# BRUTE FORCE MATCHING BETWEEN CAMERA SHOTS AND SYNTHETIC IMAGES FROM POINT CLOUDS 

R. Boerner ${ }^{\text {a,* }}$, M. Kröhnert ${ }^{\text {a }}$<br>${ }^{\text {a }}$ Institute of Photogrammetry and Remote Sensing, Technische Universität Dresden, Germany richardbo@web.de, melanie.kroehnert@tu-dresden.de

## Commission V, WG V/5

KEY WORDS: 3D-Monoplotting, Line Matching, Line Extraction


#### Abstract

:

3D point clouds, acquired by state-of-the-art terrestrial laser scanning techniques (TLS), provide spatial information about accuracies up to several millimetres. Unfortunately, common TLS data has no spectral information about the covered scene. However, the matching of TLS data with images is important for monoplotting purposes and point cloud colouration. Well-established methods solve this issue by matching of close range images and point cloud data by fitting optical camera systems on top of laser scanners or rather using ground control points. The approach addressed in this paper aims for the matching of 2D image and 3D point cloud data from a freely moving camera within an environment covered by a large 3D point cloud, e.g. a 3D city model. The key advantage of the free movement affects augmented reality applications or real time measurements. Therefore, a so-called real image, captured by a smartphone camera, has to be matched with a so-called synthetic image which consists of reverse projected 3D point cloud data to a synthetic projection centre whose exterior orientation parameters match the parameters of the image, assuming an ideal distortion free camera.


## 1. INTRODUCTION

Conventional approaches for the registration of images and point clouds use the combination of optical systems and laser scanners or tie points. Obviously, camera-scanner-systems as well as ade-quate control data are highly cost-expensive. With the aid of modern smartphones, it is possible to measure the exterior orientation from smartphone-taken images using inbuilt Micro-Electronic-Measurement-Systems (MEMS) for orientation and Global Navigation Satellite Systems (GNSS) for position information. Furthermore, these information could be used for a reverse projection (hereafter referred to as re-projection) of 3D point cloud data which provides synthetic images of ordered 2D pixels with respectively additional exterior informaZtion about the mapped objects. In this paper, captured images are referred to as real images, because they are taken by (smartphone) cameras and thus not derived from point clouds.

In case of user's navigation through urban regions which were previously covered by 3D point cloud data, the approach aims for the possibility of 3D information for each point that refers to a e.g. imaged cityscape by real-to-synthetic image registra-tion. Thereby, the initial measured position and orientation could be used to re-project 3D points regarding the camera projection centre to a virtual image plane, which generates the mentioned synthetic image. A successful coregistration enhances the con-catenation of real and synthetic pixels (figure 1).

After the co-registration of smartphone and synthetic images the smartphone-taken image is used for an analysis of the object geometry (as done with real images in Becker et al., 2004). Furthermore, the colour information from the image pixels can be transformed into the 3D space for colourisation of TLS data (Meierhold et al., 2010).
Georeferenced point clouds with higher-ordered accuracies up to
*Corresponding author
a few centimetres could be used to improve the smartphone measurements regarding the exterior orientation.


Figure 1: Schematic visualisation regarding the matching process of co-registered real and synthetic image data.

Extracted elements from smartphone images, like lines, could be transformed in the object space with the aid of the co-registered synthetic image, providing 3D information that could be used further to extract the height information of lines in one captured image (e.g for window height determination).

However, the registration of real and synthetic images is a considerable challenge. There are strong distinctions between the datasets, especially in case of using intensity point clouds. A registration approach using SIFT is presented in Meierhold et al. (2010). Unfortunately, further tests have shown that to deal with real and synthetic images means to deal with strong differences in terms of colour distribution. Therefore, default matchers (like SIFT) seem to be inoperative.
To deal with the issue, the registration of smartphone and synthetic images uses the 3D geometry information of the point cloud
data, which is extracted from the synthetic image.
Geometric primitives like points or lines enable the correlation of both image types. The first step comprises the detection of 2D and 3D lines in the smartphone-taken and the synthetic image respectively which will be used further to obtain correspondences between 2D and 3D lines. Hence, the synthetic image represents a hybrid between the 2D and 3D space, it is used for a robust and fast 2D detection to get corresponding 3D lines (chapter 3).
Secondly, the extracted lines are still used for a following correlation process (chapter 4) whose resulting matches represent an approximation for a further resection (Meierhold et al., 2008). Chapter 5 presents the results from line extracting and matching. The next chapter shows the used datasets and outlines the projection method for synthetic images.

## 2. STUDY REGION \& DATA AQUISITION

For investigation purposes, two point clouds of different buildings belonging to Technische Universität Dresden are used: the principalship (projected coloured point cloud data, figure 2) and the Hülssebau, the location of the Institute of Photogrammetry and Remote Sensing. Up to this, figure 3 shows the projected intensity values of the point cloud data due to their distance concerning the re-projection centre.


Figure 2: Synthetic image of principalship. Input data: coloured point cloud.


Figure 3: Distance image of the projected point cloud from the TU-Dresden's Hülssebau. Distances belong from 3D points to the projection centre.

The re-projection bases on the common central perspective (see equation 2 and 3 in chapter 4). Using such a central perspective without further investigations leads to transparent foreground objects (like walls or trees). Point cloud data are no continuous representation of a scene, there are spaces between the measured points. These spaces are adopted by the re-projection and will
lead to pixels without information inside the foreground objects. These information gaps will be filled with information of background pixels. Therefore, the mentioned re-projection will show strong image noise. To detect falsely background information inside foreground objects, resulting from this noise, a pyramid approach is used. It can be simplified described as inflating the re-projected points to create continuous image areas which are used for noise reduction and to provide homogeneous image areas. The tree in the patio of building Hülssebau (figure 3) shows the result of the approach. There are blank pixels inside the tree which would be filled with pixels from the wall behind the tree. But thanks to the filter method, the pixels inside the tree are left blank.

For the image acquisition of real images, the Android smartphone Samsung Galaxy S5 is used, providing a 16 MP camera and a pixel size of $1.12 \mu \mathrm{~m}$ (GSMArena, 2016). However, a resolution of about 3 MP is used which is a compromise in the level of detail and lower resolution respecting the synthetic images.

## 3. LINE EXTRACTION FROM SMARTPHONE AND SYNTHETIC IMAGES

There are three essential steps for the extraction of lines from camera and synthetic images. Firstly, the edge points in both image types are detected, which is described for the smartphone and synthetic images in chapter 3.1 and 3.2. The second step comprises the segmentation of the edge points to get linearly areas of potential edges (chapter 3.3). Finally, a linear regression by the method of least squares (Niemeier, 2008) is used for the calculation of the line parameters from each segment.

### 3.1 Smartphone Images

There are several approaches for solving the term of image line extraction. In this case, an adapted compass edge detector using Prewitt kernels (Prewitt, 1970; implemented in Catalano, 2016) is applied. Disturbing image signals, like edges of roofing tiles, are reduced by the application of a bilateral filter technique (Tomasi \& Manduchi, 1998; implemented in Bradski, 2000) prior to the edge detection.
The edge detection uses common structure elements to get directed gradients. For the gradient, referring to the north, the following kernel is used:

$$
\text { kernel }_{n o r t h}=\left[\begin{array}{lll}
-1 & 0 & 1 \\
-1 & 0 & 1 \\
-1 & 0 & 1
\end{array}\right]
$$

For all other directions, the kernel $_{n o r t h}$ has to be rotated in the specified way. Hence, by a rotation about 45 degrees to the east, the northeast kernel is defined by:

$$
\text { kernel }_{\text {northeast }}=\left[\begin{array}{ccc}
-1 & -1 & 0 \\
-1 & 0 & 1 \\
0 & 1 & 1
\end{array}\right]
$$

In sum there are eight gradients, pointing clockwise from the north to the south and back to the north.
Against the approach of Catalano (2016), pairs of gradients are used to get the total count of four directions with regards to their adjustment. Thus, the vertical gradient is produced by the kernels which point to the north and the south. Further gradients are the northeast to the southwest, east to west and the southeast to the northwest. Joining kernels is implemented by focusing on the maximal value of both kernels to the resulting pixel. To get the final edge image, all directions are combined in the same way.

The gradient, which provides the highest value on the current pixel, is used to map his gradient direction to the current pixel. Therefore, each edge point gets additional data for his edge direction.
A key advantage of the merging and extraction of edge direction is that the adopted direction from each edge pixel can be used for the segmentation. Furthermore, a threshold, applied by the merge of kernels or rather gradients is used to reduce image noise.

### 3.2 Synthetic images

There are several ways to detect lines in the 3D space. As supposed in Meierhold et al. (2008) distance images offer the possibility to detect edge points by checking their differences in distance respectively to the projection center. Nevertheless, this may lead to false edge points in case of planes which point perspectivally in depth direction. Becker et al. (2004) present a line detection approach that uses the intersection of planes within the point cloud. For this, 3D points belonging to one plane are investigated. By a least square approach, the resulting plane is calculated. Unfortunately, this robust method is very computational intensive and the detection of points belonging to one plane is a considerable challenge.

In the following, an additional extraction technique for 3D lines due to synthetic images is explained which uses 3D information of each re-projected point and the same four gradients as described in chapter 3.1. Instead of grey values of the re-projected neighbouring pixels (depending on the specified gradient directions), their 3D information is used to indicate pixels as edge points.
For their determination, all pixels along the respective direction are observed regarding their directly adjacent pixels.
Due to the current considered pixel and its both neighboured pixels along one gradient, there are three 3D points that describe two 3D vectors in the object space. Edge points are detected by checking the angle between these two vectors.
Both vectors will intersect each other at an angle of about 0 or 180 degrees, in case of belonging to one plane. Otherwise, the angle will differ up to 30 or rather 150 degrees.


Figure 4: Schematic visualisation of the edge detection approach

Figure 4 exemplary outlines the mentioned angle between two vectors of one gradient. The figure shows the horizontal gradient from west to east in the image as the projection of $I_{\text {synt }}$ and the corresponding vertical edge in the object space. Each pixel $\left(P_{i}\right)$ in $I_{\text {synt }}$ has additional 3D information $\left(C_{i}\right)$ which will be used to construct vectors in space. These red marked vectors point from the current pixel to the western and eastern adjacent pixel at an intersection angle $\leq 180$ degrees and thus they refer to an edge
for the horizontal image gradient.
Other image gradients represent space vectors which point to edges by intersection angles $\leq 180$ degrees too. The pixels regarding the north-south gradient describe two vectors with an intersection angle of about 180 degrees. Therefore, the vertical image gradient does not indicate an edge in the mentioned space. Obviously, the vertical edges in space can be mapped to their respective vertical image gradients on condition of "all gradients point to edges, except one". That gradient, referring to the excepted one and thus fulfil this condition, is taken as the direction of the currently considered edge pixel.

These edge pixels directions are used to improve the output image by morphology or so-called directional histograms. For the first one, a morphologic closing with a directed line structural element is applied which combines points next to each other into one line.
The histogram filters noise by indicating the main direction inside an image area. To solve this issue, the image is clustered in sub-images, where the related histogram is calculated by counting pixels of the same direction. This histogram shows the main direction by the bin with the most entries. Noise is indicated by bins with counts less than a threshold regarding the main direction. The result from this histogram filter is shown in figure 5, with empirical estimated values for the subimage dimension of 30 by 30 pixels and a threshold of $60 \%$ to indicate noise.


Figure 5: Result before (top) and after histogram filtering (below) for the point cloud from the principalship. The different directions of edge points are drawn in different grey values.

Beside the angle threshold, a second parameter is used which describes the distance between two neighbouring pixels along one gradient. A step size, defined by the distance from a re-projected pixel to its origin, allows the adjustment of fixed distances in the object space in relation to their neighbours. Considering that, an empirical estimated value of about 7 cm is used. It is necessary to adjust the pixel size for re-projection purposes regarding the pixel-to-re-projection centre distance. The in this way adjusted pixel size is used to get an approximation of the pixel count, representing the 7 cm in the image with respect to the pixel's distance.

The adjustment is done with the aid of the re-projection pixel size, the pixel-to-re-projection centre distance and the intercept theorem. The re-projection use a pixel size given in 1 m distance, because of formula simplification. Considering that, the pixel size of the whole image with a pixel-to-re-projection centre distance of 1 m is given as $p s$, the intercept theorem provides:

$$
\frac{p s \cdot W / 2}{1[\mathrm{~m}]}=\frac{p s_{d} \cdot W / 2}{d_{\perp}}
$$

where
$p s_{d}=$ adjusting pixel size
$d_{\perp}=$ distance from the image plane
at the adjusting pixel to the camera origin
W = image width

Furthermore, the intercept theorem describes the connection between $d_{\perp}$ and the distance between the re-projected point (pixel) and its (synthetic) origin. Thus, the adjusted pixel size concerning the current pixel is calculate by:

$$
\begin{equation*}
p s_{d}=p s \cdot \frac{d}{\sqrt{1[\mathrm{~m}]+\tilde{c}^{2}+\tilde{r}^{2}}} \tag{1}
\end{equation*}
$$

with

$$
\begin{aligned}
& d= \text { distance from pixel to origin }[\mathrm{m}] \\
& \tilde{c}= \text { column of the pixel with the origin of the } \\
& \text { image system in the image centre }[\mathrm{m}] \\
& \tilde{r}=\text { row as above }
\end{aligned}
$$

### 3.3 Segmentation

The segmentation for grouping edge pixels to line segments comprises two steps.

First, all neighbours with the same direction are grouped in segments by starting at one edge pixel going to the next neighbour in edge direction (figure 6). In case of non-existing neighbours with respect to the current direction, or deviating directions of the adjacent neighbours, the segment is complete.


Figure 6: First segmentation step. Left: coloured starting pixel to each segment. The arrows show the direction for corresponding neighbours. Right: resulting segments

The second step involves the combination of neighbouring segments pointing in the same direction. For the indication of neighbouring segments, their bounding boxes are calculated. Afterwards the bounding boxes are checked for intersection. In that case, two pixels of these segments are analysed for contingents. To reduce noise, another histogram filter approach is applied regarding the length of the estimated segments, which are defined by the diagonal of the segment bounding box. Further on, the histogram is calculated in a fixed range, for example from zero to a threshold with respect to the longest segment. Subsequent to the calculation, all segments smaller than the length, defined by the maximum bin of the histogram, will be deleted.
Empirical estimations show that an range of 0 to $15 \%$ of longest segment reduces noise and vegetation (figure 7). Furthermore, a range of $15-50 \%$ keeps only outer building edges. Consequently, the filter approach offers the possibility for handling the level of detail by moving the calculation interval.


Figure 7: Segmentation result in relation to figure 2, e ach segment is drawn in a different colour.

## 4. BRUTE FORCE MATCHING

The matching looks for the best approximation of the image correlation by re-projecting 3 D information from the point cloud to each possible camera system and detects the highest correlation point. The orientation is handled as a fixed value considering the condition that smartphone MEMSs have a higher degree of accuracy compared to smartphone GNSS sensors.
To delimit the correlation operation, the accuracy parameters from smartphone measurements are used to determine a correlation interval. The interval limits the range of possible positions of reprojection centres. Furthermore, the interval is generalised by a fixed number of points on each coordinate axis in the global coordinate system, to get a finite number of correlation points (for example 5 points on each axis will create 125 correlation points and further 125 possible camera systems).

Prevailing smartphone sensors are used to get approximations of the user's and hence his camera's position and orientation in consideration to a georeferenced point cloud. These approximations serve as initial parameters for the correlation. With help of the defined interval, the correlation leads to more appropriate approximations for the resection than the sensor values only.
The transformation of 3D information to the camera system, and so far to one correlation point, is defined by (Szeliski 2010):

$$
\begin{equation*}
\vec{X}_{c}=\mathbf{R}\left(\vec{X}_{g}-\vec{X}_{0}\right) \tag{2}
\end{equation*}
$$

$$
\text { where } \quad \begin{aligned}
\vec{X}_{c}= & \text { point in the camera system } \\
\mathbf{R}= & \text { rotation matrix from the } \\
& \text { global coordinate system (GLCS) } \\
& \text { to the camera coordinate system (CMCS) } \\
\vec{X}_{g}= & 3 \mathrm{D} \text { point in the GLCS } \\
\vec{X}_{0}= & \text { camera position in the GLCS }
\end{aligned}
$$

$\vec{X}_{g}$ is calculated from the synthetic image by an interpolation of line points near to the re-projected segment point. The initial values for $\mathbf{R}$ and $\vec{X}_{0}$ are delivered by the smartphone sensors. Afterwards, the coordinates are transformed from camera to the image system by:

$$
\begin{equation*}
\binom{\text { column }}{\text { row }}=\binom{x_{h}-c \cdot \frac{x_{c}}{z_{c}}}{y_{h}-c \cdot \frac{y_{c}}{z_{c}}} \tag{3}
\end{equation*}
$$

The interior orientation $\left(x_{h}, y_{h}, c\right)$ is estimated from the image dimension and the half of the horizontal view angle (H):

$$
\begin{aligned}
x_{h} & =\text { columns } \cdot 0.5 \\
y_{h} & =\text { rows } \cdot 0.5 \\
c & =\frac{\text { columns } \cdot 0.5}{\tan H}
\end{aligned}
$$

For the line matching purpose in relation to the synthetic and the smartphone image the following linear equation is used:

$$
\begin{equation*}
\vec{g}(t)=\vec{a}+t \cdot \vec{b} \tag{4}
\end{equation*}
$$

with $\quad \vec{a}=$ starting point of the line
$\vec{b}=$ line direction
$\vec{a}$ and $\vec{b}$ are calculated using two re-projected points from the corresponding 3D line. Therefore, the 2D information from the primary re-projected point is used for the determination of $\vec{a}$. The difference of the first and second re-projected point determines the vector $\vec{b}$.

For the lines in the smartphone image the homogeneous linear equation is used with:

$$
\begin{equation*}
0=a \cdot x+b \cdot y+c \tag{5}
\end{equation*}
$$

The parameters $a, b$ and $c$ are acquired segmental by the least square regression. This equation can be transformed in the linear equation 4 using two points of the same line and calculating $\vec{a}$ and $\vec{b}$ as described above.

Corresponding lines have to fulfil two conditions: First, the line directions have to be equal, which is checked by their scalar product. For lines whose normalized vectors $\vec{b}$ intersect each other at an angle near to 0 or 180 degrees, the scalar product has to be close to $\pm 1$.
For the second condition, the starting point of an observed line refers to an element of its surrounding lines (solve equation 5 for the starting point).

With the coordinates from one 2D point ( $\mathrm{x}, \mathrm{y}$ ) and the considered line parameters ( $a, b$ and $c$ ) the right side of equation 5 calculates the distance from the point to the line which helps to assess the quality of the line match.

This assessment is done by the distance of two lines, which is calculate for parallel lines (whose vectors $\vec{b}$ are equal), whereby the lowest distance indicates the best match.

All lines are matched pairwise in each correlation step. For the correlation coefficient, the mean value of all matching distances is calculated. The minimum mean value marks the maximal correlation coefficient and thus the best fitting correlation point, which indicates the final matched camera position. In conclusion, the correlation gives the best approximation for projection parameters and line matches.

## 5. RESULTS

Chapters 5.1 and 5.2 show the results from line extraction and matching through each examples of TU Dresden's Hülssebau and principalship, as well as the prospected future work.

### 5.1 Line Extraction

Figure 8 shows the smartphone-taken image of TU Dresden's principalship. Additionally, figure 9 presents the edge image after application of the modified Prewitt operator. The threshold for kernel merging is set to a grey value of 100 , which shows the best compromise regarding noise reduction and edge retentions.


Figure 8: Photo of the principalship


Figure 9: Edge image after application of the modified Prewitt operator (chapter 3.1)

Obviously, the application of Prewitt's operator detects a considerable amount of the building's main edges and is less influenced by so-called edge noise due to vegetation.

The directions for the edge points, detected by modifications (chapter 3.1) coincide with the direction of detected lines in the image (e.g. horizontal lines in the image lead to edge points which adopt the horizontal direction). Unfortunately, Prewitt's edge detection fails in case of edges with low contrasts which is a common problem in several edge detection approaches. These edges are detected by their geometric information due to their position within the point cloud (figure 5).

In practice, the best results are produced at an angle threshold of 30 degrees, whereby space vectors that intersect each other at an angle of about 30 to 150 degrees mark an edge. Furthermore, the use of a minimal distance in relation to two neighbouring pixels referring to one gradient of about two pixels reduces noise.

The result for the building Hülssebau is shown in figure 9. Figure 10 visualises a minor disadvantage of the segmentation technique. The diagonal edge along the roof cannot be combined to one segment. Because of its cluster structure, the roof edge is no longer considered as one close building edge and will be eliminated by the histogram filter approach which, as already known, depends on the segment length. To reduce information loss due to this type of clustering, the segmentation procedure that combines
two neighbouring segments should be enhanced. The clustering needs control by a distance parameter for the neighbourhood of two segments.


Figure 9: Edges detected from the re-projected point cloud data of building Hülssebau.


Figure 10: Detail image of the right wall of building Hülssebau. Green lines point to the results after application of the regression task.

### 5.2 Matching

Figure 11 shows the matching result for the principalship. The graphic visualises the combined starting points of the matched lines. The empirical estimated projection parameters point out that a suitable matching result could be achieved by a distance less than 20 pixels. The pros and cons regarding the matching become visible along the matched roof edges. One advantage is plain to see: the linear equation has no limitations in terms of the line lengths. Thus, the algorithm offers the possibility for the registration of several segments to one connected line, which supports the handling of concealments. Shortcomings are noticeable due to double matchings (like the roof edge from the principalship inside the point cloud image which is near to the
roof edge in the acquired image and thus matched to this). Further investigations will handle this issue by an additional check of the closest matches. For concealments, the behaviour of the line parameters needs more investigations. If these parameters are within a previous defined threshold interval, a double match is probably a match of two parts of the same line.

To verify the influence of projection parameters to the correlation, two synthetic images are used. Their projection parameters are known and therefore the influence of false initialisation points can be examined.
Empirical tests show that if the orientation (initialisation of the rotation matrix) equals to true values, the correlation should provide suitable matches. However, deviations in one single orientation angle of about 3 degrees, influence the correlation procedure in a negative way and probably lead to false matches if these deviations are not taken in consideration.
Furthermore, it is recommended to apply the distance calculation in object space (to prevent image scale dependencies) with a reconstructed plane from the imaged lines and the 3D lines from the point cloud. For plane reconstruction at least three points are sufficient and comprises e.g. the projection centre and two points of one imaged line.

## 6. CONCLUSSIONS

This paper presents a algorithm for the matching of captured and synthetic images, derived from a re-projected point cloud regarding a conceived re-projection centre which is almost conform to the camera's projection centre. The approach comprises line extraction methods for both image types. Afterwards, the line matching depends on projection parameters estimated by a bruteforce correlation.

The line extraction works with intensity point clouds as well as coloured point clouds. With the aid of the point's geometric informations, edges could be estimated which will be unrecognisable by common edge detection approaches in case of real images because of low contrasts. Histogram filters are used to optimize the edge images or define a relative range of lengths where the used segments have to lay inside. Thus, the histogram filter defines the level of detail in the resulting edge image and the line matching uses the main edges only.

Future works will enhance the extraction process in case of edges pointing in depth direction by further investigations regarding the neighbourhood of segments.

Currently, the geometric matching gets best results including approximated position and orientation values for the captured image. Moreover, for more robustness due to roughly approximated


Figure 11: Schematic representation of the line matching result for the principalship with a smartphone (left) and a synthetic image (right). Matched lines are shown by coloured connections of their starting points.
input parameters, an additional window-based correlation procedure is used but more investigations in terms of the orientation could provide enhancements concerning this topic too (e.g. by definition of correlation intervals with more than three dimensions). Otherwise, a correlation which refers to the orientation and line directions parameters only, could be done before correlating their positions.
To eliminate the influence of the image scale, the calculation of the distance between two lines has to be done in the 3D space.

For structure influence reduction and line closeness, the main edges (like the longest lines) should be used for the matching only and could be combined with the histogram filter method for a so-called pyramid matching. The first level of the pyramid will be given by matching the longest edges. The resulting image position and orientation will be refined by invoking further lines.

## References

Becker, R., Benning, W. and Effkemann, C., 2004. Kombinierte auswertung von laserscannerdaten und photogrammetrischen aufnahmen. In: Zeitschrift für Geodäsie, Geoinformation und Landmanagement, 5/2004, pp. 347-355.

Bradski, G., 2000. The opencv library. In: Dr. Dobb's Journal of Software Tools, 25(11), pp. 120-126.

Catalano, D., 2016. Catalano-framework. https: //github.com/DiegoCatalano/Catalano-Framework (06 Apr. 2016).

GSMArena, 2016. Samsung galaxy S 5 datasheet. http://www . gsmarena.com/samsung_galaxy_s5-6033.php (06 Apr. 2016).

Meierhold, N., Bienert, A. and Schmich, A., 2008. Line-based referencing between images and laser scanner data for imagebased point cloud interpretation in a cad-environment. In: The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Beijing, China, Vol. XXXVII, Part B5, pp.437-443.

Meierhold, N., Spehr, M., Schilling, A., Gumhold, S. and Maas, H.-G., 2010. Automatic feature matching between digital images and 2D representations of a 3D laser scanner point cloud. In: The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Newcastle upon Tyne, UK, Vol. XXXVIII, Part 5, pp. 446-451.

Niemeier, W., 2008. Ausgleichungsrechnung - Statistische Auswertemethoden. Walter de Gruyter GmbH \& Co. KG., Berlin.

Prewitt, J. M. S., 1970. Object enhancement and extraction in Picture processing and Psychopictorics. Academic Press, New York, pp. 75-150.

Szeliski, R., 2010. Computer vision: algorithms and applications. Springer Science \& Business Media, New York.

Tomasi, C. and Manduchi, R., 1998. Bilateral filtering for gray and color images. Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India. http://homepages.inf.ed.ac.uk/rbf/CVonline/ LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html (10 Apr. 2016).

