Automation in Construction 21 (2012) 24–31

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Robust adaptive flow line detection in sewer pipes

Simon Kirstein [a,*], Karsten Müller [b], Mark Walecki-Mingers [a], Thomas M. Deserno [a]

[a] Department of Medical Informatics, RWTH Aachen University, Pauwelsstr. 30, 52057 Aachen, Germany
[b] Research Institute for Water and Waste Management, RWTH Aachen University, Mies-van-der-Rohe-Str. 17, 52074 Aachen, Germany

ARTICLE INFO

Article history:
Accepted 11 May 2011
Available online 11 June 2011

Keywords:
Asset management
Maintenance
Urban drainage
Sewers
Flow line
Side view
Image processing
Canny edge detection
Hough transform
Dijkstra's shortest path algorithm

ABSTRACT

As part of a novel approach to automatic sewer inspection, this paper presents a robust algorithm for automatic flow line detection. A large image repository is obtained from about 50,000 m sewers to represent the high variability of real world sewer systems. Automatic image processing combines Canny edge detection, Hough transform for straight lines and cost minimization using Dijkstra's shortest path algorithm. Assuming that flow lines are mostly smoothly connected horizontal structures, piecewise flow line delineation is reduced to a process of selecting adjacent line candidates. Costs are derived from the gap between adjacent candidates and their reliability. A single parameter α enables simple control of the algorithm. The detected flow line may precisely follow the segmented edges ($\alpha = 0.0$) or minimize gaps at joints ($\alpha = 1.0$). Both, manual and ground truth-based analysis indicate that $\alpha = 0.8$ is optimal and independent of the sewer's material. The algorithm forms an essential step to further automation of sewer inspection.

1. Introduction

Sewer maintenance and rehabilitation strategies are developing rapidly because many problems such as odor, blockade, and collapse continue to occur in sewer systems all over the world [9]. For instance, recent studies in Germany have shown that approximately 20% of public sewers have defects requiring short or medium term rehabilitation [1]. It can be assumed that these figures also hold for other countries where sewer systems have been built or extended in the last century; in particular, in the United States and Canada, Australia, and most other European countries. Sewer system operators must ensure a reliable disposal of wastewater and, therefore, develop strategies to establish and effectively preserve proper condition of sewer systems. Hence, closed-circuit television (CCTV) inspection systems have been established, and manual evaluation of these images has been standardized in order to improve quality management, reliability, and appropriate scheduling procedures for rehabilitation. Nonetheless, the quality of remote inspection depends essentially on the qualification and the momentary motivation of the operating personnel [10,21]. More than ten years ago, Moselhi et al. devised semi-automatic inspection systems supporting the human operators in finding more objective results [18,19]. Using a frame grabber and image processing software, they started with developing software indicating defects in the sewer system.

2. Automation in sewer inspection

As of today, sewer inspection still is performed using robots that are guided through the pipes providing optical images from the sewer's inner wall. Most frequently, the data still is inspected by a human observer, regardless of whether the imaging is provided as video vs. photography vs. laser scan, analog vs. digital, or in color vs. gray scale vs. black and white. However, novel imaging devices support complete capturing of the sewer's inner walls and enable full decoupling of imaging and evaluation. Automatically assessing the state of a sewer is a multi-stage process using image processing technologies. In the first step, regular sections are identified before the remaining areas with unknown optical events and structures are analyzed. In recent years, several approaches have been published on automation in sewer inspection and damage classification. Sinha et al. have proposed classification systems based on neuro-fuzzy algorithms [25] and morphological segmentation [24], where edges in gray scale images have been used as feature describing the cracks, and a neural network has been used as classifier. They have focused on cracks and performed experiments on rectangular sections cut out of the images, showing either a crack or not. The investigation was based on 225 images [13] from various pipes. Duran et al. have proposed a camera/laser-based profiler combined with an artificial neural network [7]. Since laser technology captures depth information, the laser profiles were used to separate different classes of sewer surface irregularities. In

[*] Corresponding author. Tel.: +49 241 8088790; fax: +49 241 8082426.
E-mail addresses: skirstein@mi.rwth-aachen.de (S. Kirstein), mueller@fiw.rwth-aachen.de (K. Müller), mwalecki@mi.rwth-aachen.de (M. Walecki-Mingers), deserno@ieee.org (T.M. Deserno).

0926-5805/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.autcon.2011.05.009

particular, longitudinal and radial cracks, holes, joints and other obstacles were differentiated using a database of 140 profiles for training and another 100 for testing. More recently, Mashford et al. have proposed a sewer image analysis module that classifies regions of interest (ROIs) as being one of a number of classes, such as a hole, corrosion, pipe connection, deposit or tree root [15,16]. They describe a pixel-based segmentation approach using support vector machines. Although a large number of training and testing cases (i.e. pixel values) were used they were all taken from a number of small rectangular patches extracted from the same 393 m section of a certain sewer [17]. Morphological techniques were used for the detection of flow line, pipe joints and adjoining defects. However, such efforts have not yet been successful. Although error rates of zero percent have been published, the algorithms have turned out to fail when applied to image data taken from urban center's real life. The reasons are twofold. First of all, the databases that have been used in previous studies are too small to represent differences in appearance of regular and damaged pattern as they occur in real life. In addition, the image processing and analysis algorithms that have been developed operate rather locally, use static thresholds and cannot adapt to a specific sewer characteristic.

3. Adaptive sewer analysis

In contrast, our approach to automatic image analysis is made "intelligent" using a-priori knowledge that is either derived

- *extrinsically* from meta-information collected with the image such as construction material, age and dimension of the sewer system, or
- *intrinsically* from image pre-analysis such as contrast calibration, positioning of joints or flow line detection.

Furthermore, our algorithm operates globally on the entire sewer taking into account all the neighboring structures and the entire characteristics of illumination and texture of the sewer wall. Fig. 1 shows the major steps of the image processing workflow. In the first step, an unwrapped sewer image (side-view) is computed and manholes, couplings, as well as the flow line are detected, since flooded areas and manholes as well as coupling regions need special adapted filters for further analysis [20]. In this paper, we present an automatic and robust algorithm for flow line detection based on side-views of a sewer. Recent work into flow line detection [17] is suggesting mathematical morphology to solve the problem of delineation. This approach, however, assumes the typical appearance of flow lines as solid dark areas on a brighter sewer wall background, which may be valid only in stonework or concrete sewers that were cleaned and dried appropriately before CCTV inspection, avoiding any water surface reflections. Hence, our approach is rather edge- than intensity-based, using local contrast as feature for segmentation, and applies image transforms on larger neighborhoods that may cover up an entire pipe. Obtaining piecewise candidates for flow line sections, we optimize the flow line – which is assumed a-priori to be rather continuous – using a global optimization scheme that covers the entire sewer.

4. Real-life inspection data

Data from real-world sewer inspections is characterized by its high variance. Different materials, illumination, degrees of abrasion and obstacles such as dirt or roots are only few of the numerous factors influencing the color, texture, and intensity of the obtained images. In order to cope with this variance, a broad compilation of sewer inspection data has been collected and used for developing and evaluating the image processing algorithms (Table 1). All data has been acquired using the Panoramo© system (IBAK Helmut Hunger, Kiel, Germany). A side-view color image is composed of all Joint Photographic Experts Group (JPEG) files from the Audio Video Interleave (AVI) container of the Panoramo system. Usually, one - meter of side-view is coded with about 500 × 500 pixels. Hence, the source images may have a dimension of 500 × 60,000 pixels representing a sewer 120 m in length.

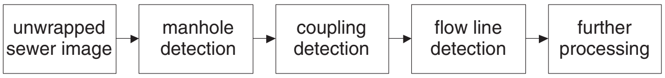

Fig. 1. Workflow of automatic pipe inspection system.

Fig. 2 emphasizes the difficulties one has to face when aiming at designing a robust algorithm for automatic delineation of flow lines. In each panel, a piece approximately two meters in length has been cut out from a sewer showing its inner surface in an unwrapped side-view. The left column (Panels a, f, k) displays concrete pipes at different contrast levels. Although the flow line appears darker than the sewer wall, it may not be delineated correctly if the gray scale intensity is used as feature for segmentation (Panel a). Columns two to five show examples from stonework sewers. The rightmost column (Panels e, j, o) displays artifacts from flashlight, where bright reflexes are obtained within the flow line. Here, intensity-based approaches will surely fail since brightness within the flow line and the sewer's wall is inverted. But also in the other examples, intensity-based segmentation clearly is inappropriate. Furthermore, Panels d, i, and n (fourth column) suggest that edge-based techniques may fail either, if the flow line is considered locally within a short part of the entire sewer. Hence, successful approaches will need to consider edges and flow lines from neighboring parts, assuming a rather constant flow of waste water.

5. Robust flow line detection

To identify the flow line, an approach that simulates the behavior of a human observer is proposed: horizontally oriented structures that differ in intensity level to surrounding areas are observed automatically. The borders of these structures, which are defined by a strong local gradient, are considered as borders of the flow line. In the following, we refer to *sewer*, *pipe*, and *segment*, when describing the entire water line, the physical piece between adjacent couplings and equidistant virtual parts within a pipe, respectively. Therefore, our approach intrinsically copes with inlets and outlets, as they may change the water level within a pipe.

In terms of image processing, we refer to edge detection when finding the maximum local gradient perpendicular to the flow direction [2]. However, edge detection delivers a binary edge map that is not necessarily composed of straight connections between the beginning and ending of a segment. Applying the Hough transform [12,6] to the edge maps provides straight line candidates as well as their probability. To determine an appropriate sequence of candidates, Dijkstra's shortest path algorithm [5] is applied after converting the problem to a graph representation. The entire image processing pipeline (Fig. 3) is described in detail in the following subsections.

Table 1
Pipe inspection image material used.

Pipe diameter	Material [m]			Total length
[mm]	Concrete	Plastic	Stoneworks	[m]
≤250	671	234	22,597	23,502
251–350	4294	841	7403	12,538
351–450	683	172	1837	2692
451–550	997		285	1282
551–650	1060		656	1716
>650	3777		37	3814
Total length [m]	11,482	1247	32,815	45,544

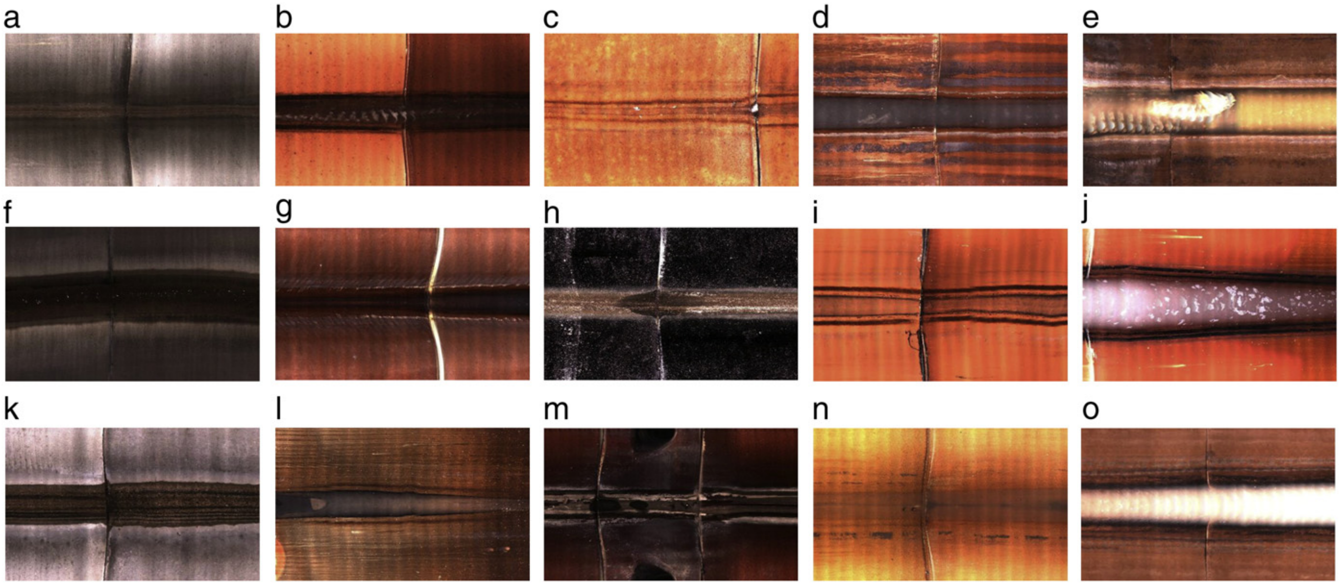

Fig. 2. Variety of flow line appearances.

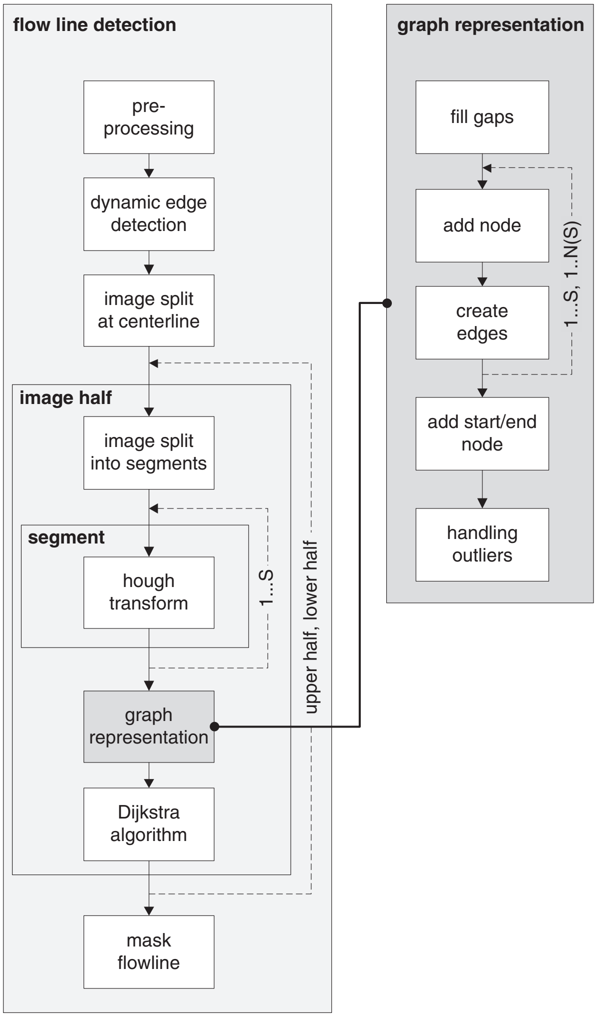

Fig. 3. Workflow of flow line detection.

5.1. Pre-processing

In the first step, the images are converted from red, gree, blue (RGB) color space into gray scale. This conversion was made adaptive to the pipe material, which is known from meta-information. Stoneware sewers, which are mainly red colored, tend to have the most significant information in the R-channel. For concrete and polyethylene sewers, all channels are included into gray scale conversion. To reduce the noise, anisotropic Gaussian smoothing is applied. Here, the horizontal dimension is set sixty times larger than the filter kernel's height maintaining horizontal structures (Fig. 4b).

5.2. Dynamic edge detection

Extracting meaningful edges from a variety of sewers (cf. Fig. 2) requires an adaptive algorithm that is based on Canny edge detection [2] according to an implementation by Deriche [3]. The Canny algorithm derives a local gradient and suppresses non-maxima following a hysteresis: the higher threshold must be exceeded before a new edge is generated, but a gradient exceeding the lower threshold is sufficient to follow an already existing edge line. As a result, a binary edge map is obtained, where the thickness of edges in an eight-connected neighborhood is exactly one.

According to the sewer dimension, we assume a minimal number of edge pixels and automatically adapt the hysteresis such that the resulting edge map contains a pre-defined number of edge pixels (Fig. 4c). Finally, the edge map is cleaned from short edge segments or those having an inappropriate aspect ratio and, therefore, do not contribute to the flow line, which is always oriented horizontally (Fig. 4d).

5.3. Image splitting

Constructing a continuous border line within a sewer based on partly horizontally overlapping edges is a complex problem that is simplified by splitting the sewer into its left and right parts. This corresponds to a cut through the center of the flow line in the side-view. Since pipe joints (couplings) have already been detected in an earlier step of the workflow (Fig. 1), where vertically oriented edges are extracted from the side-views in a similar way [20], each sewer is decomposed into its pipes. Furthermore, pipes are sub-divided into equally sized segments.

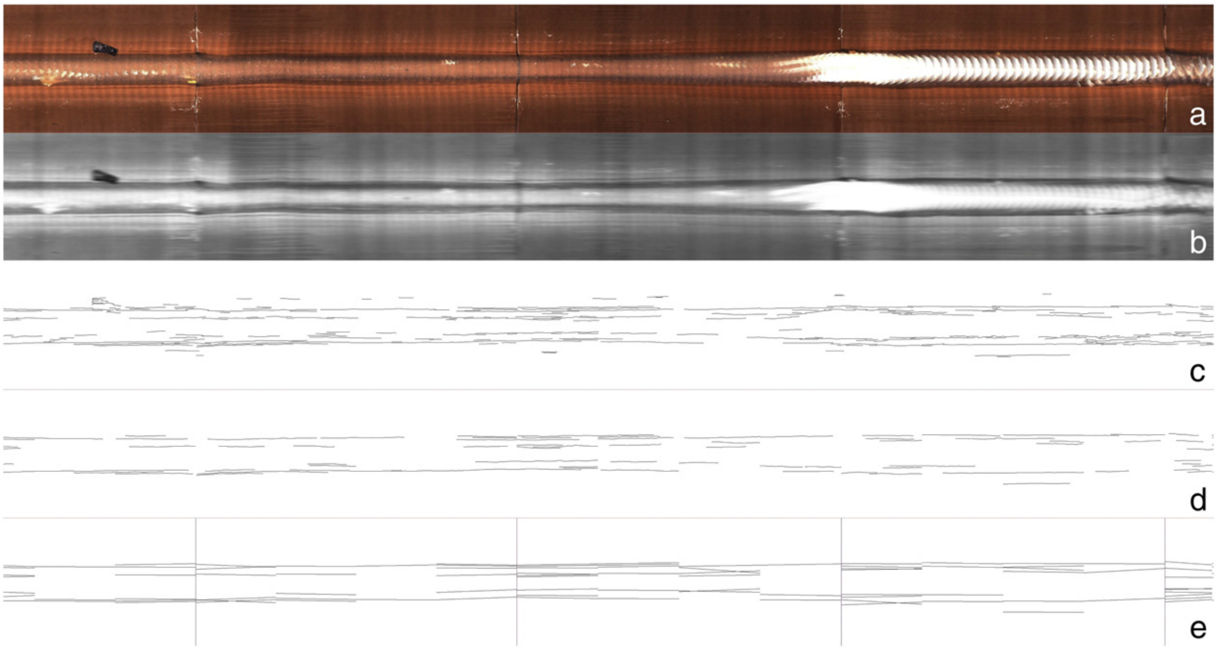

Fig. 4. Pre-processing steps. a) original, b) smoothed, c) dynamic edge detected, d) edges filtered, e) Hough transformed.

5.4. Hough transform

As a result of the edge detection and image splitting, several curved line segments are received in each segment, which are considered possible candidates of the flow line edge. To cope with gaps in the edge map, and to obtain straight line segments, a Hough transform is applied [6]. Within each segment, straight lines connecting the segment's left and right border are determined. All line candidates are described by their position and slope. For each candidate, the number of supporting edge pixels is stored and used to describe the candidate's reliability. Line candidates outside a reasonable range of slope or below a certain reliability are disregarded from further processing. All candidates now have the same length, which indicates the segment width. For instance in Fig. 4e, each pipe is modeled with four equally-sized segments.

5.5. Graph representation

At this stage, determining the flow line is modeled as finding an optimal path that is composed of appropriate line candidates from each of the segments. To solve this problem, we introduce a formal model. Fig. 5 shows the introduced functions in the context of the sewer image with three segments. For each segment, Canny edges (curved lines) and three lines candidates are shown. The relative reliability is derived from the number of supporting edge pixels. Gaps between line candidates are measured as vertical distance at the joint.

5.5.1. Mathematical notation

Let L_i^s denote the i-th line candidate in segment s. Then, the relative reliability $d(L_i^s)$ is given by

$$d(L_i^s) = \frac{H(L_i^s)}{W(s)} \tag{1}$$

where $H(L_i^s)$ represents the number of matched Hough pixels of line i in segment s, whereas $W(s)$ denotes the width of segment s in pixels. Hence, a perfect line candidate represented in an eight-neighborhood pixel connectivity model yields $d=1$.

The offset $O(L_i^s,\ L_j^{s+1})$ between line candidates from adjacent segments is derived from the gap in vertical direction. Let $Y_B(L)$ and $Y_E(L)$ denote the y-position of the first and the last pixel of a line candidate L, respectively. This yields

$$O\left(L_i^s, L_j^{s+1}\right) = |Y_E(L_i^s) - Y_B\left(L_j^{s+1}\right)|. \tag{2}$$

For further calculations, the offset $O(L_i^s, L_j^{s+1})$ is normalized by the maximal offset found in the entire sewer

$$O_{\max} = \max_{\substack{s=1..(S-1)\\ i=1..N(s)\\ j=1..N(s+1)}} \left(O\left(L_i^s, L_j^{s+1}\right)\right) \tag{3}$$

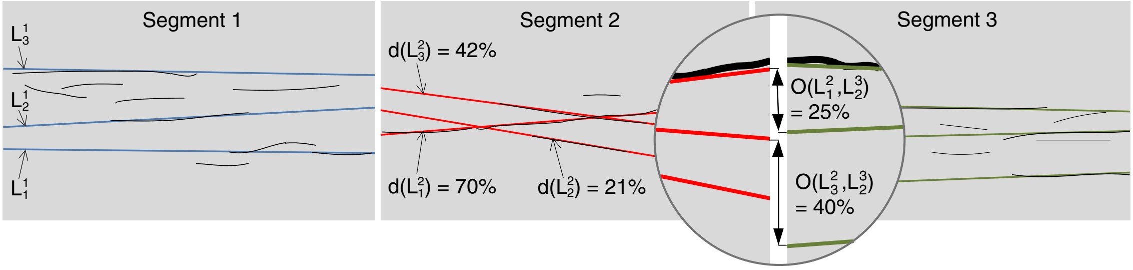

Fig. 5. Intensity and offset of line candidates.

where S represents the total number of segments and *N(s)* is the number of line candidates in segment s. The normalized offset $o(L_i^s, L_j^{s+1})$ is given by

$$o\left(L_i^s, L_j^{s+1}\right) = \frac{O\left(L_i^s, L_j^{s+1}\right)}{O_{\max}} \tag{4}$$

5.5.2. Optimization problem

To retrieve a piecewise linear but smooth representation of the actual flow line, we assign a cost function to a sequence of line candidates.

For each line segment and connection point, the costs incorporate the reliability from Hough transform $d(L_i^s)$ as well as the normalized offset $o(L_i^s, L_j^{s+1})$, respectively. To adapt the balance between the cost components, we introduce a weighting factor α. The costs between two line candidate L_i^s and L_j^{s+1} in two adjacent segments s and $s+1$ are now defined as

$$c\left(L_i^s, L_j^{s+1}\right) = \alpha \cdot o\left(L_i^s, L_j^{s+1}\right) + (1-\alpha) \cdot d\left(L_j^{s+1}\right) \tag{5}$$

For the sake of symmetry, we define a virtual segment $s=0$ with

$$N(0) := \mathrm{N}(1)$$
$$Y_E\left(L_i^0\right) := Y_B\left(L_i^1\right) \quad \forall i = 1 \ldots N(1)$$

resulting in $o(L_i^0, L_i^1)=0\,\forall i=1\ldots N(1)$. In other words, only the reliability d of the first segment is taken into account for the overall costs, as the offset is zero for all connecting line candidates. Using $c(L_i^s, L_j^{s+1})$, the best flow line, is given now by the minimization problem

$$\min_{(r_0,\ldots,r_S)\in R}\left(\sum_{s=0}^{S-1} c\left(L_{r_s}^s, L_{r_{s+1}}^{s+1}\right)\right) \tag{6}$$

where R is the set of all possible line combinations over all segments. This minimization problem, however, turns out to be ill-posed and brute-force methods are incapable of solving it due to the high number of line candidate combinations in a long sewer. Assuming ten candidates in each of the five segments per pipe and twenty pipes per sewer yields $100^{10}=10^{20}$ possible combinations.

5.5.3. Filling gaps

Although edge detection is processed dynamically, gaps between segments may occur if parts of the edge maps do not yield sufficient pixels. This can lead to segments which contain no line candidates after applying the Hough transform. As the minimization problem is only defined when the cost function c delivers a result for every transition between segments, it is necessary to fill these gaps. This can be achieved introducing virtual line candidates, each of which is connecting the endpoint of a line candidate in a segment *s-1* with a starting point in segment $s+1$. Such virtual candidates are placed for every end–start combination. The intensity of virtual lines is defined as $d\equiv 0$.

5.5.4. Problem transformation

To find a solution in reasonable time, the minimization problem is transformed into that of finding the shortest path through a weighted graph. Each line candidate $L_i^s\,\forall i=1..N(s)$ in all segments $s=1\ldots S$ is transformed into a node in the graph. Edges are defined between each node representing a line candidate L_i^s and all nodes representing line candidates in an adjacent segment $L_j^{s+1}\forall j=1..N(s+1)$. The costs assigned to the edges are given by the cost function $c(L_i^s, L_j^{s+1})$. To define the starting point of the graph, a node B is added and connected to each node $L_i^1\,\forall i=1..N(1)$ representing a line candidate in the first segment. The edges are weighted by the $(1-\alpha)\cdot d(L_i^1)$. As the endpoint of the graph, a node E is added, which is connected to all nodes $L_i^S\,\forall i=1..N(S)$. The edge costs are defined as $c=0$ (Fig. 6).

Mathematically, let

$$M(s) := \left\{L_1^s, \ldots, L_{n(s)}^s\right\} \tag{7}$$

denote the set of all possible line candidates in segment s. A node n in the graph can then be seen as

$$n \in B \cup \bigcup_{s=1\ldots S} M(s) \cup E \tag{8}$$

The set P of all possible paths from the beginning B to the end E can then be defined as

$$P := B \times M(1) \times M(2) \times \ldots \times M(S) \times E \tag{9}$$

so that one possible path p is given by

$$P \ni p = \left\{B, L_{i_1}^1, L_{i_2}^2 \ldots L_{i_S}^S, E\right\} \tag{10}$$

Hence, the cost for a path p yields

$$C(p) = (1-\alpha)\cdot d\left(L_{i_1}^1\right) + \sum_{s=1..S-1} c\left(L_{i_s}^s, L_{i_{s+1}}^{s+1}\right) \tag{11}$$

5.5.5. Handling of outliers

Outliers may occur in some of the segments. In such cases, only line candidates are returned from the Hough transform that are far away from the flow line marking scratches or abrasions. For instance in Fig. 2 d, e, n, prominent edges in horizontal orientation are seen at a distance from the flow line. If the optimization algorithm has to choose from such outliers only, the result turns suboptimal. To handle outliers, artificial lines are introduced similar to the virtual lines, as described already in Section 5.5.3. Adding all the direct connections allows the optimizer to select a smooth path. As the artificial line candidates are direct connections, their offset $o(L_i^s, L_j^{s+1})$ to the candidates in the next segment is zero. However, to make them less attractive, we use the offset $o(L_i^{s-1}, L_j^{s+1})$ instead when calculating the cost for these candidates.

5.6. Dijkstra algorithm

According to the problem definition in (6), the optimal flow line is now given by the path p in the graph from B to E with the lowest costs $\min_{p\in P}\{C(p)\}$. Lowest costs in this case is equivalent to the shortest path p from B to E. In 1959, Edsger W. Dijkstra proposed an optimal algorithm to find the shortest path between two nodes in a graph with non-negative edge costs [5], which is being used in this article.

6. Results

Line candidates are extracted by pre-processing, dynamic edge detection, and Hough transform. In (5), the costs function c combines the strength d with the offset o to the adjacent segment, and the weighting factor α controls the ratio between these two values. In Fig. 7 from top to bottom, the results of gradually increasing α are emphasized. Cyan colored lines refer to real lines detected by the Hough transform whereas red (darker) lines are virtually or artificially added. Each pipe has been divided into four segments. In Fig. 7a, $\alpha=0$, which means only the line intensity d is taken into account when choosing the line candidates. The resulting flow line tends to be very patchy as a close connection to adjacent lines is not taken care of. Increasing α levels the flow line. At $\alpha=0.8$, a smooth flow line is obtained resembling closely

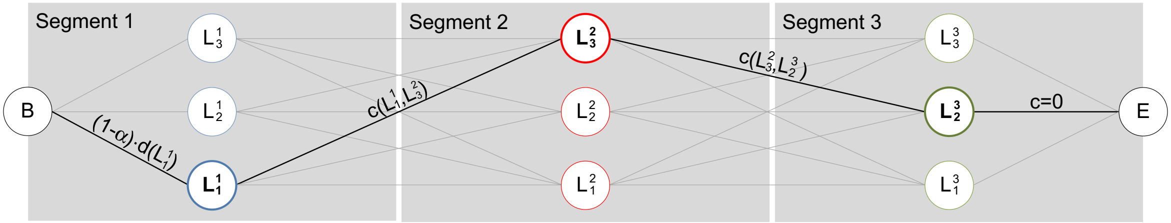

Fig. 6. Graph representation of minimization problem to find best line candidates.

the original line. Higher values tend to over flatten the results, putting too much stress on the artificially introduced lines.

To determine the optimal alpha systematically, ten sewers of different material have been arbitrarily selected from the database, and the actual flow line (ground truth) was annotated manually. To cope with inter-observer variabilities, ground truth labeling was performed by experienced engineers (KM, TD) as well as computer scientists (SK, MW) individually localizing as many supporting points on the flow line as assumed subjectively appropriate. In total, a sewer length of 474 m was annotated with its left and right flow lines. This reference data was compared to the flow lines computed automatically using $0.0 \le \alpha \le 1.0$, and the Dice coefficient [4] was determined (Fig. 9, left scale). Surprisingly, $\alpha = 0.8$ performs best for all users labeling the ground truth. This is due to the clear minimum of the standard deviation of the distance between ground truth and detected line (Fig. 9, right scale).

As already demonstrated in Fig. 2, flow lines appear differently. Fig. 8a, for example, shows a discontinuous flow line that was delineated precisely. Although insufficient edge information is present in several segments, our algorithm correctly approximates the missing data. In Fig. 8b, obstacles and reflections are shown: Again, the algorithm precisely is following the actual flow line. The sewer in Fig. 8c contains many residues resulting in long horizontal structures parallel to the actual flow line. Due to the continuous criteria, outlying structures are disregarded. In Fig. 8d, a sewer built of concrete is shown. Both, inner and outer parts from the flow line are represented with low intensity. Here, the inner flow line was selected as it more closely resembles straight and smooth lines. The algorithm is also robust against high variations in brightness within a sewer, e.g., as obtained from pipe to pipe (Fig. 8d). Again, the delineated flow line matches exactly.

7. Discussion

In this article, an approach to flow line detection using Hough transform and optimization by Dijkstra's shortest path algorithm has been presented. Although Dijkstra's algorithm was proposed over 50 years ago, it is still applied frequently in image processing and pattern recognition. Gunkel et al. [11], for example, recently have presented an application of crack detection in plastic pipes. Based on the paths closed with Dijkstra, the authors statistically analyzed the crack behavior. Yu et al. [26] use a similar technique for crack analysis in concrete tunnels. Interestingly, the field of application here is very close to ours. However, there are many other fields of application in computer vision [8]. For example, Rebelo et al. [22] optimize the detection of staff lines in scanned handwritten musical scores. The excellent results obtained in our work determining the best-fitting flow line in long sewers composed of several pipes with a wide range of materials and diameters is another example emphasizing the potential of Dijkstra's algorithm.

Beside the flow line detection algorithm, we presented a large collection of real world data to cope with the varieties in sewer wall appearance. In fact, the database is annotated completely with position information of joints, events and damages. For flow lines however, an objective definition of the ground truth is crucial. In fact, flow lines of

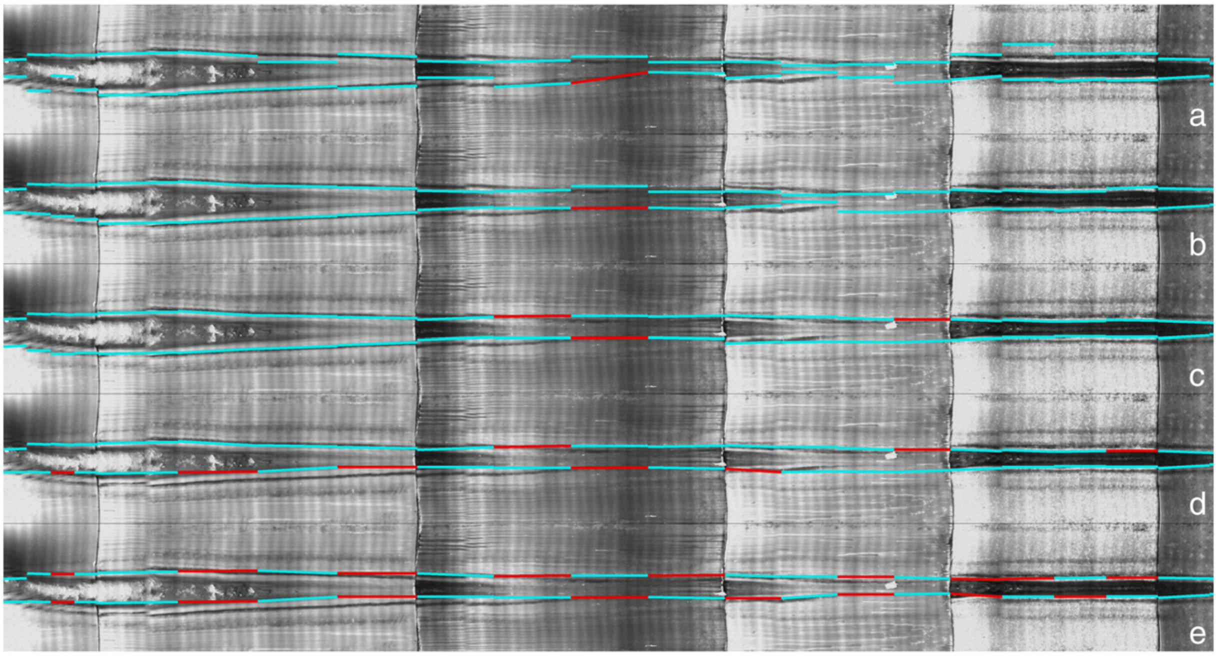

Fig. 7. Influence of smoothing factor α. a) $\alpha = 0.0$, b) $\alpha = 0.4$, c) $\alpha = 0.8$, d) $\alpha = 0.9$, e) $\alpha = 1.0$.

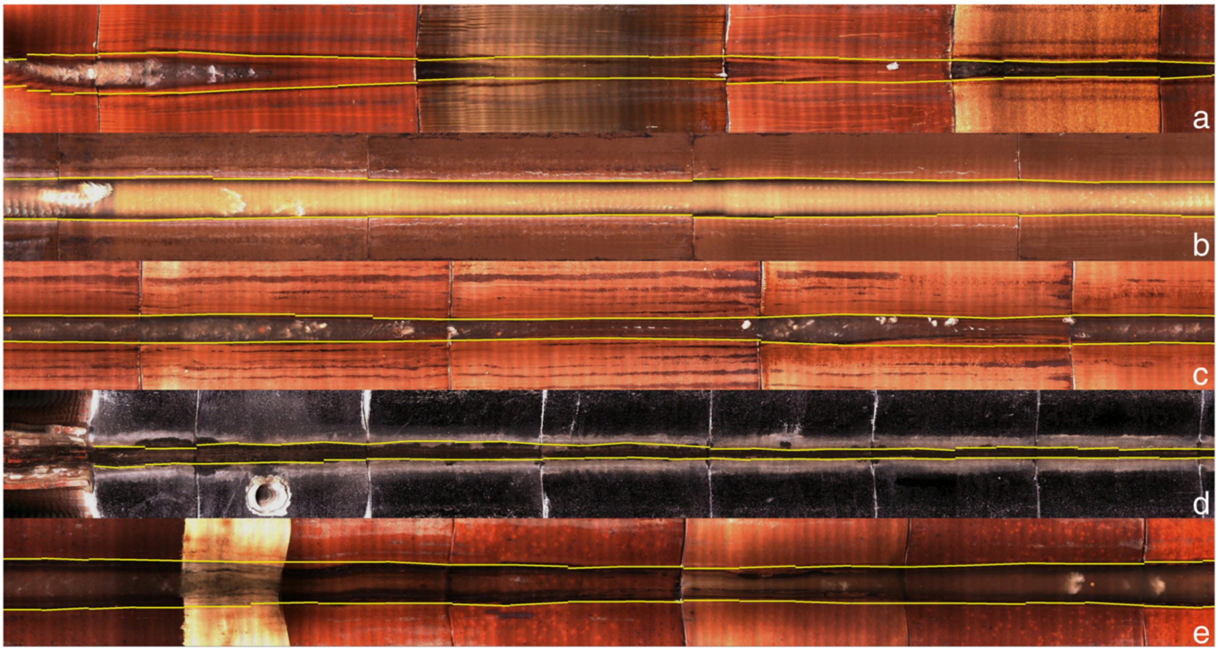

Fig. 8. Detected flow line on different materials, age and diameters. a) stoneware (1965) 200 mm, b) stoneware (1982) 250 mm, c) stoneware (1970) 200 mm, d) concrete (1978) 250 mm, e) stoneware (1976) 200 mm.

different flooding levels may be seen, they may be covered by scratches or other artifacts, or they may not be determined sharply. Hence, any manual ground truth labeling is observer dependent. Such an “inter-observer variability” is a well known effect in medical research [14]. In our study, it is emphasized by the different Dice curves plotted in Fig. 9. Unexpectedly, the optimal choice of α is not affected by the observer labeling the ground truth. This may be due to the fact that both, human observer and high α values tend to smooth the flow line. Anyway, we see it advantageous that alpha must neither be adapted to the type, size nor material of the sewer. This eases further image processing.

The importance of large reference databases was already acknowledged by Schindler and Stamm [23,27]. Here, the authors used sewer images of 10,000 m in length for developing an image-based detection of inlets. Such an inlet detection forms the next step in automation in sewer analysis. Having the joints and flow lines delineated, inlet detection can be restricted to the remaining areas, according to the general processing chain (Fig. 1). Then, it may be advantageous to differ regular from irregular sewer wall sections, and apply further algorithms for segmentation and classification to the irregular structures. However, there is still a long way to go before automatic CCTV-based sewer inspection can be applied in practice.

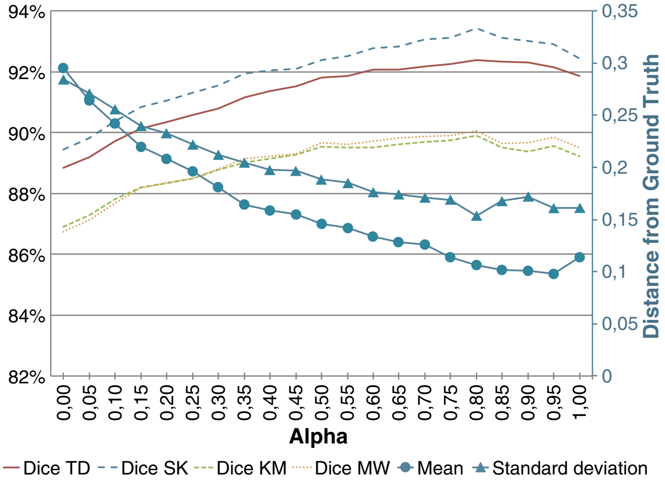

Fig. 9. Quality of detection as a function of α and the user determining the ground truth.

Nonetheless, flow line detection also supports manual inspection. Since separation of regular from irregular areas is based on a robust flow line detection, the inspector might now focus on those parts of a sewer, where irregularities have been detected but not yet classified automatically. Therefore, our robust algorithm already contributes to further automation in urban drainage construction and maintenance.

Acknowledgment

This research was partly funded by the Work Group of Industrial Research Associations "Otto von Guericke" (AiF), Germany, No. KF0110802LF8.

References

[1] C. Berger, J. Lohaus, Der Zustand der Kanalisation in der Bundesrepublik Deutschland. Ergebnisse der ATV-Umfrage 2004, http://www.dwa.de 2004.
[2] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986) 679–698.
[3] R. Deriche, Using Canny's criteria to derive a recursively implemented optimal edge detector, International Journal of Computer Vision 1 (1987) 167–187.
[4] L.R. Dice, Measures of the amount of ecologic association between species ecology, Journal of Ecology 26 (1945) 297–302.
[5] E.W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1959) 269–271.
[6] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM 15 (January 1972) 11–15.
[7] O. Duran, K. Althoefer, L.D. Seneviratne, Automated pipe defect detection and categorization using camera/laser-baser profiler and artificial neural network, IEEE Transactions on Automation Science and Engineering 4 (2007) 118–126.
[8] P.F. Felzenszwalb, R. Zabih, Dynamic programming and graph algorithms in computer vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 99 (2010) (PrePrints).
[9] R.A. Fenner, Approaches to sewer maintenance: a review, Urban Water 2 (4) (2000) 343–356.
[10] G. Gangl, T. Ertl, F. Kretschmer, H. Kainz, R. Haberl, Development of a guideline for sewer operation and maintenance in Austria, Proc. Sewer Operation and Maintenance SOM 06, Vienna, Austria, 2006.

[11] C. Gunkel, A. Stepper, A.C. Müller, C.H. Müller, Micro crack detection with Dijkstra's shortest path algorithm, Technical Report, Department of Statistics, Technical University Dortmund, Dortmund, 2009.

[12] P.V.C. Hough, Machine analysis of bubble chamber pictures, Proc. Int'l Conf. High Energy Accelerators and Instrumentation, 1959.

[13] S. Iyer, S. Sinha, A robust approach for automatic detection and segmentation of cracks in underground pipeline images, Image and Vision Computing 23 (10) (2005) 921–933.

[14] T.M. Lehmann, From plastic to gold: a unified classification scheme for reference standards in medical image processing, Procs SPIE 4684 (3) (2002) 1819–1827.

[15] J. Mashford, P. Davis, M. Rahilly, Pixel-based colour image segmentation using support vector machine for automatic pipe inspection, Lecture Notes Computer Science 4830 (2007) 739–743.

[16] J. Mashford, D. Marlow, S. Burn, An approach to pipe image interpretation based condition assessment for automatic pipe inspection, Advances in Civil Engineering 2009 (2009) 1–11.

[17] J. Mashford, M. Rahilly, P. Davis, S. Burn, A morphological approach to pipe image interpretation based on segmentation by support vector machine, Automation in Construction 19 (2010) 875–883.

[18] O. Moselhi, T. Shehab-Eldeen, Automated detection of surface defects in water and sewer pipes, Automation in Construction 8 (5) (1999) 581–588.

[19] O. Moselhi, T. Shehab-Eldeen, Classification of defects in sewer pipes using neural networks, Journal of Infrastructure Systems 6 (3) (2000) 97–104.

[20] K. Müller, B. Fischer, Objective condition assessment of sewer systems, Proc. LESAM 2007-2nd Leading Edge Conference on Strategic Asset Management, 2007, pp. 1–13.

[21] E.C. Ottenhoff, H. Korving, Verification tool for sewer database quality, Proc. Sewer Operation and Maintenance SOM 06, Vienna, Austria, 2006.

[22] A. Rebelo, A. Capela, J.F.P. da Costa, C. Guedes, E. Carrapatoso, J.S. Cardoso, A shortest path approach for staff line detection, Proc. Automated Production of Cross Media Content for Multi-Channel Distribution, 2007. AXMEDIS'07. Third International Conference on, Nov. 2007, pp. 79–85.

[23] M. Schindler, C. Stamm, Digitale Bildverarbeitung in der Röhreninspektion, IMVS Fokus Report, 2007, pp. 32–38.

[24] S. Sinha, P. Fieguth, Segmentation of buried concrete pipe images, Automation in Construction 14 (2006) 73–83.

[25] S. Sinha, F. Karray, Classification of underground pipe scanned images using feature extraction and neuro-fuzzy algorithm, IEEE Transactions on Neural Networks 2 (2002) 393–401.

[26] S. Yu, J. Jang, C. Han, Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel, Automation in Construction 16 (3) (2007) 255–261.

[27] Q. Zhong, M. Schindler, C. Stamm, Lateral detection, Proc. VMV 2008: Vision, Modeling and Visualization, Konstanz, Germany, 2008.