SINGLE IMAGE ORIENTATION USING LINEAR FEATURES AUTOMATICALLY EXTRACTED FROM DIGITAL IMAGES

Nadine Meierhold[a], Armin Schmich[b]
[a] Technical University of Dresden, Institute of Photogrammetry and Remote Sensing
[b] kubit GmbH, Fiedlerstraße 36, 01307 Dresden
Email: Nadine.Meierhold@tu-dresden.de, armin.schmich@kubit.de

Abstract: As a first step for an automatic image orientation the paper will present a method for line extraction from digital image data. The extracted lines are used to orient an image to a laser scanner point cloud of the same scene. The image-line extraction consists of two parts – an automatic one and an interactive one: At first lines are detected, vectorised and afterwards visualised in the software product *PointCloud*. Since there may be some extracted lines, which are not eligible for image orientation, in the second step the user selects only appropriate features and measures corresponding 3D lines in the point cloud. The implemented method was evaluated with two different data sets taken from building facades. Thereat, the orientations of two images were determined involving either point features, manually measured or automatically extracted image-lines.

1. Introduction

The image orientation is a main task in photogrammetric applications. For single images it can be performed by spatial resection using control points, which are known in object space and measured in image space. For the purpose of a combined interpretation of digital images and laser scanner data, the geometric referencing of these data is necessary. This requires control features, which are known in the reference system of the laser scanner. Concerning an automatic process for single image orientation, the usage of linear features is more applicable than using point features because the extraction of straight lines from 3D point clouds may be considered to be easier than the extraction of discrete points.

A comparison of two different methods for line-based image orientation already was shown in Meierhold et al. [5]. Based on the arrived conclusions, the line-based image orientation in this paper is realised by an approach suggested by Schenk [8]. The image orientation using this approach is based on straight object lines, which are defined in the laser scanner point cloud and described by line equations, and on points on the corresponding image lines. Because of this point-to-line correspondence, the line extraction method has to extract image-lines which are described by single points instead of line equations.

The image orientation as well as the associated user interaction is carried out entirely in the environment of *AutoCAD* and its runtime extension *PointCloud*, which is a software product of the company *kubit GmbH*. The line extraction method, which will be described in the next section, and the line-based image orientation were implemented and tested with two different data sets. The results will be shown in section 3.

2. Line extraction method

One important step for an automatic image orientation is the automation of the feature extraction. The automatic part of the implemented method for line extraction consists of three main steps (Fig. 1), which are presented in the next sections more in detail:

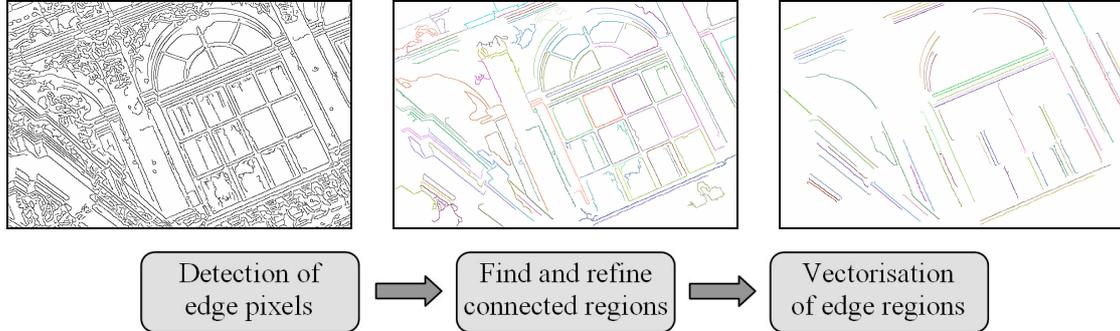| Detection of edge pixels | → | Find and refine connected regions | → | Vectorisation of edge regions |

Figure 1. Workflow of image-line extraction (left to right): inverted edge map, remaining regions after elimination, basic regions for vectorisation after refinement

The aim of the line extraction is to find long edges with unchanging sign of curvature. For the purpose of image orientation, only these lines are eligible which can also be located in the 3D point cloud (e.g. physical object boundaries). Because edge detection methods only localise discontinuities in the image signal, which can be caused by many effects (e.g. textures or shadows), there are detected many ineligible edges. The automation of a selection algorithm seems rather to be more complex. Hence, the selection of appropriate line features is realised interactively at this state of work.

2.1. Edge detection

There are a lot of methods for the detection of edge pixels from images. For the implemented method a Canny edge detector was chosen [3]. At first the gradient magnitude and gradient direction is calculated by convolving the smoothed grey scaled image with the Sobel masks of Eq. 1.

$$S_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \qquad S_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix} \tag{1}$$

Subsequently, the Canny edge detector contains the thinning of the gradient magnitude array by a non-maximum suppression and the detection and linking of edges applying the two thresholds T_1 and T_2 (hysteresis). The result is a binary edge map depicting white edge pixels on a black background.

2.2. Edge selection and refinement

To get connected edge regions, the edge map is processed with a connected component analysis using a 8-connectivity. 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.
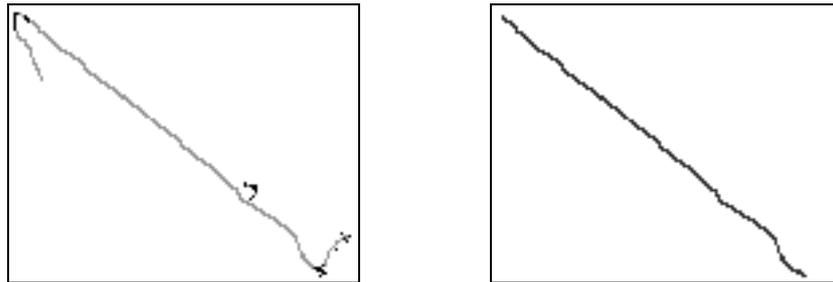
Figure 2. Refinement of an edge region: Region with black coloured break pixels (left) and revised region (right)

To revise the regions and to separate different line parts included in one region, a post-processing step analyses the changing edge direction for each pixel. 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 by comprising the directions of 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. 2 left) and the connectivity analysis is started again without the marked pixels (Fig. 2 right).

2.3. Vectorisation of edge regions

Because the image-lines have to be described by a sequence of points, the idea for the vectorisation is to simplify the edges describing them by parts of straight lines. This can be realised by a curve fitting algorithm independently suggested by Ramer [7] and Douglas & Peucker [4]. This top-down approach begins with an initial line which connects the start and end point of a pixel chain. Then the line is recursively divided at the pixel with the largest distance to the line until all pixel distances under-run a predefined approximation error ε.
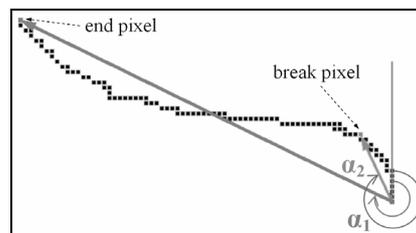

Figure 3. Calculation of azimuth for direction condition

This basic algorithm is customised by three conditions while the first one influences the functionality of the algorithm and the other two evaluate the detected line parts or polylines before saving.

Direction condition: This condition is implemented to extract only lines with an unchanging sign of curvature. Before each splitting step, the azimuths from the start to the end pixel (Fig. 3 α_1) and from the start pixel to the potential break pixel (Fig. 3 α_2) are calculated. The sizes of the azimuths define on which side of the initial line the break pixel resides. At the first splitting step the direction of the break pixel is memorised and all further break pixels have to reside on the same side.

Connectivity condition: Before a line part is stored, it is checked if the pixels assigned to this line part are connected. By this connection, a gap of at most two pixels is allowed.

Length condition: After vectorisation of the entire edge region, 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 2.2, the polyline is rejected.

In the case the polyline only consists of two vertices, the resulting 2D line is obtained by least squares adjustment involving all edge pixels whose distance to the line is below ε.

3. Results

3.1. Data sets

Two different data sets of building facades are chosen for evaluating the presented method for image-line extraction and the following image orientation based on the extracted lines. The point clouds of both datasets were acquired with the terrestrial laser scanner Riegl LMS Z-420i. For the "Nymphenbad", which is a part of the historical Zwinger in Dresden, the point cloud has a total of three Million points with a point spacing from 1.4 to 13 mm. The part of the point cloud corresponding to the chosen image data of the second data set, taken from the "Kommode" in Berlin, comprises one million points. Because of a large scanning distance the point spacing varies from 12 mm to 35 mm. The image of the "Nymphenbad" was acquired with a 6 Mpix mirror reflex camera with 7.8 μm pixel size. For the acquisition of the image data of the second data set, also a 6 Mpix mirror reflex camera was used. With 5.4 μm, it has a smaller pixel size. Additionally, signalised and natural control points were obtained on the facade of the "Kommode" using a reflectorless measuring total station.

3.2. Line Extraction

The presented line extraction method was applied to both images using the preset values summarised in the following table:

parameter	"Nymphenbad"	"Kommode"
T_1	$0.4*T_2$	$0.4*T_2$
T_2	150	140
C_1	70 pixel	60 pixel
R_1	20 deg	20 deg
ε	1 pixel	1 pixel

Table 1. Preset values for processing the data sets

A total of about 1200 lines were extracted from the image of the "Kommode" and 902 lines from the "Nymphenbad" image. From Fig. 4 is obvious that there are extracted many long lines which correspond to object boundaries so that they can also be located in the 3D point cloud. This is a basic prerequisite for the image orientation. Furthermore, it can be seen (e.g. in the area of the bicycles or columns) that the algorithm is able to vectorise curved edges and to separate edges situated at close quarters.

Disadvantagous for the image orientation concerning the assignment between 2D and 3D features is the fact that partly more than one polyline was extracted for one object edge. These collinear lines have to be merged. At this moment, it is realised interactively in the CAD environment. However, Boldt et al. [1] and Mohan & Nevatia [6] suggest algorithms for the automation of this step.
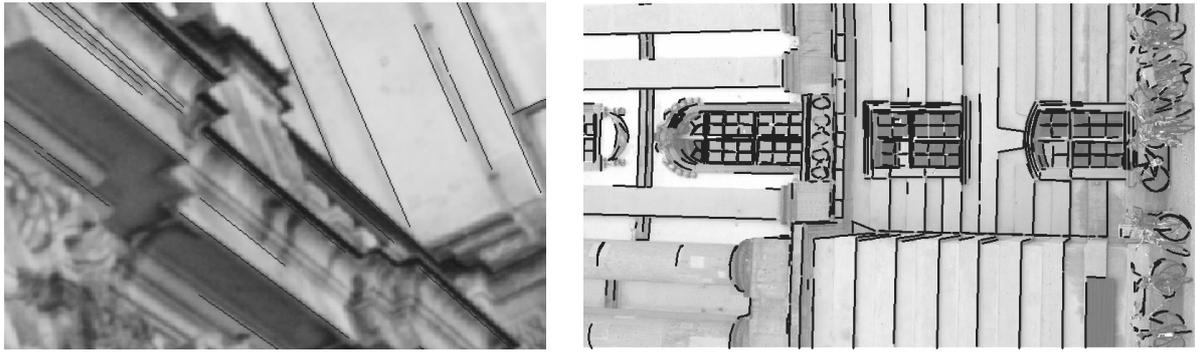
Figure 4. Detail of the "Nymphenbad" image (left) and of "Kommode" image (right) overlaid with extracted lines (black)

3.3. Image orientation

The extracted lines were imported in *PointCloud*, where appropriate features were selected for image orientation and the corresponding object lines were measured in the point cloud. For both datasets three image orientations were calculated independently using different types of features. The first one was a classical photo resection, whereat 11 signalised and 9 natural control points were used for the data set "Kommode" and 22 natural points were used for the dataset "Nymphenbad", which were manually measured in the image and in the point cloud. The line-based image orientation was performed by involving manually measured 2D lines on the one hand (method Line1) and automatically extracted 2D lines on the other hand (method Line2) [1]. The standard deviations resulting from the image orientations are summarised in Tab. 2. Additionally to the exterior orientations of the images, the interior orientations of the cameras and the radial distortion parameters A_1 and A_2 by Brown [2] were determined simultaneously for both data sets.

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]
Nymph.	Point	1.05	0.004	0.008	0.003	0.063	0.061	0.064	0.017	0.016	0.029
	Line1	1.08	0.011	0.036	0.007	0.094	0.080	0.090	0.039	0.026	0.048
	Line2	0.53	0.008	0.053	0.018	0.061	0.052	0.059	0.052	0.025	0.041
Komm.	Point	2.30	0.278	0.034	0.066	1.069	1.235	1.025	0.392	0.255	0.206
	Line1	3.13	0.341	0.055	0.065	1.470	1.485	1.427	0.501	0.290	0.281
	Line2	3.06	0.296	0.046	0.063	1.231	1.112	1.191	0.431	0.248	0.257

Table 2. Standard deviations of applied spatial photo resections

Comparing both data sets, it is noticeable that the precision of the image orientation is better for the "Nymphenbad" data set. Reasons can be more depth information to solve correlations between the exterior and interior orientation parameters, a better distribution of the features in the image or a better accuracy of the object features caused by a smaller point spacing.

Comparing the several orientation methods, it is obvious that the point-based photo resection performs only marginally better than using linear features. In case of the "Nymphenbad", it is primarily caused by the fact that the object points were measured in the point cloud directly.

---

[1] "Nymphenbad": 16 manual and 23 automatic lines; "Kommode": 28 manual and 33 automatic lines

Hence, the accuracy of the object points depends on the point spacing of the point cloud: with an increasing point spacing, the accuracy of the object points decreases. From the almost identical or slightly better precisions of method *Line2* compared to *Line1*, one can conclude that the automatically extracted image lines are eligible for the purpose of image orientation. The marginally better precision primarily results from the higher redundancy caused by the fact that the automatically extracted polylines often consist of more than two image points.

4. Conclusions

The presented method is able to extract lines from digital images which can also be located in the 3D point cloud and are therefore eligible for the purpose of image orientation. The comparison to interactively measured image-lines, performed with two different data sets, showed that almost identical or marginally better orientation results are obtainable with automatic extracted lines. To increase the accuracy, the line extraction method can be extended to a sub-pixel accurate line extraction. The ability of the presented method to extract curved lines is advantageous concerning the processing of undistorted images, where straight object lines are imaged as curved lines.

Future work will deal with an automatic selection of appropriate lines from the total of extracted image-lines and with the automation of feature extraction from the 3D point cloud.

References:

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

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

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

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

[5] Meierhold, N., Bienert, A. & Schmich, A.: 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, 2008

[6] Mohan, R. & Nevatia, R.: 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, 1989

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

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