REFERENCING OF IMAGES TO LASER SCANNER DATA USING LINEAR FEATURES EXTRACTED FROM DIGITAL IMAGES AND RANGE IMAGES

N. Meierhold[a, *], A. Schmich[b]

[a] Technische Universität Dresden, Institute of Photogrammetry and Remote Sensing, 01062 Dresden, Germany - Nadine.Meierhold@tu-dresden.de
[b] kubit GmbH, Fiedlerstraße 36, 01307 Dresden, Germany - armin.schmich@kubit.de

Commission III, Workshop Laserscanning

KEY WORDS: Close range photogrammetry, terrestrial laser scanning, image orientation, line extraction, range image

ABSTRACT:

This paper deals with the referencing of single images taken independently from the laser scanner (camera not mounted on the scanner) to a laser scanner point cloud of the same scene. The single image orientation is a standard task in photogrammetric applications and can be performed by a classical photo resection. Thereby, the determination of the exterior orientation is realised indirectly by measuring imaged object points in the photo. In architectural applications, linear features can often be extracted easier than points. Hence, the image orientation in this paper is based on linear features instead of point features.

To decrease the effort of user interaction for feature measurement, this paper will present a method for line extraction from image data, which are eligible for image orientation. The first step of line extraction is the edge detection using a Canny edge detector. Afterwards, the resulting 1 pixel wide edges are vectorised in parts of straight lines by a customised Douglas-Peucker algorithm. The presented method is applied to extract 2D lines from digital image data and 3D lines from laser scanner data. At this, the basis for the 3D line extraction is a range image and an intensity image generated from the point cloud.

The line extraction is applied to an image of a building facade, which subsequently is oriented with different levels of user interaction involving the extracted lines. To validate the results, the image orientation additionally was obtained with existing control points.

1. INTRODUCTION

The geometric referencing of digital image data and 3D point clouds e.g. given by a terrestrial laser scanner is the prerequisite for different levels of integrated data interpretation like texture colouring of the point cloud for visualisation purposes, interactive object modelling by monoplotting-like procedures or automatic point cloud interpretation. The referencing can be realised by classical photo resection or a line-based image orientation. A main reason for the usage of linear features is that discrete points are often not represented by a laser scanner point caused by the sub-sampling characteristics of the laser scanner. Dependent on the point spacing, an improvement of the orientation accuracy is expected using linear features.

The image orientation method is implemented in the software product *PointCloud* of the company *kubit GmbH,* which is a runtime extension of *AutoCAD* and allows a combined interpretation of digital images and laser scanner point clouds. All user interactions concerning the image orientation are carried out in the CAD environment. These interactions include the measurement of point or line features as well as the approximate setting of the camera position and the viewing direction to the object.

In view of a fully automatic image orientation, a first step is the automation of the feature extraction. Concerning image processing, there was a lot of research work in Computer Vision. Two important areas are the edge detection in a grey level image and the transformation to more symbolic features such as lines (Loh et al., 2001). Established methods for detecting edge pixels are based on analysing the maxima of the first derivative (Canny, 1986) or on the analysing of zero-crossings of the second derivative (Marr & Hildreth, 1980). Afterwards, these edge pixels can be transformed to lines by methods such as the Hough transform, edge linking or line approximation.

For the purpose of line-based image orientation, there have to be extracted lines from the digital image data, which can be located in the 3D point cloud as straight lines. This comprises physical boundaries of the imaged objects and other visual distinctive edges, which also cause differences in the intensity values obtained by the terrestrial laser scanner. The image line extraction in this paper is realised by a Canny edge detector and a method for line extraction belonging to the group of line approximation.

Concerning the automation of 3D line extraction from laser scanner point clouds, Alshawabkeh et. al. (2007) implemented a method which analyses the curvature of the object using a range image of the point cloud. The idea of analysing the range image is picked up in this paper, while, as first approach, the range image is processed with the same algorithm implemented for the line extraction from digital images.

2. LINE-BASED IMAGE ORIENTATION

The orientation of a single image primarily contains the determination of the exterior orientation. That includes the orientation angles (ω,φ,κ) and the position of the perspective

[*] Corresponding author

centre (X_0, Y_0, Z_0) in relation to the object reference system given by the laser scanner data. In case of using point features, the image orientation can be realised by a classical photo resection.

Meierhold et. al. (2008) showed that line features are equally well or even better suited for the referencing between images and laser scanner data than discrete points, what is additionally enhanced by an increasing point spacing. Two different methods for line-based image orientation were tested: a line-to-line approach (Schwermann, 1995) and a point-to-line approach (Schenk, 2004). Because the method using correspondences between image points and object lines performed better, it is the basis for the image orientation of our further work.

Using the point-to-line approach, the image orientation is based on the definition of straight object lines in the laser scanner point cloud and on the measurement of points on corresponding image lines. The object lines are described by a 4-parameter representation (Roberts, 1988).

For a correct orientation, the projection ray from the perspective centre (Fig. 1 *O*) through the image points should intersect the corresponding object lines given by points P_1 and P_2 in Fig. 1.

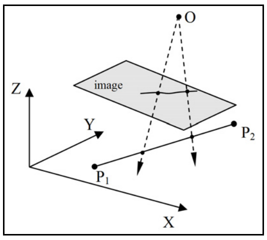

Figure 1. Principle of line-based image orientation

This geometry can be described mathematically by extending the collinearity equations in such a way that the object coordinates (*X,Y,Z*) are substituted with the respective row of the line equation:

$$x = x_0 - c \cdot \frac{r_{11} \cdot (X - X_0) + r_{21} \cdot (Y - Y_0) + r_{31} \cdot (Z - Z_0)}{r_{13} \cdot (X - X_0) + r_{23} \cdot (Y - Y_0) + r_{33} \cdot (Z - Z_0)} + dx$$

$$y = y_0 - c \cdot \frac{r_{12} \cdot (X - X_0) + r_{22} \cdot (Y - Y_0) + r_{32} \cdot (Z - Z_0)}{r_{13} \cdot (X - X_0) + r_{23} \cdot (Y - Y_0) + r_{33} \cdot (Z - Z_0)} + dy \qquad (1)$$

$$\text{with} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_s \cos\alpha \cos\theta - Y_s \sin\alpha + t \cos\alpha \sin\theta \\ X_s \sin\alpha \cos\theta + Y_s \cos\alpha + t \sin\alpha \sin\theta \\ - X_s \sin\theta + t \cos\theta \end{pmatrix}$$

where r_{ij} = elements of rotation matrix
c, x_0, y_0 = interior orientation
dx, dy = imaging errors
x, y = coordinates of image point
X_s, Y_s = positional line parameter
α, θ = orientation line parameters

3. IMAGE-LINE EXTRACTION

The aim of image-line extraction is to localise physical boundaries of objects or visual distinctive edges in the digital image data. For the purpose of image orientation, long edges with unchanging curvature direction should be detected.

3.1 Edge detection

Discontinuities in the image signal (Fig. 2 a) are often localised by analysing the first derivative or the second derivative of the signal.

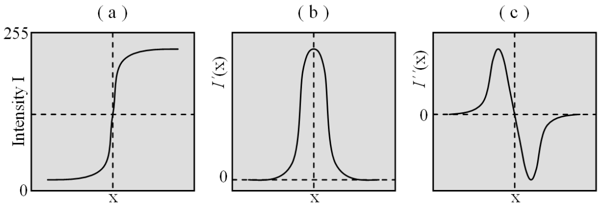

Figure 2. First derivation (b) and second derivation (c) of an edge function (a)

An edge causes a maximum in the first derivation (Fig. 2b) and a zero-crossing in the second derivation (Fig. 2c). Therefore, edges can be localised by finding local maxima in the first derivative or zero-crossings in the second derivative of the signal.

In this paper, the gradient-based edge detector presented by Canny (1986) is used, which is based on maximum searching in the first derivation. In a simple version the edge detection contains of the following steps:

1. calculation of grey scale image from RGB image by e.g. averaging the three channels (if necessary)
2. noise reduction by convolving the image with a Gaussian filter
3. calculation of gradient magnitude and direction for each pixel with a Sobel operator
4. thinning edges by non-maximum suppression
5. hysteresis: detect and link edges using two thresholds

In a discrete image, the gradient can be calculated from the grey value differences of adjacent pixels. The used Sobel difference operator approximates $\partial I / \partial x$ and $\partial I / \partial y$ with the masks:

$$S_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \qquad S_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix} \qquad (2)$$

The gradient magnitude and direction result from the following equations:

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \qquad \varphi = \tan^{-1}\left(\frac{\partial I}{\partial y} \bigg/ \frac{\partial I}{\partial x}\right) \qquad (3)$$

Fig. 3 shows the coloured digital image of a building facade and the respective grey scale image, which is the input for edge detection. Convolving the grey scale image with the Sobel masks of Eq. 2 result in the gradient magnitude image of Fig. 4 (left).

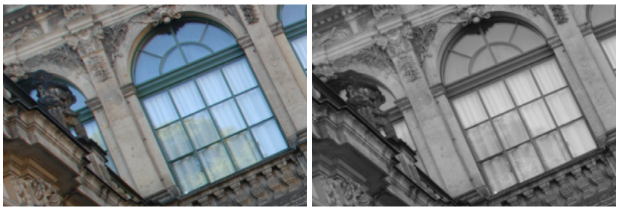

Figure 3. Digital image of a building facade (left) and grey scale image (right)

To locate the maximum of the gradient magnitude and consequently the position of the edge, the broad ridges in the magnitude array are thinned by non-maximum suppression. In this step, each pixel in the magnitude image is compared to the pixels in a defined neighbourhood. If the magnitude value of the current centre pixel is not greater than the values of adjacent pixels along the gradient line, it is set to zero. The result of non-maximum suppression is a thinned magnitude image with one pixel wide ridges.

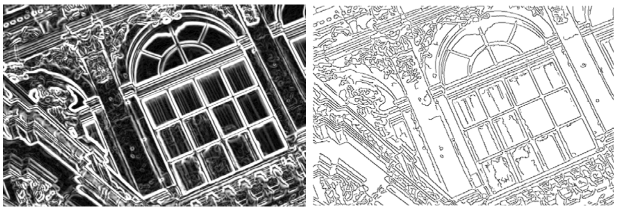

Figure 4. Image of gradient magnitude (left) and resulting inverted edge image after thresholding (right)

To get the final edge map, the map of local maxima is thresholded using two preset values T_1 and T_2. In this edge linking process, all pixels with magnitude values above the higher threshold T_2 are marked as edge pixels. Based on these pixels, adjacent pixels are linked to the same edge, if its magnitude is greater than the lower threshold T_1 (Fig. 4, right).

To reach a higher level of automation, the presetting of thresholds can be avoided by analysing the histogram of the local maxima map for determination of the thresholds T_1 and T_2.

3.2 Edge selection and refinement

To detect connected regions from the edge map, a connected component analysis was implemented. Afterwards, these connected regions are analysed for changes in edge direction to refine the regions and consequently to get better conditions for the following vectorisation.

3.2.1 Connected component analysis: The connected component analysis (alternatively connected component labelling) is an algorithm in computer vision to determine connected regions in raster images. But it can also be applied to other data types.

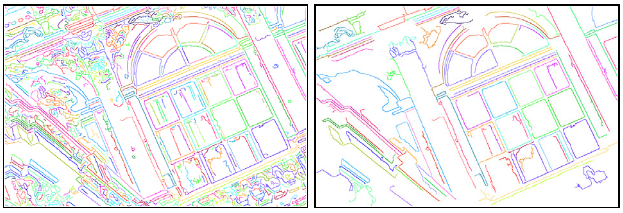

Figure 5. Resulting regions after labelling (left) and remaining regions after elimination (right)

In our case, it is applied to the edge map which is a binary image depicting white edge pixels on a black background. To generate connected regions, the 8-neighbors of each edge pixel are analysed. Beginning with the upper left corner, the edge map is linear scanned for edge pixels. If a white pixel is found, the current pixel and all white pixels in the defined neighbourhood get the same label. Then the algorithm continues with the found neighbours until no further pixel can be found belonging to the same region. Subsequently, the next unlabeled edge pixel is searched. The labelling of the regions is shown in Fig. 5 (left) in different colours.

Because digital images contain a lot of information, there are many small edge regions (e.g. caused by ornaments or textures) which are not eligible for image orientation. Such regions are eliminated in the next step by analysing the axially parallel bounding box of each region. If the diagonal of the bounding box is smaller than a preset value C_1 the region is eliminated (Fig. 5 right). The threshold is chosen on behalf of the minimum edge length that has to be detected by the algorithm.

3.2.2 Refinement of regions: To revise the regions and to separate different line parts included in one region, a refining step is necessary which analyses the direction of the edge in any pixel.

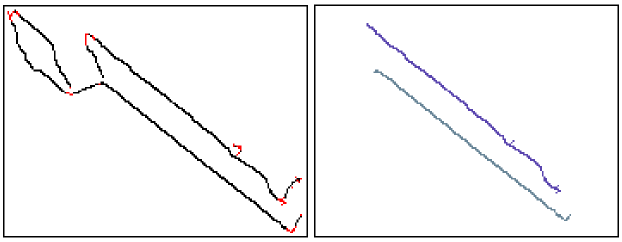

Figure 6. Refinement of two regions: regions with red break pixels (left) and revised regions (right)

For each pixel, the connected pixels in a 9x9 neighbourhood are used to adjust a straight line. The obtained line direction is assigned to the current pixel. Afterwards, for each pixel the variation of the direction is calculated comprising the edge pixels in a 3x3 neighbourhood to find abrupt changes in edge direction. Pixels with a standard deviation above a threshold R_1 are marked (Fig. 6 left) and the connectivity analysis is started again without the marked pixels (Fig. 6 right). The following image shows the remaining main edges from Fig. 5.

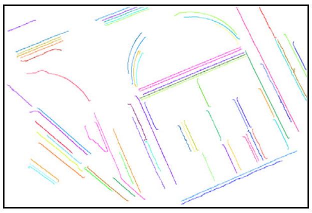

Figure 7. Resulting labelled edges detected from digital image

3.3 Vectorisation of edge regions

For the purpose of image orientation using the method presented in section 2, points on image lines are required. Therefore, the next step deals with the vectorisation of detected edges. The implemented vectorisation method operates region-by-region to describe the edges by parts of straight lines. To generate these polylines, a customised Douglas-Peucker algorithm is used.

3.3.1 Principle of Douglas-Peucker: The Douglas-Peucker algorithm is a splitting method for curve fitting independently suggested by Ramer (1972) and Douglas & Peucker (1973). Beginning with the line, which primarily connects the start and end point of the chain, the distance d of all other points lying within the line is calculated (Fig. 8 step 1). At the point with the largest distance (point P_1), the initial line is splitted if the distance exceeds a predefined approximation error ε. The iterative process is continued until no further break point can be found.

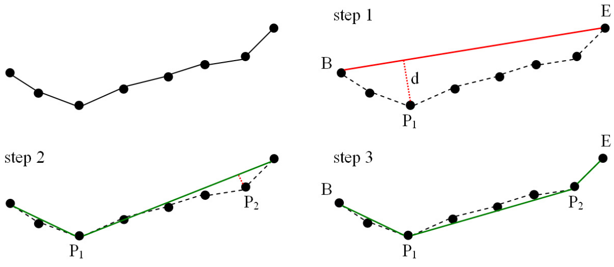

Figure 8. Principle of Douglas-Peucker algorithm

3.3.2 Determination of start and end pixel: The shape of the regions to be processed is varying, so that choosing the first and last pixels of an edge as start and end points is not always applicable (e.g. in case of closed edges like circles). To obtain a complete vectorisation as good as possible, the start and end points should have the greatest possible distance to each other. Therefore, the axially parallel bounding box of the current edge region is analysed (Fig. 9 green) to detect the axial direction with the biggest extension of the edge. Then the edge pixels, which define the bounding box in this direction, become the start and end points (Fig. 9 red pixels inside circles), where the start pixel is defined as the lower one.

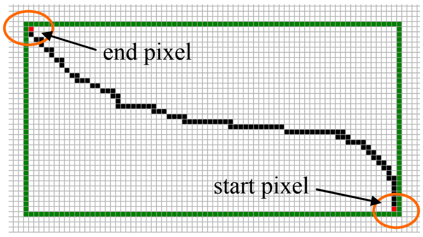

Figure 9. Determination of start and end pixels for Douglas-Peucker vectorisation

3.3.3 Splitting: For extraction of line segments which are eligible for image orientation, the principle of Douglas-Peucker algorithm was enhanced for some conditions:

- direction condition
- connectivity condition
- length condition

The direction condition influences the functionality of the Douglas-Peucker algorithm while the other two conditions evaluate the detected line parts or polylines before saving.

Direction condition: Before each splitting step, the azimuths from the start to the end point (Fig. 10 red) and from the start point to the potential breakpoint (Fig. 10 green) are calculated.

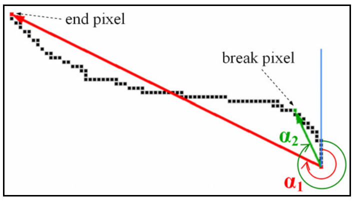

Figure 10. Azimuth calculation for direction condition

The size of the azimuths defines on which side of the initial line the breakpoint resides. On the right side, the azimuth to the breakpoint is larger than to the end point and on the left side it is smaller. At the first splitting step the direction of the breakpoint is memorised and all further breakpoints have to reside on the same side. This condition is suggested because only polylines with an unchanging sign of curvature are required for image orientation.

Connectivity condition: If no more splitting is necessary, because all pixel distances to the line part are lower than the predefined value ε, the line part is stored in a list. Before it is stored, the line part is checked while the pixels assigned to the line part are connected. By this connection a gap of two pixels is allowed.

Length condition: After vectorisation of the entire edge, the length of the resulting polyline is checked. It is calculated by summing up the length of the several line parts. If the total length is smaller than the minimum edge length already used in section 3.2.1, the polyline is rejected.

4. 3D-LINE EXTRACTION USING RANGE AND INTENSITY IMAGES

The idea is to use the presented method for 2D line extraction from digital images for the 3D straight line extraction from laser scanner data. For this purpose, a 2D representation of the point cloud is calculated in the form of an intensity image and a range image. These images are processed with the method described in section 3. Afterwards, the 2D lines are transferred in 3D object space.

4.1 Generation of range and intensity array

The range and intensity arrays are calculated from the polar coordinates acquired with the laser scanner. This coordinates include the vertical angle θ, the horizontal angle φ and the range r. The image format is defined by the angles, where the image height is given by the range of the vertical angle and the image width by the range of the horizontal angle. The number of pixels depends on the scan resolution in both directions.

Because the intensity ranges between 0 and 255, for storage of the intensity image an 8 bit array is sufficiently. The pixel value v_I of the intensity image is calculated from Eq. 3 which corresponds to a histogram stretching to enhance the image contrast.

$$v_I = \frac{255}{I_{\max} - I_{\min}} \cdot (I - I_{\min}) \tag{3}$$

where I_{min}, I_{max} = smallest and greatest intensity value
I = measured intensity value

Usually, an 8 bit image is unsufficiently to code the whole measured range distance[1]. Therefore, the range array is stored with a radiometric resolution of 32 bit. A range resolution of 1 mm effect an overly high variance of the pixel values which would disturb the edge detection. Therefore, the range value r is linear transformed in distance steps v_r according to the accuracy of range measurement (Eq. 4).

$$v_r = \frac{1}{\sigma_r} \cdot (r - r_{\min}) \tag{4}$$

where σ_r = accuracy of range measurement [mm]
r = measured range value [mm]
r_{min} = smallest range of the current point cloud [mm]

[1] with a geometric resolution of 1 mm, only a range distance of about 2.5 m could be displayed

If there is more than one laser scanner point per pixel, the range and the intensity values of the laser scanner point with the smallest range are stored. For the later step of 3D line extraction, additionally the respective Cartesian coordinates of the laser scanner point are stored in a separate array with the same dimensions as the arrays.

4.2 Extraction of 3D lines

The range and the intensity image are processed independently. At first 2D lines are extracted using the presented method.
Afterwards, every vertex of the 2D polylines is transferred to 3D object space. This can be realised easily, because the elements of the range and intensity arrays correspond to the elements of the array of the Cartesian 3D coordinates. If a polyline consists of more than two vertices, the 3D straight line is obtained by least squares adjustment.

5. RESULTS

5.1 Data set

The methods were tested with a data set taken from the building facade of the "Kommode" in Berlin. The 3D point cloud was obtained with the terrestrial laser scanner Riegl LMS Z-420i. The part of the point cloud, which corresponds to the image scene, has a total of 1 Million points with a varying point spacing from 12 mm to 35 mm (depending on the scanning distance). The digital image data was acquired with a 6 Mpix mirror reflex camera from Fuji with 5.4 µm pixel size. Additionally, signalised and natural control points were obtained using a reflectorless measuring total station.

5.2 Results of line extraction

parameter	digital image	range image	intensity img.
T_1	$0.4*T_2$	$0.4*T_2$	$0.4*T_2$
T_2	140	80	60
C_1	60 pixel	40 pixel	40 pixel
R_1	20 deg	20 deg	20 deg
ε	1 pixel	1 pixel	1 pixel

Table 1. Preset values for processing the dataset

Applying the presented method, a total of about 1200 polylines were extracted from the selected digital image and a total of 190 3D straight lines from the point cloud (142 from the range image and 38 from the intensity image). The Tab. 1 summarises the used preset values for processing the data set. The range image was generated using Eq. 4 with $\sigma_r = 7$ mm. This value was determined in the course of verifications concerning the accuracy of the Riegl LMS Z-420i (Teschke, 2004).
The extracted polylines and 3D straight lines are then imported in the software product *PointCloud* (Fig 11), where a postprocessing step follows. Often there are edges in the image or in the point cloud which are represented by more then one line. These separate collinear lines are merged interactively in *PointCloud*. In the case of merging 3D straight lines an adjusted 3D line is calculated involving the end points of the merged lines.
To avoid user interaction, an approach suggested by (Boldt et al., 1998) and (Mohan & Nevatia, 1989) can be implemented for an automatic merging of collinear line parts.

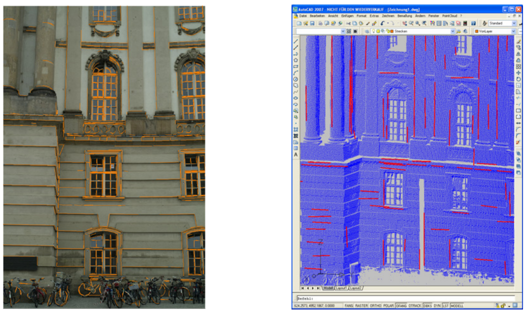

Figure 11. Image overlaid with extracted 2D polylines in orange (left) and point cloud overlaid with extracted 3D straight lines (right)

Because the line extraction algorithm do not recognise which edges or lines belong to the facade, there are many lines which are not eligible for image orientation e.g. in the area of the bicycles. Hence, appropriate features were selected and the correspondences between 2D and 3D lines were assigned interactively in the CAD environment.

5.3 Results of image orientation

After line extraction the exterior orientation of the image is calculated. Because the interior orientation of the camera is unknown, it is determined simultaneously together with the radial distortion parameters A_1 and A_2 (Brown, 1971).
At first, the image was oriented by a classical photo resection using 11 signalised and 9 natural control points. Additionally, four different versions of line-based image orientation were tested while the way of feature extraction varies. Table 2 summarises the different methods, the numbers of involved line features and the resulting redundancy. The workflow of the image orientation in *PointCloud* was already shown in (Meierhold et. al., 2008).

name	extraction of 2D lines	3D lines	number of lines	redund -ancy
method 1	interactive	interactive	28	45
method 2	automatic	interactive	33	60
method 3	interactive	automatic	13	26
method 4	automatic	automatic	23	61

Table 2. Tested variations of line-based image orientation

With all the methods the determination of the image orientation was possible, but the results are different. The comparison of the classical photo resection and the resection methods using linear features are shown in Tab. 3 and Tab. 4.
From the standard deviation it is obvious, that the methods 1 and 2 using interactively measured object lines as well as the point-based resection yield nearly the same orientation results. The surprisingly low accuracy of point-based orientation can be caused by an insufficient distribution of the points in the image and by the fact, that the control points were acquired by reflectorless measurement. Because many control points are located on object edges, the accuracy of range measurement can be unreliable, depending on the incident angle of the laser beam sent by the total station.

method	s_0 [pixel]	internal standard deviations of unknowns								
		s_{X0} [m]	s_{Y0} [m]	s_{Z0} [m]	s_ω [°]	s_φ [°]	s_κ [°]	s_c [mm]	s_{x0} [mm]	s_{y0} [mm]
point-based	2.30	0.278	0.034	0.066	1.069	1.235	1.025	0.3921	0.2545	0.2055
method 1	3.13	0.341	0.055	0.065	1.470	1.485	1.427	0.5014	0.2898	0.2805
method 2	3.06	0.296	0.046	0.063	1.231	1.112	1.191	0.4309	0.2477	0.2565
method 3	3.43	2.353	0.453	0.344	5.894	6.051	5.686	3.0865	1.3527	1.2319
method 4	4.15	1.422	0.526	0.363	3.133	3.056	3.097	1.9343	1.4053	0.8563

Table 3. Standard deviations of image orientation by applying the five resection methods

method	differences of estimated parameters to point-based resection								
	X_0 [m]	Y_0 [m]	Z_0 [m]	ω [°]	φ [°]	κ [°]	c [mm]	x_0 [mm]	y_0 [mm]
method 1	0.997	-0.031	-0.194	-0.152	0.354	0.257	1.605	0.130	-0.006
method 2	0.409	0.032	-0.147	0.255	1.248	-0.207	0.838	0.772	-0.134
method 3	-1.999	-1.203	0.569	-5.263	1.021	5.964	-2.725	0.612	-1.092
method 3	-6.510	-0.045	1.909	4.441	8.799	-4.089	-9.600	5.738	-3.283

Table 4. Comparison of orientation parameters obtained by point-based and line-based single photo resection

The poorest results yield the line methods based on automatically extracted object lines. Look at an extracted 3D line more in detail, the reason is obvious (Fig. 12). The accuracy of the 3D lines obtained with the presented algorithm is insufficient for the purpose of image orientation. But the method can be used to obtain good approximations for lines in the point cloud, so that the search area can be reduced.

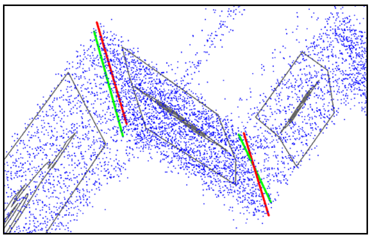

Figure 12. Positional accuracy of automatically extracted 3D lines (green) with 3D line resulting from plane intersection (red)

6. CONCLUSIONS

The presented automatic approach for image-line extraction is a suitable algorithm to determine image-lines for the purpose of image orientation. The results showed, that line features are a valuable option for the referencing between image data and laser scanner point clouds. Accuracy-wise, the point-based resection as well as both line-based methods perform almost identical using interactively measured object lines. One advantage of the automatic line extraction is that more features are extracted so that more observations are involved in the orientation process. This is obvious from the high redundancy of the second tested line method. Another important advantage of the implemented line extraction approach is that lines do not have to be imaged as straight lines. In contrast e.g. to the Hough transform, the algorithm is able to extract curved lines. That means there is no necessity for determination of lens distortion and resampling of the image before image orientation.

The presented approach for 3D line extraction only yields approximate positions of object lines, so that only an approximate exterior orientation of the image could be obtained. But analysing the range and intensity images generated from the point cloud is a reliable option to reduce the search area for 3D line extraction from the point cloud: The approximate positions of 3D lines can be detected in the image data. Subsequently, the point cloud in the surrounding can be extracted and analysed to obtain the accurate positions of the 3D lines e.g. by plane fitting and intersection.

ACKNOWLEDGEMENT

The research work is funded by resources of the European Fund of Regional Development (EFRE) and by resources of the State Saxony within the technology development project “CAD-basierte bildgestützte Interpretation von Punktwolken”.
Initiated from the Humboldt-Universität zu Berlin represented by the engineering department, the used data set of the “Kommode” was acquired in cooperation with the Brandenburg University of Technology, chair of surveying.

REFERENCES

Alshawabkeh, Y., Haala, N. & Fritsch, D., 2007. Registrierung terrestrischer Bild und LIDAR Daten für die Dokumentation von Kulturdenkmälern. *Photogrammetrie - Fernerkundung - Geoinformation (PFG)*, Heft 3, pp.199-208

Boldt, M., Weiss, R. & Riseman, E., 1989. Token-based extraction of straight lines. *IEEE Transactions on Systems, Man, and Cybernetics*, 19 (6), pp. 1581-1594.

Brown, D.C., 1971. Close-range camera calibration. *Photogrammetric Engineering*, 37 (8), pp. 855-866

Canny, J., 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8 (6), November 1986, pp. 679-698.

Douglas, P. & Peucker, T., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica*, 10 (2), pp. 112-122.

Loh, A.W.K., Robey, M.C. & West, G.A.W., 2001. Analysis of interaction between edge and line finding techniques. *Pattern Recognition*, 34 (6), June 2001, pp. 1127-1146

Marr, D. & Hildeth, E., 1980. Theory of edge detection. *Proc. Roy. Soc. London*, Vol. 207 of Series B, pp. 187-217

Meierhold, N., Bienert, A. & Schmich, A., 2008. Line-based referencing between images and laser scanner data for image-based point cloud interpretation in a CAD-environment**.** *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science,* Vol. 37, part B5, WG V/3, pp. 437ff.

Mohan, R. & Nevatia, R., 1989. Segmentation and description based on perceptual organization. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (1989), San Diego, CA, pp. 333-341.

Ramer, U., 1972. An iterative procedure for the polygonal approximation of the plane curves. *Computer Graphics and Image Processing*, 1 (3), pp. 244-256.

Roberts, K. S., 1988. A New Representation for Lines. *IEEE Proceedings of Computer Vision and Pattern Recognition*, (1988), pp. 635-640.

Schenk, T., 2004. From point-based to feature-based aerial triangulation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58 (2004), pp. 315-329.

Schwermann, R., 1995. Geradengestützte Bildorientierung in der Nahbereichsphotogrammetrie. Dissertation, *Veröffentlichung des Geodätischen Instituts der Rheinisch-Westfälischen Technischen Hochschule Aachen*, Nr 52.

Teschke, A., 2004. Genauigkeitsuntersuchung des terrestrischen Laserscanners Riegl LMS-Z420i. Diplomarbeit, *Technischen Universität Dresden, Institut für Photogrammetrie und Fernerkundung*