can be obtained by the geometrical registration of a 360° laser scanning taken from the same recording position. The second possibility is to attach the RGB-value to each laser scan point.



Figure 1: Picture of hallway

If we observe this kind of a laser scan point cloud from the recording location, it will give the impression of an image. Compare Figure 1 (picture of hallway) and Figure 2 (laser scan point cloud of the same hallway).



Figure 2: RGB laser scan point cloud of hallway

Adjusting the appropriate parameters settings from central to panoramic perspective the laser scan will even look like a panoramic image.

Furthermore, such 'distance-value-added' panoramic images allow the creation of virtual panoramic images. It is feasible to create an image positioned at an arbitrary point, observing a

point cloud obtained by a combination of several, not necessary 360° RGB-laser scans. Selecting a new virtual viewing point will require filtering of the points to be included in the panoramic image. This is to say, points hidden by walls and other obstruction in the scene (due to the new viewing position) should be excluded. As these walls are obtained by the laser scan points themselves, we could detect them by a face-fitting algorithm.

This paper presents an approach for creating distance-value-added panoramic images from segmented point cloud. Through this kind of presentations we could support three-dimensional visualizations as they offer psychological cues like relative size, linear perspective and shadow (Lynn, 2003), in conjunction to geometrical measures supported by the attached depth-value on each element in the image.

The paper is organized in three general parts. Section 2 describes an approach to obtain a virtual 'distance-value-added' panoramic image out of this RGB-laser scans. Section 3 analyses the spatial database organization of laser scan points and the corresponding color information obtained from (panoramic) pictures in a geo-database management system (geo-DBMS). Section 4 will focus on deriving panoramic images from the point clouds stored within a Geo-DBMS. The final section 5 summarizes the possibilities and outlines further research.

2. DERIVING VIRTUAL 'DISTANCE-VALUE-ADDED' PANORAMIC IMAGES

2.1 Laser scanners and panoramic images

A terrestrial laserscanner produces a cloud of reflected points. The spatial resolution, i.e. number of points scanned, depends on the field of view and the sampling rate. The Riegl LMS-Z210i, for example, offers a field of view of 360° horizontal and 60° vertical. It is possible to mount a color camera on top of the scanner that - after a proper registration - will result in a colored point cloud at the end of a measurement cycle (Fröhlich, 2004).

The main goal of these 'colored point clouds' is to give a visual feedback of the scanned area. It will help to identify tie points that need to be scanned with a high spatial resolution, and it will indicate areas that have to be re-scanned, as for example people are walking by during the scanning.

The field of view of a standard camera is by far smaller than the laserscanner and the camera has to be positioned in several directions in order to cover the same region as the laser. The Cyrax HDS300, with a field of view of $360^\circ \times 270^\circ$ needs 111 individual images of 1024×1024 pixels for one scan (Jansa, 2004).

One way to improve that limitation is to use a panoramic camera to obtain an image with the same field of view as the scanner in one step. There exist a wide range of panoramic cameras (Luhmann, 2004), i.e. analog swing lens, rotating and spherical fish-eye cameras. As these cameras are stand-alone devices, a complex registration of the RGB-values of the digitized and rectified photos to the point cloud is needed. Digital line scanners, like the EYESCAN used by (Haala, 2004) offer a panoramic spatial resolution up to 10.200 (row) pixels by 43868 (column) pixels. With a clever set up, i.e. the recording of both the laserscan and the panoramic picture at the same position the co-registration process of both data sets can

be simplified considerably. But as both recording will take a considerably amount of time, temporal differences will occur.

Another possibility is to use laserscanners in combination with a digital camera that can be calibrated and oriented by standard methods in order to provide the necessary geometric information for a precise resampling (Luhmann, 2004). The camera will be mounted at the laserscanner such that the rotation axis passes the principal point of the lens, as for example the Riegl LMS-Z210i, or more or less integrated within the Leica / Cyrax 2500 scanner. This scanner has a field-of-view of only 40° by 40° , but it will provide true-color images simultaneous.

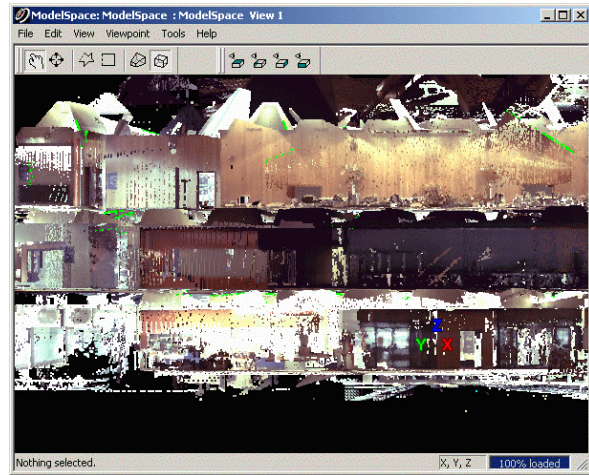


Figure 3: Visualization of Aula point cloud in Cyclon

This kind of limited field-of-view scans, but also panoramic scans, will only cover a restricted area of the area of interest. The scans have to be adjusted and linked together to one covering dataset. A neat example of this process is given in Figure 3, where 237 individual scans of the interior of the Aula of Delft University (see Figure 4) are combined to one point cloud of more than 25 million points (Meijers, 2005).



Figure 4: Exterior of Aula Delft University of Technology

2.2 Distance-value-added images

In this paper, we present another way in acquiring 'distance-value-added' panoramic images, as we call the 'colored point clouds' to emphasize the value of having both the RGB-values and the distance of the scanned points to the observation point.

Furthermore our approach makes use of the vector geometry (edges, polygons) derived from the point cloud.

Within most photogrammetric and laseraltmetric applications and in research the cloud of points, the set of (panoramic) images, or a combination of both (Jansa, 2004) is used only as the input to reconstruct the 3D objects measured. The point cloud is discarded when the walls, roofs, trees or any other object of interest are derived. In some cases the color information of the images is used for texture mapping, but no actual linkage between the derived objects and the original laser points is kept.

That is a pity, because most of the time the derived faces and edges do not represent the measured object topological correct. Edges are not connected, faces are loose and intersect with each other. A lot of work has to be performed to construct a valid representation of the object. And the final 3D-model will not give the idea or the impression as the panorama's or colored point clouds gave when visualized.

The derived edges and faces can be used however to filter the original point cloud. Figure 5 shows an example of 50 randomly chosen faces that are colored according to the original laser scan points. Points where the distance to the nearest face is exceeds a certain threshold can be considered as outliers and can be removed. The points 'at' a face will, caused by the non-optimal measuring precision of the laser, give a coarse appearance that now can be adjusted according the derived face.

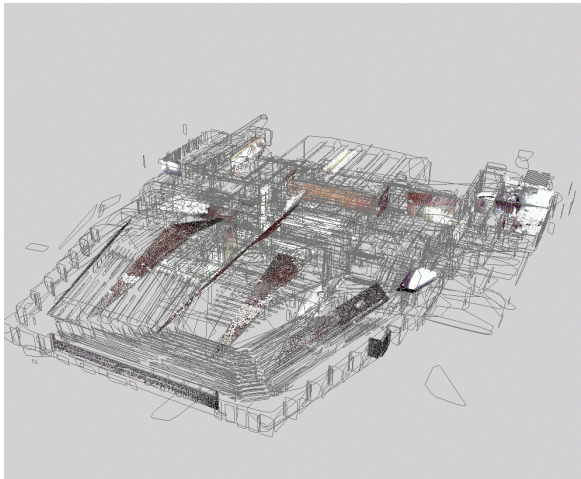


Figure 5: 3D Model of Aula with some faces colored by original laser scan points

But, more important, we can use the faces of the model to determine which points are visible or not visible from a certain observation position.

Some neat examples exist of laserscan point clouds that give the impression of being a real photograph, even if the color mapping is not based on the actual RGB values, but on the reflection value. These kinds of visualizations are only possible when we observe the colored point cloud acquired by one scan from the recording position. Each scan point within the portrayal is in reality also visible, otherwise it won't be scanned anyway.

If we choose an observation point at another place than a recording position this property doesn't hold anymore. Non-visible points will show up in the laserscan image when we

combine two or more (panoramic) laserscans in one scene. Therefore we have to determine which points are visible and which ones are obstructed from the observation point.

This calculation can be done in two steps. First a hidden face removal algorithm is applied and then all points connected to these faces are discarded from the scene. And that gives us the possibility to create colored point cloud visualization from each desired observation location. This kind of visualization can be in central perspective, but also as a panorama. This last example we will call a virtual 'distance-value-added' panoramic image.

3. SPATIAL DATABASES FOR RGB-LASER SCANS

3.1 Geo-DBMS versus file organization

Depending on the laser scanner (see section 2.1) the number of points in one scan can vary from few hundreds points to millions points. These points are usually organized in a file format, which is specific per scanning software. This file is optimized for fast upload into the memory and (in case of insufficient memory) smart algorithms for disk swapping.

Such software is very convenient for 3D modeling of relatively small areas or simple objects. A large point cloud (more than 20 million points) causes drastic delay in the systems (despite the optimizations). Furthermore, the process of 3D modeling is mostly manual. For example to initiate a plane fitting through group of points, the user has to select (point to and click on) at least one point in the point cloud. If the scanned object is very complex, it becomes almost impossible for the user to decide which objects are significant for modeling. Finally, this software cannot be used to create distance-value-added panoramic images as described in the previous section.

To avoid these major problems we propose the organization of the point cloud in a geo-DBMS. Currently most of the large DBMS offer support of spatial data types, amongst which points. Most of the spatial data types are defined according to the Open Geospatial Consortium Specifications.

The apparent advantages of storing points in a DBMS are:

- Standard format (i.e. schema) that can be 'understand' by different front-ends (GIS, CAD and other software)
- Possibilities for spatial indexing
- Possibilities for attaching other information to the points
- Spatial queries, which allow only a limited set of the data to be used at a certain moment.
- The points can be accessed and modified by different users and the modifications are readily available for everyone.

Besides the advantages listed above, clearly, there are a lot of reasons to use geo-DBMSs: control on shared data and crash recovery, automatic locks of single objects while using database transactions, advanced database protocol mechanisms to prevent the loss of data, data security, data integrity and operations that comfortably retrieve, insert and update data.

To import the data into a DBMS, the point cloud has to be first exported. The software for visualization and processing laser points offer usually several formats for data exchange. For example, Cyclon supports ASCII export formats containing the 3D coordinates of the laser scan points, intensity and the RGB values recorded at this point. The file format exists in two variants: with and without indication of individual scans.

The structure is trivial, i.e. x,y,z, intensity, red, green, blue:

```

62308 --number points in first scan
20.325 29.380 11.512 -383 25 6 25
21.393 29.841 10.543 -394 25 6 25
21.436 29.875 9.990 -357 25 6 25
...
856992 --number points in second scan,
etc.
21.371 29.840 9.912 -384 25 6 25
21.404 29.855 9.919 -386 25 6 25
....

```

Since the tests in the paper are performed in Oracle Spatial 10g, a short description of the Oracle Spatial object-relational model is given.

3.2 Organization of point cloud in Oracle Spatial 10g

In general, the object-relational model of Oracle Spatial differs from the OGC SQL specifications. Separate data types for point, line, and polygon do not exist but they still can be distinguished on the basis of additional parameters. The name of the spatial data type is SDO_GEOMETRY. It is defined as a complex object on top of the relational database:

```

CREATE TYPE sdo_geometry AS OBJECT (
  SDO_GTYPE NUMBER,          -- type and dimension
  SDO_SRID NUMBER,           -- coordinate system
  SDO_POINT SDO_POINT_TYPE,  -- defines point
  DO_ELEM_INFO MDSYS.SDO_ELEM_INFO_ARRAY, --
  interpretation of coordinates
  SDO_ORDINATES MDSYS.SDO_ORDINATE_ARRAY list of
  coordinates
);

```

All the parameters in the description have specific values, a combination of which defines a particular spatial type. For example, GTYPE is used to distinguish between the type of geometry and its dimension. GTYPE 3001 and 3003 mean that the data types are respectively 3D point and 3D polygon. An example of 3D point follows:

```

sdo_geometry (
  sdo_gtype = 3001,
  sdo_srid = 90112,
  sdo_point = (13.5,19.5, 2.33),
  sdo_elem_info = (NULL),
  sdo_ordinates = (NULL)
);

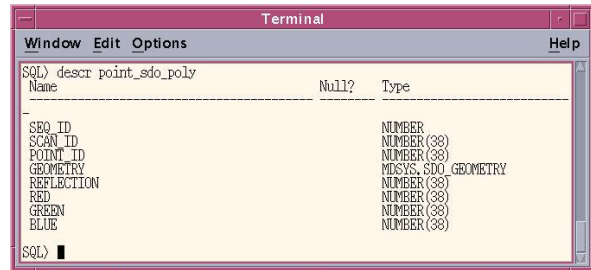
```

Once created the tables containing geometry have to be indexed (many of the operations work only with spatially indexed tables) and registered in the user metadata table. The metadata table is the first table to be read by the front-end applications. It provides information related to the names of the columns containing geometry and the range of the data stored in the tables. More information about the data types and spatial functions of Oracle Spatial is given in Stoter and Zlatanova, 2003

Laser scan points can be recorded in the Oracle Spatial point type using two approaches. Following the first one, the ASCII file format is imported in a table with the same structure. The coordinates are registered as floats in separate columns. A PL/SQL script can be further used to convert the relational representation to the SDO_GEOMETRY format. The second approach is direct record in the Oracle Spatial data type format. It is much more efficient if large data sets are considered. In case of a point cloud larger than 20 million points (as in the case study with the Aula), load in the database can last 1-2 hours

(depending on other processes running at that moment). The first approach (first floats and then record in point data type) requires almost twice as much time compared to the first one. The indexing of large point cloud always takes a lot of time (between 10-12) hours.

Figure 6 shows an example of point cloud organization using the SDO_GEOMETRY type. SEQ_ID is the sequential number of the points, SCAN_ID stands for the ID of the scan a point belongs to and the POINTS_ID is the ID of the point within a particular scan.



Name	Null?	Type
SEQ_ID		NUMBER(38)
SCAN_ID		NUMBER(38)
POINT_ID		NUMBER(38)
GEOMETRY		MDSYS.SDO_GEOMETRY
REFLECTION		NUMBER(38)
RED		NUMBER(38)
GREEN		NUMBER(38)
BLUE		NUMBER(38)

Figure 6: Point cloud organization in Oracle Spatial 10g

Once stored as point data types, laser scan points can be accessed by Microstation Geographics and ArcGIS. Discussion on the required operations is provided in Stoter and Zlatanova 2003, and Zlatanova et al 2002.

Point cloud stored in the database may bring some advantages in creating panoramic images since the points can be processed per scan (using the scan_id reference). Still the information is not sufficient because the points in one scan usually belong to several objects (walls, tables, chairs). Although the faces are well visible as separate objects there is no way to distinguish automatically between them.

3.3 Organization of polygons from segmented points

The next step we have applied is automatic segmentation and plane fitting through a set of segmented points. The segmentation is performed in software developed by Section FRS, Aerospace Engineering, Delft University of Technology. Vosselman et al., 2004 discussed in detail the applied approach.

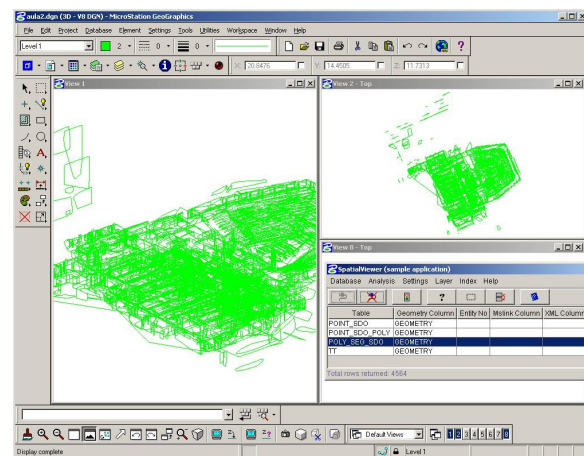


Figure 7: Convex polygons created after point segmentation and plane-fitting procedure.

For the scope of this paper it is only important to mention that segmentation of a point cloud is done using proximity criteria of

points and/or similarity of locally estimated surface normals. See Figure 7 for an example of this plane-fitting algorithm applied on the Aula-dataset.

Since most of the objects of interest in a building can be approximated with planes, only the plane-fitting algorithm is used. Knowing the parameters of the plane and the points belonging to one segment, a convex polygon is computed. A threshold is used to avoid keeping very small segments/polygons. The ID of the polygon is stored as additional information in the point table.

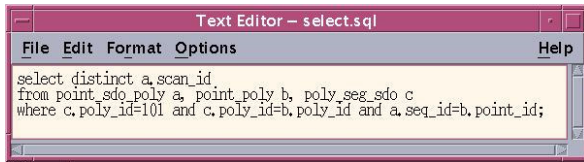


Figure 8: SQL query retrieving all the points that belong to a polygon 101

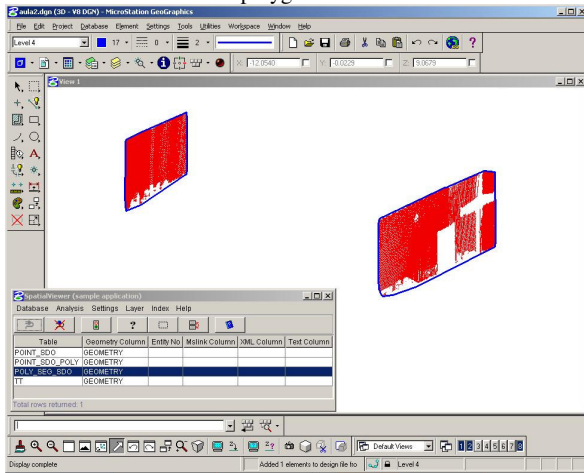


Figure 9: Visualization of two polygons with the laser scans points belonging to them.

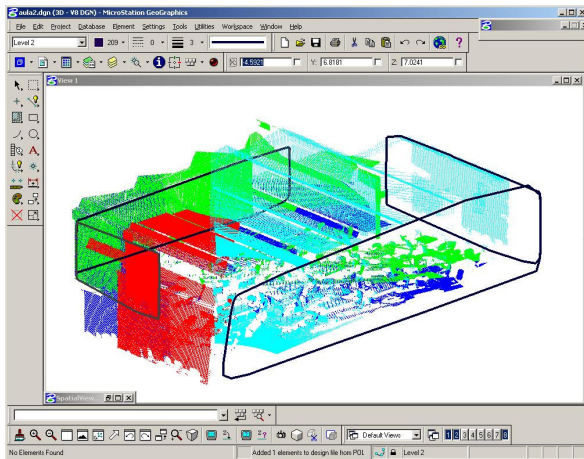


Figure 10: Scan points in one room.

The link to the polygons ID provides the needed information to distinguish between individual objects, see Figure 8 for an example of this kind of query.

Using a combination of queries with respect to the scan_id and poly_id, it is possible to obtain:

- All points that belong to a particular object, e.g. a wall, see Figure 9.
- All points that are recorded in one room, see Figure 10.
- All points belonging only to the walls in a particular room

4. DERIVING PANORAMIC IMAGES FROM POINT CLOUDS STORED WITHIN A GEO-DBMS

Some of the figures included in this paper, i.e. Figure 11 are produced by the 3D Photo software provided by TerraImaging.



Figure 11: Distance-added-value image of interior of Aula

Although this software is not yet capable of mapping the data into a panoramic projection, it is dedicated to the visualization of large RGB-colored point clouds in real time. High performance is reached by a dedicated rendering mechanism of the videocard supported by OpenGL. A Z-buffer algorithm decides which points are to be shown. This method works fine for visualizations of aerial laserscan, as the viewpoint is far above the scene, see Figure 12.

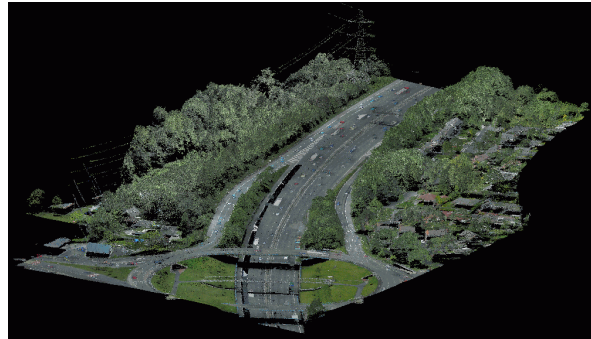


Figure 12: Aerial laserscan visualization

If we choose a viewpoint more within the scene relying on the videocard will result in points glancing through, resulting in stunning, but non-realistic images, like Figure 13. And moreover, it is somewhat strange to supply 20 million or more points to be rendered, when the display is limited to a resolution of only 1024 x 800 pixels.



Figure 13: See-through image of point cloud of Aula

To prevent that situation and to create realistic distance-value-added images we need a proper organization of the large amount of separate RGB-laserscan points within the geo-database, and a sufficient query mechanism to extract the points to be visualized. As we want to choose our observation point anywhere in the scene, it should be made possible to select the 'visible' scan points through an efficient query on the geo-database.

We assume that all laser points are segmented to derive the faces they belong to, deleting the points too far away from a face. If this relation is preserved in the database it can be used to select the laser scan points belonging to a certain face (see section 2), but also to exclude the points not visible because other faces obscure the face it belongs to. Figure 14 shows an example of Field-of-View query implemented at database level.

The linkage between the laser points and the facets they belong to is quite important as current geo-DBMS do offer support for 3D data storage, indexing and querying to a certain extent. A selection on all points inside a certain 3D bounding box performs very well, but the retrieval of the points seen from a certain observation point and taken into account obscuring facets is not yet, and will not, be supplied within the geo-DBMS core.

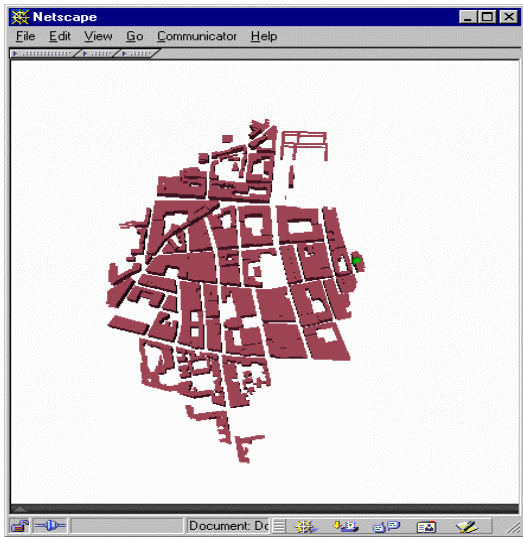


Figure 14: Extraction of polygons in Viewing frustum (implementation at database level using spatial functions of Oracle Spatial, Zlatanova et al 2002)

Therefore a workaround is needed. As we want to create a panoramic image, we map the faces to a panoramic projection and then intersected them by each other. The rear parts are removed, resulting in the coverage of the visible parts of the faces. Now we use the known linkage between the facets and the laser points by a selection at attribute level of the laser points attached to the visible parts of the faces. The distances from the observation point to laser points are calculated and for visualization purposes the points are sent to the rendering system, resulting in our requested distance-value-added panoramic images.

5. CONCLUSIONS AND RECOMMENDATIONS

In this paper we have presented a new approach for creating 3D distance-value-added panoramic images, which make use of

laserscan point cloud (with recorded RGB per point) and 3D geometry derived from the same point cloud. The points, the polygons and the corresponding links are organized in a database managements system.

The presented approach shows promising advantages compared to traditional methods in two general directions. Panoramic images can be automatically created from an arbitrary point (with no need of new measurement in case of new viewing positions). Overload of the system due to large amounts of data can be avoided by extracting only those points and polygons needed for the creation of a particular panoramic image. Algorithms for point extraction within viewing frustum can be developed either at the database level or at client level (e.g. ArcGIS).

The ideas presented in this paper need further implementation and testing. Algorithms for Field-of-View and back-face culling of polygons and corresponding elimination of the laser scan points are the first algorithms to be developed.

ACKNOWLEDGEMENTS

We would like to acknowledge the point segmentation work and the fitting of the faces by our colleague Tahir Rabbani at Delft University, faculty of Aerospace Engineering.

This publication is the result of the research programme of the Delft Centre for Sustainable Urban Development, part of the Delft University of Technology.

The GDMC is the research and development center for Geo-Information technology of Delft University of Technology

REFERENCES

- Frölich, C. (2004). Terrestrial Laser Scanning - New perspectives in 3D-surveying., Proceedings ISPRS working group VIII/2 'Laser-Scanners for Forest and Landscape Assessment' Freiburg, Germany 03-06 October 2004
- Haala, Norbert; Ralf Reulke; Michael Thies and Tobias Aschoff. (2004). Combination of Terrestrial laser scanning with high resolution panoramic images for investigations in forest applications and tree species recognition. "Panoramic Photogrammetry Workshop", Dresden, IAPRS, volume XXXIV, PART 5/W16
- Jansa, J., N. Studnicka; G.; Forkert; A. Haring; H. Kager. (2004). Terrestrial laserscanning and photogrammetry - acquisition techniques complementing one another. www.isprs.org/istanbul2004/comm5/papers/184.pdf
- Luhmann, Thomas. (2004). A historical overview of panorama photogrammetry. "Panoramic Photogrammetry Workshop", Dresden, IAPRS, volume XXXIV, PART 5/W16
- Meijers, Martijn; Jane van Ree and Walter Vroom. (2005). Laserscannen Aula TU Delft, VI Matrix
- Stoter, J.E. and S. Zlatanova. (2003), Visualisation and editing of 3D objects organised in a DBMS, J. Proceedings of the EuroSDR Com V. Workshop on Visualisation and Rendering, 22-24 January 2003, Enschede, The Netherlands, 16p.

Usery, E. Lynn. (2003). "Autostereoscopy – Three-dimensional Visualization Solution or Myth?", Geospatial Visualization and Knowledge Discovery Workshop, November 18-20, 2003, Lansdowne, Virginia

Vosselman, G.; Gorte, B.; Sithole, G.; and Rabbani, T. (2004). Recognising structure in laser scanner point clouds. In International Conference on Laser-Scanners for Forest and Landscape Assessment and Instruments, Processing Methods and Applications, Freiburg. Germany.

Verbree, Edward and Arno van Anrooij. (2004). "Interactive navigation through distance added valued panoramic images", in: H.-G. Maas, D. Schneider (editors): "Panoramic Photogrammetry Workshop", Dresden, IAPRS, volume XXXIV, PART 5/W16, ISSN 1682-1750

Zlatanova, S.; A.A. Rahman and M.Pilouk. (2002), 3D GIS: current status and perspectives, in Proceedings of the Joint Conference on Geo-spatial theory, Processing and Applications, 8-12 July, Ottawa, Canada, 6p. CDROM