MIRROR IDENTIFICATION AND CORRECTION OF 3D POINT CLOUDS

P.-F. Käshammer and A. Nüchter

Informatics VII – Robotics and Telematics
Julius-Maximilians University Würzburg, Germany
`andreas@nuechti.de`

Commission V, WG V/4

KEY WORDS: mirror identification, jump edge detection, panorama creation, plane detection, 3D point cloud correction

ABSTRACT:

In terrestrial laser scanning (TLS), the surface geometry of objects is scanned by laser beams and recorded digitally. This produces a discrete set of scan points, commonly referred to as a point cloud. The coordinates of the scan points are determined by measuring the angles and the time-of-flight relative to the origin (scanner position). However, if it comes to mirror surfaces laser beams are fully reflected, due to the high reflectivity. Mirrors do not appear in the point cloud at all. Instead, for every reflected beam, a incorrect scan point is created behind the actual mirror plane. Consequently, problems arise in multiple derived application fields such as 3D virtual reconstruction of complex architectures. The paper presents a new approach to automatically detect framed rectangular mirrors with known dimensions and to correct the 3D point cloud, using the calculated mirror plane.

1 INTRODUCTION

Scanning reflective surfaces with a 3D laser scanner yields incorrect 3D point clouds, since the emitted laser light is reflected by the surface and a wrong distance is determined. Fig. 1 presents an example, which has won the LiDAR-as-art-contest in 2013. The problem has impact on many post processing steps, e.g., automatic scene modeling and object recognition. In the area of mobile robotics, laser scanners are commonly used for system navigation and robotic mapping or simultaneous localization and mapping (SLAM), i.e., mobile robots have to be able to locate themselves and to identify obstacles. Faulty sensor values of the environment lead to unwanted behavior of the robot and to inconsistent maps.

The scientific contribution of this paper is the implementation of a new approach of solving this problem for the special case of framed rectangular mirrors whose dimensions are known. Thereby, an identification criterion is created, which allows to break down the problem of mirror detection into several subproblems, which are solved by methods of image processing and mathematics. Up to our knowledge, for the first time, a tool has been created to identify mirrors in 3D laser scans and to correct the corresponding 3D point clouds. Throughout the paper, we will demonstrate the algorithms using 3D data acquired with a pulsed Riegl VZ-400 3D laser scanner in multiple mirror situations. The evaluation shows, that the resulting algorithms are applicable without modification to 3D scans acquired using the phase-shift principle. First, we describe the detection of potential mirror contours in a 2D panoramic representation of the point cloud based on a jump edge detection and 3D contour extraction algorithm. Second, we show how actual mirror contours are identified, how the mirror plain is calculated using the Principal Component Analysis (PCA) and also how the point cloud is corrected.

2 RELATED WORK

Not much research has been done dealing with the problem of faulty laser scanner measurements on mirrors and windows. There is only one approach by Shao-Wen Yang and Chieh-Chih Wang Yang and Wang (2008, 2011). In their work, problems of both mirrors and windows are addressed and integrated into the robot's localization, mapping, and navigation framework. They introduced a sensor fusion technique to detect potential obstacles us-
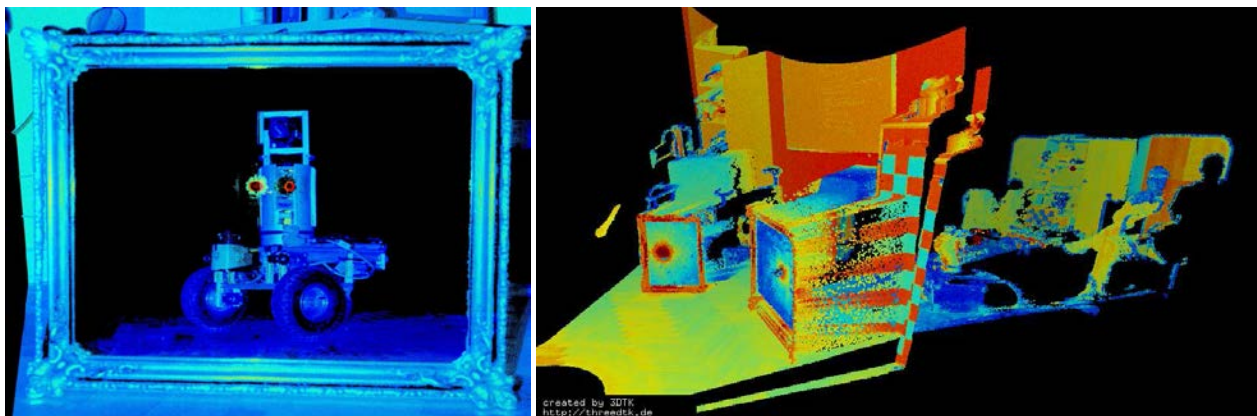

Figure 1: Left: The image depicts how our robot Irma3D sees itself in a mirror. The laser looking into itself creates distortions as well as changes in intensity that give the robot a single eye, complete with iris and pupil. Thus, the image is called "Self Portrait with Duckling". Image courtesy Jan Elseberg. Right: Scanned scene with two mirrors.

ing sonar sensors and a laser scanner. However, their approach works in 2D only. Mirrors are finally described as obstacles in a 2D occupancy grid map. In contrast to that, this paper describes more a tool to find the exact 3D mirror position and to correct 3D point clouds after the recording of the laser scan.

Only a small number of researchers have looked into the issue of detecting specular surfaces and correcting the errors automatically. A related problem is the detection of transparent or translucent objects. The related works in the area range from 3D point cloud processing of kinect data to environment perception and object reconstruction methods Foster et al. (2011); Ihrke et al. (2010); Klank et al. (2011); Albrecht and Marsland (2013).

3 PANORAMA CREATION

To find potential mirror contours we detect jump edges. First, a 2D panoramic range image of the point cloud is created using the *equirectilinear* projection method Hamidreza Houshiar and Nüchter (2013). When creating panorama images from laser scans, each point of the 3D point cloud is projected onto a 2D array of a certain resolution. Therefore the scanned environment of the laser scanner is seen as the projection of a sphere around the scanner. Given the 3D point in spherical coordinates (θ, ϕ, r) the projection method determines on which 2D pixel with the coordinates (u, v) a 3D point is projected. If multiple 3D points fall into a pixel, e.g., by full-wave-analysis or resolution mismatches, they are stored in a vector. The distance r of one 3D point defines the value of a pixel in the range image version of the *equirectilinear* projection. While many projection methods are available (Hamidreza Houshiar and Nüchter (2013)) for this work, the projection method *equirectangular* is used which is defined by the simple projection rule:

$$u = \theta$$
$$v = \phi$$

with longitude θ and latitude ϕ.

Depending on the resolution of the panorama image, multiple points of the point cloud are usually projected on the same pixel (u, v). If this occurs, there are two possibilities: The map method *FARTHEST* takes the point with biggest range r as represented pixel; the map method *NEAREST* the one with smallest range r. Depending on the point cloud and the mirror situation both methods have different effects, so it has to be chosen individually for each scan, cf. Fig 2.

4 JUMP EDGE DETECTION AND CONTOUR EXTRACTION

Jump edges are detected in the panoramic range image and a corresponding jump edge image of same resolution is created. Between two adjacent pixels of an image, there is a jump edge if the difference in their value is greater than a certain threshold. The sign of the difference describes which of the two pixels is closer to origin, i.e., scanner pose. A jump edge is only represented in the jump edge image, if the considered pixel is closer to origin than the one compared to. This is important so that the detected contour describes only the mirror frame and contains no pixels within the actual mirror plane. Otherwise 3D contours cannot be extracted properly.

Extracting the detected 2D contours of the jump edge image is a known problem of image processing and is done by using the corresponding OpenCV-function `findContours` in `CvRetrExternal`
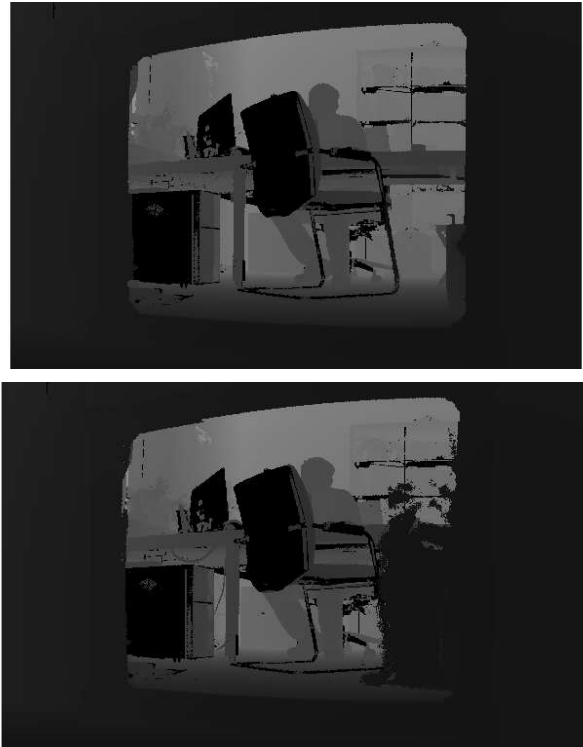

Figure 2: Part of a range image of a mirror. Undefined pixel appear black. Without fillZero-function. Top: Method farthest. Bottom: Method nearest. Not ideal case with 3D points on the mirrors plane.

mode. Subsequently the corresponding 3D contours are extracted by testing which 3D point is projected on the considered 2D contour pixel, which is the reverse process of creating the range image. Note: The chosen map method changes the outcome of the extracted 3D contour. To get the frame point as 3D contour point, the map method has to be *nearest*.

In the ideal case, all 3D mirror points are laser beam reflection and are therefore further away from the scanner and need to be corrected. However, in some cases 3D points occur that describe the actual mirror surface due to not perfect reflections or detection using the full-wave-analysis, cf. Fig. 2 bottom picture. The 2D contour will then be shifted to the inside of the mirror and will not describe the mirror frame anymore. Consequently, 3D contours might not seem to be connected what leads to the need of filtering these contours or erroneous calculated plane. In those cases the map method *farthest* is chosen to avoid this problem (Fig. 2 top picture).

5 IDENTIFICATION OF MIRROR CONTOURS

So far we describes how potential 2D and corresponding 3D contours are extracted. Now the actual mirror contours have to be identified among all detected ones.

In a first step a connected filter is applied on all detected contours. It calculates the distances of all 3D contour points towards their next neighbor. If the distance between two neighbored contour points is larger than a certain threshold, the considered contour is seen as unconnected and therefore filtered. If a considered 3D mirror contour contains points that do not describe the mirrors frame due to previous problems in detecting the jump edges, this filter decreases in efficiency because the used threshold has to be

Figure 3: Top: Range image of scanned bathroom environment with five mirrors of same size and resolution of 3600×1000. Middle: Corresponding jump edge image with a threshold of 0.1 m. Bottom: Extracted 2D contours of picture above.
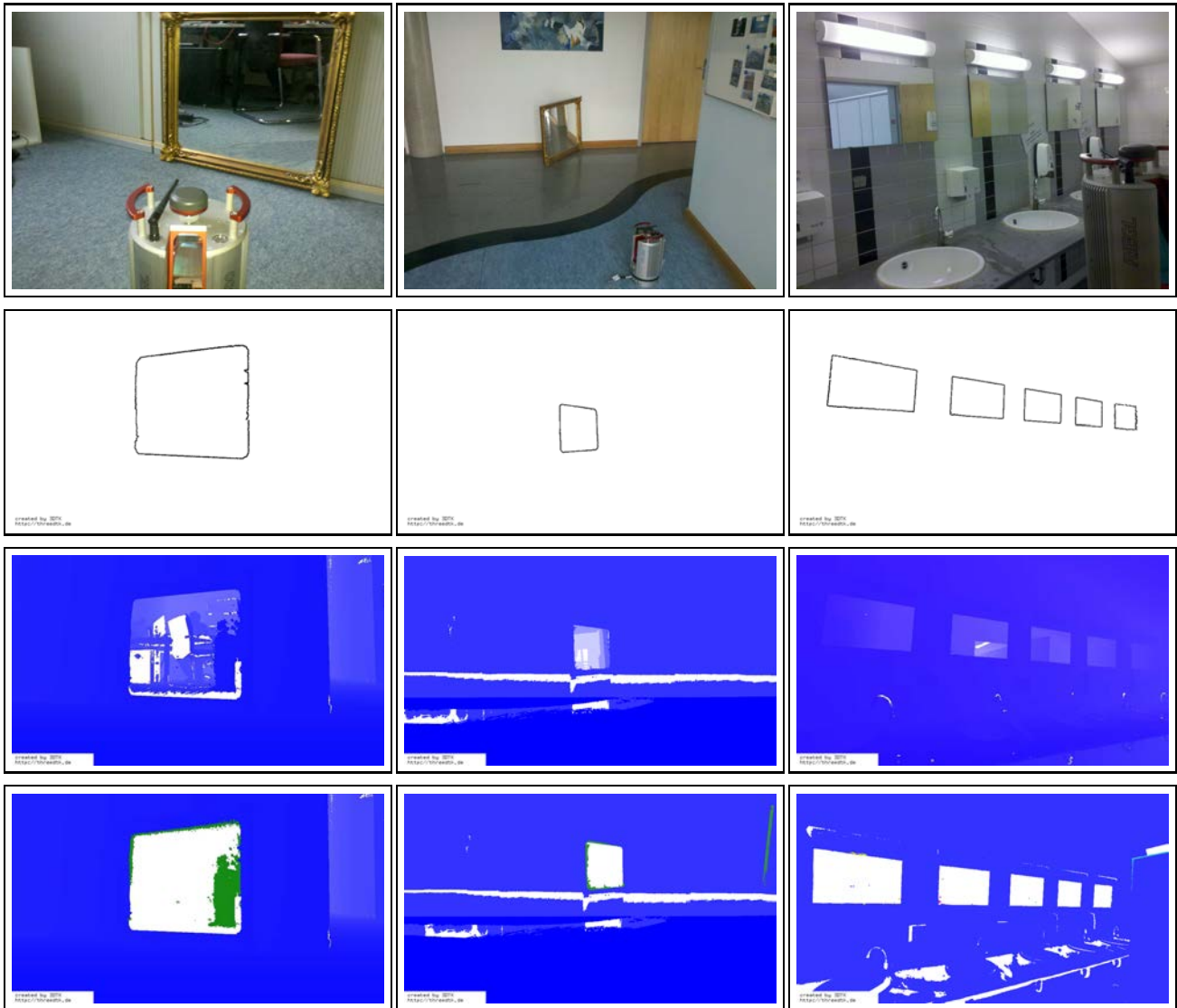
Figure 4: Left: Mirror of size 56×64 cm in an office environment. Middle: Same mirror placed in an office corridor. Right: 5 mirrors of size 40×60 cm in a bath room. Top row: Photo of the scenes. Second row: Extracted 3D contours. Third row: Screenshots in a 3D point cloud viewer without mirror correction. Bottom: Corrected 3D point clouds.
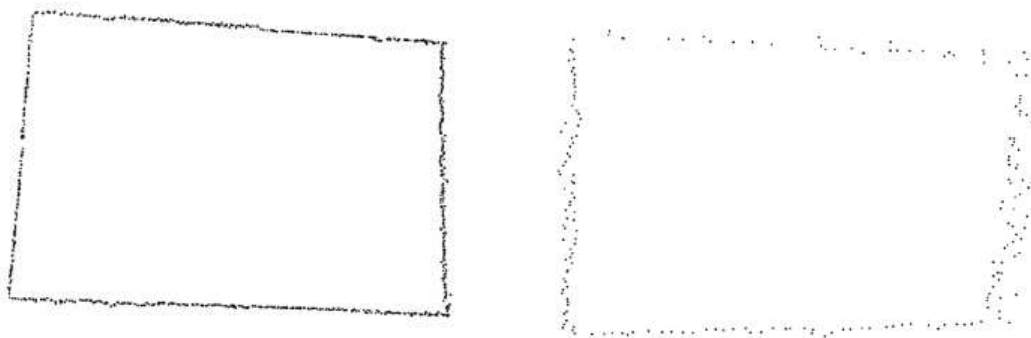

Figure 5: Two extracted mirror contour as 3D point clouds of mirrors. The right side features always more noise, since the incidence angle is much smaller theren.
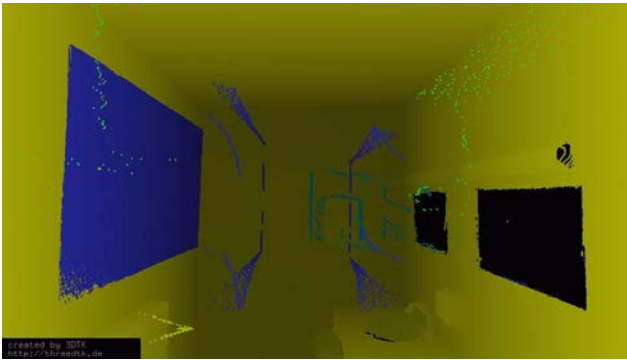
Figure 6: Screenshot of corrected point cloud acquired by Z+F 5006 Scanner in bathroom environment.

increased to not filter the actual mirror contours. Consequently more non-mirror contours pass this filter. In a second step, for all remaining 3D contours, the dimensions as well as the best fit plane through the contour points, which is the mirror plain for a mirror contour, are calculated using the Principal Component Analysis (PCA) Pearson (1901).

In a second filtering process, contours having the specified mirror dimensions within a certain margin of error are identified. For an identified mirror contour, the original point cloud is corrected at hand the corresponding 2D contour as well as the calculated mirror plane. To this end, the OpenCV function `pointPolygontest` is used to find all 3D points which are projected within the 2D mirror contour. Those are then reprojected on the mirrors plane, stored in a separate 3D point cloud and deleted from the original one.

6 DEALING WITH UNDEFINED PIXEL

There are many challenges in dealing with real-world data. E.g., dealing with undefined pixel in the created panoramic range image which originate mainly due to highly reflecting surfaces in the scanned environment, e.g., when scanning water. This means there are pixel in the panoramic image on which no 3D point is projected and therefore do not posses a range value.

Since these undefined pixel cause false jump edges and so creating incorrect contours, a method for dealing with those has been developed. Therefore a `fillZeros` function assigns values to undefined pixels by copying the closest properly measured pixel from top-left in the 2D image coordinates. This algorithm often leads to shifting of the 2D mirror contour either towards the inside of the mirror or towards the outside. The second case is not a problem because normally the contour will still describe the frame or at least the mirrors plane. In case of a shifting towards the inside, a second algorithm called `findFramePoint` fixes this by searching for 3D points which lie in the opposite direction of the mirror contours geometrical mean.

7 EXPERIMENTS AND RESULTS

The mirror identification software is implemented in C/C++ based on the Open Source software *3DTK — The 3D Toolkit*. Andreas Nüchter et al. (2015). The software has been tested with a Riegl VZ-400 terrestrial laser scanner and the Zoller+Fröhlich Imager 5006 in the same environment.

For both scanners all mirrors have been identified and thus the corresponding point clouds were corrected. The only case where
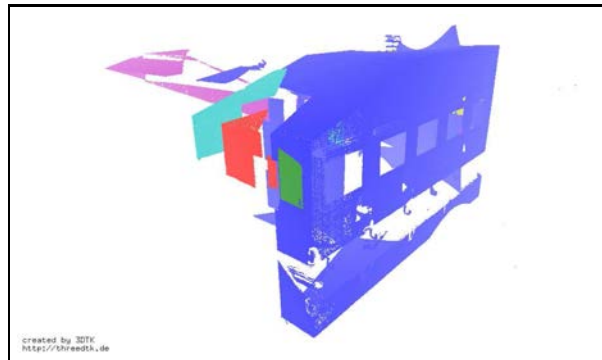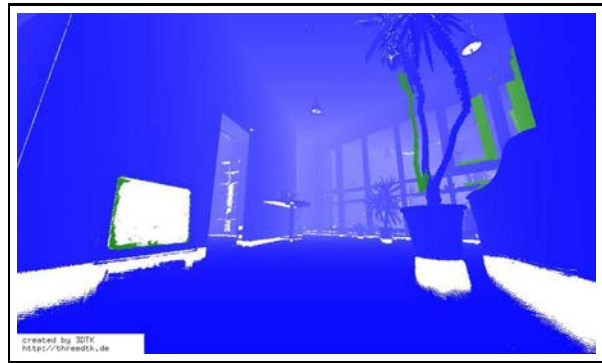

Figure 7: Screenshots of corrected point clouds in a point cloud viewer.

mirrors cannot be identified at all is if mirrors in the panoramic representation are cut of by the panorama frame. This is because the mirror contour cannot be detected as connected in the current software version. In future work, we will mind the scanners starting position with respect to the mirror. For all tested point clouds, 3D mirror contour errors are up to 9 cm in x and y dimension and up to 3.5 cm in z dimension, which is due to the discretization induced by the (relatively low resolution of the) panorama image. There a two main effects that increase the 3D mirror contours and consequently also effect the error of the mirror plane and corrected points. With decreasing incidence angle between laser beams and mirror plane, noise in the extracted frame increases, cf. Figur 5. If mirror plane and laser beams were parallel, the mirror would not appear at all in the point cloud.

Figure 8 shows the result of the automatic correction on data acquired with a Riegl VZ-400, a plused laser scanner. The top row and bottom row show that the 3D points are projected to the correct location. Figure 7 (bottom) gives a 3D view of the bottom row of Figure 8. The remaining misalignment can easily be resolved scan matching methods, such as ICP (iterative closest points). The middle row of Figure 8 demonstrates, how our software can be used to look around the corner using a mirror.

All presented algorithms are also applicable to scans of a Zoller+ Fröhlich Imager 5006, which measures according to the phaseshift principle. Nevertheless there is one difference in the outcome. The corrected point clouds of the Zoller+Fröhlich Imager contain several points "flying around" which look like 3D mirror contours but shifted relative to the scanner, cf. Figure 6. Those points arise from the different principle of operation. Due to the finite spot size, if the laser beams hits the mirrors frame, the measurments are averaged between the frame point and the mirrored point. In contrast to the Riegl, this scanner is not able to distinguish several targets by Full Wave Analysis. Since those faulty points are reprojected by the correction algorithm. Figure 6 whos a 3D point cloud with corrected mirrors and erronous contours.
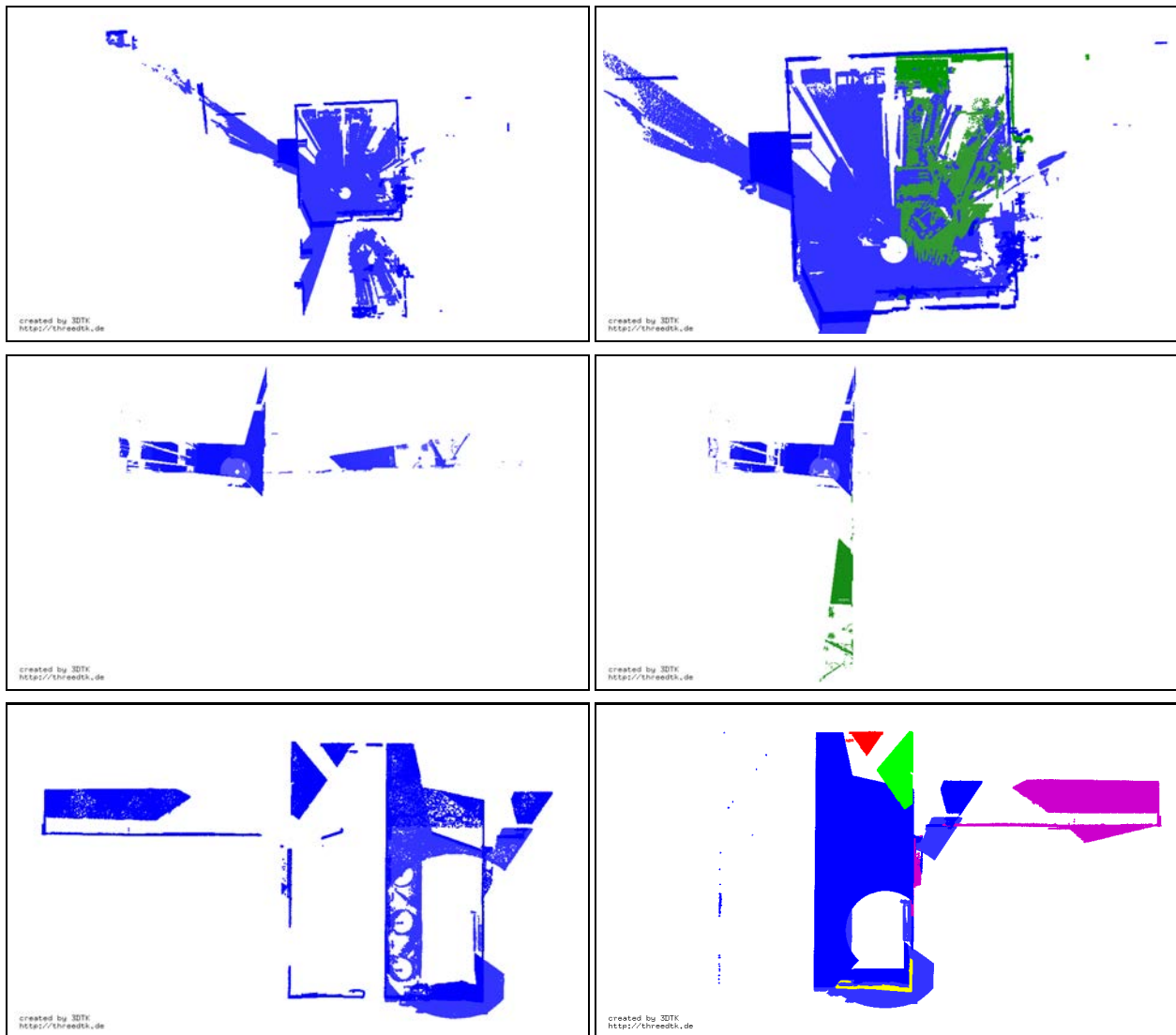
Figure 8: Screenshots of 3D point clouds in bird eye view. Left: Uncorrected point clouds. Right: Corrected point clouds. Top: Office environment. (Note that right picture is zoomed) Middle: Office corridor. Bottom: Bathroom with 5 mirrors.

8 CONCLUSIONS AND OUTLOOK

This paper describes a complete solution to correct point clouds that contain a mirror of know size. It has been demonstrated to work well in various environments and with pulsed and phse-shift scanners.

Needless to say, a lot of work remains to be done. In future work, will concentrate on arbitrary mirrors, i.e., without knowing the mirror size in advance. Furthermore, we will integrate this technology into our registration and mapping methods and aim to develop a method to distingish mirrors from windows. Overall, we aim at reducing the time needed for manually correcting and post-processing of 3D point clouds.

References

Albrecht, S. and Marsland, S., 2013. Seeing the unseen: Simple reconstruction of transparent objects from point cloud data. In: Robots in Clutter (RIC) Workshop at Robotics: Science and Systems (RSS '13), Berlin, Germany.

Andreas Nüchter et al., 2015. 3DTK — The 3D Toolkit. http://slam6d.sourceforge.net/.

Foster, P., Sun, Z., Park, J. J. and Kuipers, B., 2011. Visagge: Visible angle grid for glass environments. In: Proceedings of the IEEE International Conference Robotics and Automation (ICRA '11), Karlsruhe, Germany, pp. 2213–2220.

Hamidreza Houshiar, Dorit Borrmann, J. E. and Nüchter, A., 2013. Panorama based point cloud reduction. In: Proceedings of the 16th IEEE International Conference on Advanced Robotics (ICAR '13), Montevideo, Urugauy.

Ihrke, I., Kutulakos, K., Lensch, H., Magnor, M. and Heidrich, W., 2010. Transparent and specular object reconstruction. Computer Graphics Forum 29(8), pp. 2400–2426.

Klank, U., Carton, D. and Beetz, M., 2011. Transparent object detection and reconstruction on a mobile platform. In: Proceedings of the IEEE International Conference Robotics and Automation (ICRA '11), Shanghai, China, pp. 5971–5978.

Pearson, K., 1901. On lines and planes of closest fit to a system of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 6(2), pp. 559–572.

Yang, S.-W. and Wang, C.-C., 2008. Dealing with laser scanner failure: Mirrors and windows. In: IEEE International Conference on Robotics and Automation (ICRA), Pasadena, California.

Yang, S.-W. and Wang, C.-C., 2011. On solving mirror reflection in lidar sensing. IEEE/ASME Transactions on Mechatronics 16(2), pp. 255–265.