# Robot navigation and obstacle detection in pipelines using time-of-flight imagery

Jens T. Thielemann, Gøril M. Breivik and Asbjørn Berge

SINTEF, P.O.Box 124 Blindern, N-0314 Oslo, Norway

## ABSTRACT

Range imagery provided by time-of-flight (TOF) cameras has been shown to be useful to facilitate robot navigation in several applications. Visual navigation for autonomous pipeline inspection robots is a special case of such a task, where the cramped operating environment influences the range measurements in a detrimental way. Inherent in the imaging system are also several defects that will lead to a smearing of range measurements. This paper sketches an approach for using TOF cameras as a visual navigation aid in pipelines, and addresses the challenges concerning the inherent defects in the imaging system and the impact of the operating environment. New results on our previously proposed strategy for detecting and tracking possible landmarks and obstacles in pipelines are presented. We consider an explicit model for correcting lens distortions, and use this to explain why the cylindrical pipe is perceived as a cone. A simplified model, which implicitly handles the combined effects of the environment and the camera on the measured ranges by adjusting for the conical shape, is used to map the robot's environment into an along-axis-view relative to the pipe, which facilitates obstacle traversal. Experiments using a model pipeline and a prototype camera rig are presented.

## 1. INTRODUCTION

3D imaging of surroundings is useful for aiding robot navigation, and a lot of research effort has been applied to propose good solutions to this challenge. Mainly, research has been focused around utilizing range images from laser scanners or stereo cameras. The relatively novel technology of time-of-flight (TOF) cameras has also proven useful for robot navigation in several cases, both as a navigational aid and for use in obstacle detection and avoidance.[1–4] Kolb et al[5,6] give an overview on state-of-the-art for both the technology and its applications for TOF cameras.

Autonomy in pipe inspection is still in its infancy, and most current pipeline inspection systems use remotely operated cable-tethered robots.[7] A robust and reliable 3D imaging system is necessary to facilitate navigation for autonomous operation. Ahrary et al[7] uses stereo camera and laser scanner data, measuring distances by stereo matching. Stereo cameras are also used in the MAKRO project.[8] The autonomous robot KURT[9] does landmark detection based on ultrasound pings in order to recognize different types of pipe crossings.

In this paper we discuss the application of a TOF camera for navigation inside a pipe. The visual system is developed for use in a pipe inspection robot consisting of a train of independent modules, allowing the robot to perform snakelike maneuvers. The final robot will be able to move vertically and horizontally through pipelines of different diameters, and route around obstacles. Pipelines are a constrained environment with few available landmarks, mainly junctions and bends. This allows for a greatly simplified strategy for navigation, compared to the custom methods of simultaneous localization and mapping (SLAM),[10,11] since a coarse predefined map of pipeline landmarks coupled with a landmark detection method is sufficient for robot localization and navigation.

The novelty of applying a TOF camera in this context is mainly that TOF cameras provide dense range measurements at video frame rates. Both stereo cameras and laser scanners provide sparse sets of range measurements. Laser scanners operate by measuring distances in a coarse grid across the sensor field-of-view. Stereo cameras introduce physical restrictions on the robot due to the need for camera separation, and sparsity in data

Further author information: (Send correspondence to J.T.T)
J.T.T.: E-mail: jens.t.thielemann@sintef.no, Telephone: +47 22 06 74 77
G.M.B.: E-mail: goril.m.breivik@sintef.no, Telephone: +47 22 06 75 83
A.B: E-mail: asbjorn.berge@sintef.no, Telephone: +47 22 06 76 94

is introduced since the distance measurements are based on feature matching. Ultrasonic measurements has fairly low spatial resolution.

In our paper[12] from 2008 we presented the first results on the application of a TOF camera in a pipeline environment. Although the results were promising, it was evident that imaging with TOF in a cramped environment such as a pipeline introduces several detrimental effects. Inherent in the imaging system are several defects that lead to a smearing of range measurements, and secondary reflections of the active illumination affect the application. Explicit approaches for correcting lens smearing in the range data in conventional scenes has recently been proposed.[13,14]

The contribution of this paper consists of three parts. First, we investigate application of an error-correction approach for correcting lens smearing in our pipeline data. These results may explain why the cylindrical pipe is perceived as a cone, as concluded in our previous paper.[12] The cone, however, represents a simplified model for implicitly handling the combined effects of the environment and the camera on the measured ranges, and we choose to use this model for further investigation, instead of continuously correcting each image using the error-correction model.

The second contribution of this paper is further results on landmark detection and tracking. This is done, as before, by finding significant deviations from the idealized, cylindrical shape of the pipeline. Classification of the landmarks is slightly enhanced, taking advantage of the 3D data provided by the TOF camera, and using scale-invariant features, such as area in 3D.

Our third contribution is on obstacle detection and path planning. We propose a procedure for path planning for a robot to traverse obstructions inside a pipeline. In the procedure, the obstacle position is found relative to the pipeline, called along-axis-view, instead of relative to the robot, since this gives a more complete mapping of the surroundings. In the presented results we track obstacles in the along-axis-view, and show that path planning is not possible when the obstacle is closer than 0.6-0.7 meters, due to the narrow field-of-view of the camera.

This paper is structured as follows. A discussion of the available research on error correction of the camera defects is given in Section 2. Our approach for scene segmentation and region analysis is presented and discussed in Section 3. Section 4 gives a brief overview of the setup used in the experiments, while Section 5 presents and discusses the results of experiments performed on real datasets. Section 6 concludes the paper and gives a brief outlook on the obtained results.

## 2. CAMERA ERROR CORRECTION

TOF cameras are plagued by range measurement smearing, where less-bright pixels, in the presence of a bright object close to the camera, will appear closer to the camera than they actually are, and this is discussed in several recent papers. Inverse models estimating corrections for these errors were presented both by Kavli et al[13] and Mure-Dubois et al,[14] and we investigate these models in order to possibly explain the non-cylindrical appearance of the imaged pipelines.

In general, the smear correction methods model each pixel as a complex number with absolute value and phase representing amplitude and distance measurement. Furthermore, the signal $c_m = a_m e^{jp_m}$ measured in each pixel can be modeled as

$$c_m = c_d + c_{psf} = a_d e^{jp_d} + a_{psf} e^{jp_{psf}} \tag{1}$$

where $c_d$ is the actual signal received from the scene for this pixel, while $c_{psf}$ is the signal due to the point spread function of the camera. $a_d, a_{psf}$ and $p_d, p_{psf}$ represent the respective amplitude and phase of these signals. As demonstrated by both papers, in high contrast scenes $a_d$ and $a_{psf}$ may be of comparable size. If $p_d$ and $p_{psf}$ then differ significantly, the measurement $c_m$, and in particular $p_m$, may be significantly distorted. This may give very unreliable distance estimates for the darker parts of the obtained images.

Range measurement smearing is a substantial problem also in pipelines, and this effect has to be identified and compensated for when processing the data. Two approaches to compensate for this effect are discussed in Section 5.1. The first approach implicitly compensates for the effects by modeling the pipe as a cone instead of a cylinder, while the second approach explicitly implements the error correction method described by Kavli et al.[13]

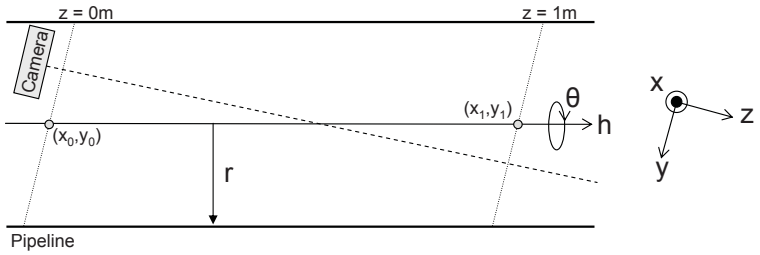



Figure 1: Illustration of the camera coordinate system $(x, y, z)$ and pipeline coordinate system $(r, h, \theta)$. The pipe is parameterized as a cylinder using the two points $(x_0, y_0)$ and $(x_1, y_1)$ at distance $z = 0$ and $z = 1$ meter from the camera focal plane, and radius $r$ of the pipeline. For the pipe coordinate system, origo is defined as the intersection of cylinder axis and the camera's focal plane, and $\theta = 0, \pi/2, \pi$ indicate respectively the left, up, and right hand side of the pipeline.

Kavli et al model the camera's smearing error as a convolution with a space-variant convolution filter. This means the observed ranges can be modelled as

$$z(r, c) = Z(r, c) * h(r, c)$$

where $Z(r, c)$ is the true range measurement, $z(r, c)$ is the observed range measurement and $h(r, c)$ is the space-variant filter. $r$ and $c$ are pixel indices. $h(r, c)$ varies strongly with image position, and has response in most of the image. In order to obtain an estimate $Z'(r, c)$ of the true $Z(r, c)$, inverse modelling is required. Kavli et al accomplish this by iteratively applying a forward model. This implies picking out the brightest pixels in the image, applying the convolution filter to these pixels, and subtracting the convoluted result from the original image. This has been shown to reduce the smearing effect by up to 60% in real images.

## 3. SCENE SEGMENTATION AND REGION ANALYSIS

Pipelines, usually being roughly cylindrical in shape, allow for a very simple initial scene segmentation. By comparing the measured 3D data points with a least squares fit of a cylinder, any sufficient deviations will be candidates for landmarks or obstacles in the pipeline. In the ideal case, 3D data points will appear cylindrical, however, as will be discussed in section 5.1, for several reasons, the data will take a conical shape. As the goal of fitting a cylinder to the data is to segment out potential landmarks and obstructions, and not to estimate the true pipeline parameters, we opt to segment out candidate regions by fitting a cone instead.

### 3.1 Fitting cylindrical shapes to 3D data

For most realistic settings, the view-direction of the camera will be fairly parallel to the axis of the pipeline. Therefore, we parameterize the cylinder to be fitted by the point where the cylinder axis intersects the camera focal plane, and the point where it intersects a second plane parallel to the focal plane one meter ahead of the camera's focal plane. The two intersection points are specified in cartesian coordinates, and with pipe radius this becomes five parameters. See Figure 1 for details. This parameterization is beneficial for later optimization because it has few continuous parameters, and they are of the same scale.

The cost function used for fitting the data to the cylinder model is inspired by the M-SAC[15] criterion. This criterion rewards good coherence between model and data, while simultaneously ignoring gross outliers coming from e.g. measurements of a wall in the end of a pipeline. The criterion is given by

$$R = \sum_i \rho(d_i) \qquad \rho(e) = \begin{cases} e^2 & e^2 < t^2 \\ t^2 & e^2 > t^2 \end{cases} \tag{2}$$

where $R$ is our cost function, $d_i$ is the distance between data point $i$ and the closest point on the cylinder candidate, and $t$ is an acceptable deviation threshold for gross outliers. Initial experiments indicate that measurement points that are either far away from the camera or have low signal amplitude are probably incorrect.

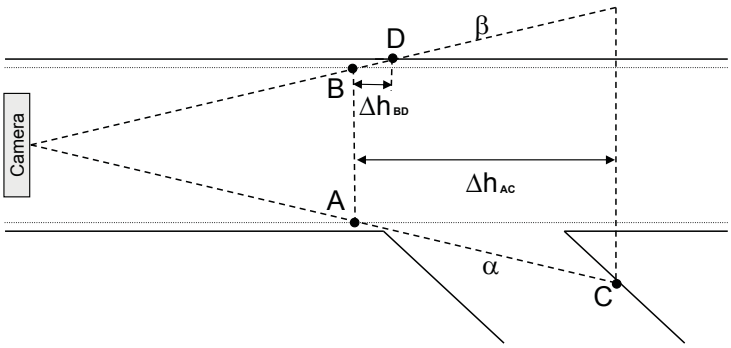



Figure 2: Illustration of blob detection process. Thick lines: Actual data. Thin lines: Fitted cylinder/cone. $\alpha$, $\beta$: Viewlines/pixels. $A, B$: Expected intersection points between viewlines $\alpha$, $\beta$ and fitted cylinder/cone. $C, D$: Actual data points along viewlines. Viewline $\alpha$ is detected as a possible landmark candidate due to the large distance $\Delta h_{AC}$ between $A$ and $C$. Viewline $\beta$ is classified as not being a landmark candidate due to the short distance $\Delta h_{BD}$ between $B$ and $D$.

When evaluating the cost function, such points are ignored by using fixed amplitude and distance thresholds to suppress them.

Optimization is done using the simplex method.[16] The optimization is initialized for the first image by using a cylinder axis coinciding with the camera axis. For subsequent frames, the result from the fit in the previous frame is used for initializing the optimization process. As the robot moves fairly little between frames, this reuse of parameters ensures faster convergence and that a more stable solution is found.

Initial experiments show that the acquired data within the range of interest appear to take a more conic than cylindrical shape, due to the point spread function of the camera, as discribed in section 5.1. We account for this by extending the model to fit a conic shape to the data, using the same principles as the above described cylinder fit. In this model, an additional parameter specifying cone steepness also needs to be estimated.

### 3.2 Extraction of candidate regions

Candidate regions for landmarks are detected by finding deviations from the fitted model. The recorded data is compared with the fitted model using the $h$-coordinate, and deviations are thresholded. See Figure 2 for illustration and explanation of the deviation measurements. Using the $h$-coordinate was in initial experiments found to give better results than e.g using the $r$-coordinate, since deviations in $h$-direction represents greater values than in the $r$-direction for the setup at hand.

Pixels resulting from the initial thresholding of deviations are grouped into blobs of connected pixels, and blobs with less than 20 pixels are excluded from further analysis. Classification is done based on a combination of blob characteristics in 3D, such are area and gradients within the blob and on the blob boundary.

Using gradients as features is inspired by Yanowitz and Bruckstein.[17] We have extended the idea for application in 3D and look for two types of gradients: internal gradients within the blob and adjacent gradients on the blob boundary. Blobs originating from true landmarks will lie in the neighborhood of a steep gradient due to openings in the pipeline. This is called an adjacent gradient and is illustrated in Figure 3.

If the cone fit is poor, very large blobs can result. Even if such blobs often also contain a true landmark, they should be rejected. The surface of the the landmark itself is relatively smooth and does not contain any steep internal gradients. Hence, the presence of a large internal gradient within the blob can ensure rejection of such over dimensioned blobs.

The ratio adjacent/internal gradient is a scale invariant measure and allows for looser thresholds on the scale variant features. Finally, area in 3D is used as a distance invariant measure, constraining landmarks to have a minimum size, regardless of the distance they are detected from.

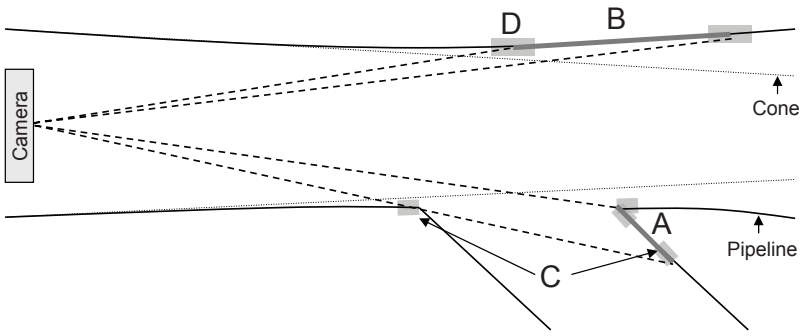



Figure 3: Illustration of cone fitting to a smeared pipeline shape, and illustration of 3D features for true/false landmark discrimination. Thick lines: Actual data. Thin lines: Fitted cone. Dashed lines: Viewlines/pixels. $A, B$: Segmented blobs, i.e. pixels used for 3D area and internal gradient computations. $C, D$: Pixels used for adjacent gradient computations. The true landmark $A$ contains an adjacent gradient on the boundary, indicated by pixels $C$. The false landmark $B$ is disregarded as a landmark due to no adjacent gradient in the boundary neighborhood $D$.

### 3.3 Tracking of landmark and robot position

After scene segmentation, the algorithm needs to report the distance and position of the landmarks for the results to become useful for a higher level navigation system. This is done by picking the blob pixel that has the lowest $h$-coordinate to determine the landmark range and position. Using this pixel for landmark localization is considered less ambiguous than possible alternatives. For instance, instead using the mean distance to the detected object will shift the estimated landmark position as the landmark gradually comes more into view, making measurements less precise.

Landmarks are tracked over time by selecting the closest neighbor in $(h, \theta)$-space for each detection point found in subsequent images. To remove spurious landmarks, the landmark association is not done if the difference in distance and angle between two consecutive landmarks exceeds predefined limits, tuned experimentally to 10 cm and 23 degrees. The $r$-component is not taken into account, as this parameter does not provide any additional information on changes over time in a pipeline of constant diameter. Landmarks appearing in less than five consecutive images are considered outliers, but landmarks are allowed to disappear from single images.

The robot position is estimated by forwarding the estimated location of the detected landmarks to a particle filter,[18] which compares the detected landmarks with a predefined map of the pipeline, indicating expected position of each landmark.

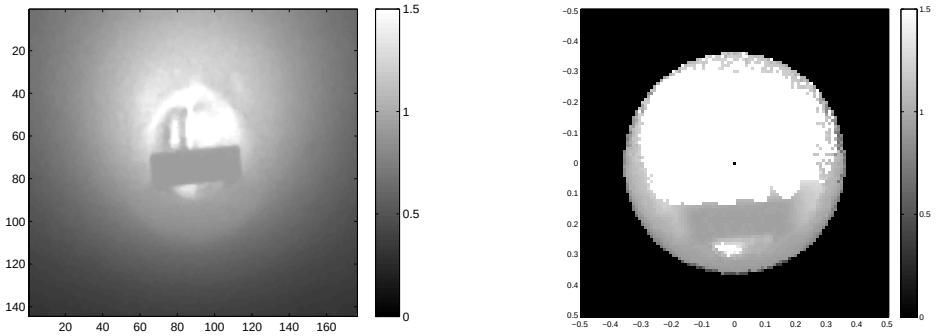
### 3.4 Obstacle detection and avoidance

The robot's ability to climb and pass pipe obstacles requires visual recognition and path planning. The pipeline is a constrained and well-defined environment, and it is convenient to localize the obstacle relative to the pipe instead of relative to the camera.

We do this by mapping the data from the camera coordinate system into an image $h_o(x, y)$, which represents a view of the pipeline along the pipe axis, with $(x, y)$ indicating horizontal and vertical axes across the pipeline, and gray scale value of each pixel indicating the estimated distance to obstacle.

Since a cone model is used for obstacle detection/segmentation, the mapping can not be done directly, as the estimated cone radius $\rho(h)$ at a particular distance $h$ into the pipeline is a function of $h$. The along-axis view is calculated as the distance $r'$ each pixel has from the pipeline's axis, mapped onto a plane at camera position in $h = 0$. $r'$ is computed as $r' = \rho(0)r/\rho(h)$, where $r$ is the estimated radius for each pixel at distance $h$ from the camera, $\rho(0)$ is the estimated cone radius at distance 0 and $\rho(h)$ is the estimated cone radius at distance $h$ from the camera. Before mapping into the image $h_o$, $r'$ and $\theta$ are converted into $(x, y)$ coordinates.

Practice show that very dark pixels in the TOF camera gives erroneous ranges, thus pixels having an amplitude of less than 500 are ignored. This amplitude threshold is set relatively low to have a low possibility of missing potential obstacle pixels. Such pixels are indicated by infinite along-axis distance in the $h_o(x, y)$ image.



(a) (b)

Figure 4: Example of obstacle mapping for detection. (a) Source image used for obstacle detection. We see an obstacle approximately 1.5 meters in the pipeline. The $(x, y)$-axis indicate pixel position and gray scale indicate $z$-data in meters from camera. (b) Along-axis view $h_o(x, y)$ of pipeline after the same image observations have been mapped into a plane where origo is center of cylinder, and $(x, y)$-axis indicate estimated horizontal and vertical distance from cylinder axis in meters. The gray scale of each pixel in $h_o(x, y)$ indicate the distance to the closest obstacle in the pipeline. White regions indicate areas in source image where intensity is too low for obstacle detection.

An example image of the mapping into image $h_o(x, y)$ is shown in Figure 4b using the source data displayed in Figure 4a. This mapping is beneficial for providing the robot with distance and position of future obstacles in a well defined reference frame relative to the pipe instead of relative to the camera.

To enable path planning for the robot, the mapped image $h_o(x, y)$ is converted to an occupancy grid $O_g(x, y, h)$.[19] This representation shows free and filled space by using a binary value in each voxel, where $h$ indicates distance along cylinder axis and $(x, y)$ are horizontal and vertical axes across the pipeline. A voxel in the occupancy grid is $1 \times 1 \times 10$ cm, with the coarsest resolution along the cylinder axis. The occupancy grid is created through thresholding the $h_o$ image, letting $O_g(x, y, h) = 1$ if $h_o(x, y) < h$, where $h$ indicates the current slice in the occupancy grid.

Each slice along the cylinder axis in the occupancy grid is a binary image indicating where there is free space, and we apply an morphologic image erosion operator with a mask with same size as the robot to search for places where the robot will fit, yielding a new occupancy grid $O_{\text{fit}}$. Preprocessing of the $h_o(x, y)$ image, using a $3 \times 3$ median filter to remove noise, reduces the risk of false detection of impassable pipelines.

The robot needs sufficient ground support, and analysis of the occupancy grid with morphological operations serves to detect regions where the transition between free and non-free regions are mostly flat. This is done by performing two sets of erosions using flat masks with the same width as the robot. The first searches for flat free zones, the second for flat occupied zones. The intersection of all these results gives possible robot positions $O_{\text{supp}}$ for each slice along the pipeline.

A cost graph between all voxels that correspond to valid robot positions is set up to plan the path up to, and possibly beyond, a detected obstacle. The cost is set to the distance between the two voxels if the movement is feasible for the robot, after taking into account restrictions in the robot's turn radius. Final path planning is done using Dijkstra's shortest path algorithm.[20] Figure 5 shows an example of a result after path planning around an obstacle.

## 4. EXPERIMENTAL SETUP

In our experiments, we used a camera of type Mesa SwissRanger SR3000. This was mounted on a mobile platform capable of moving the camera through the pipeline system with constant speed. A pipeline model consisting of 40 cm diameter sewer pipelines was used for the experiments. Figure 6 shows a sketch of the pipeline system, that includes 45° and 90° bends plus Y- and T-junctions.

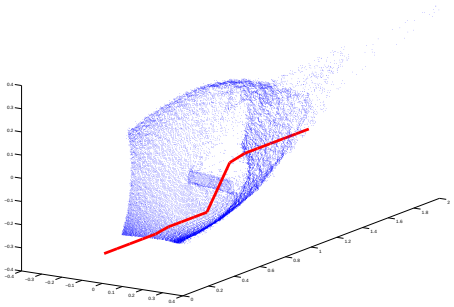


Figure 5: Illustration of a planned path based on obstacle detection. Axes are in meters. Points indicate measurements from camera, where a small obstruction clearly can be discerned. Thick line indicates planned path.

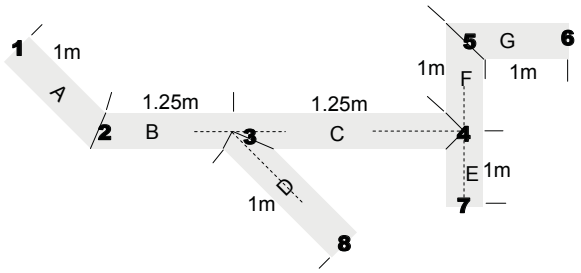



Figure 6: Sketch of the pipeline system used for experiments. Waypoint 2 is a 45° bend, waypoint 3 is a Y-junction, waypoint 4 is a T-junction and waypoint 5 is a 90° bend. The pipeline diameter is 40 cm.

# 5. RESULTS AND DISCUSSION

## 5.1 Range data modeling and camera error correction

Experiments were performed to decide the best model for description of the range data. The initial assumption was that recorded data would correspond to a cylinder. By visual inspection of recorded range data we observed that the images had some defects giving a field taking a conic shape. Further experiments also show that this shape defect is not constant, but depends on the contents of the pipeline.

### 5.1.1 Visual inspection of a pipeline without landmarks

Range images were recorded in a pipeline without any junctions or obstacles. Figure 7 shows a plot of the captured range image, seen from above. Recording defects are evident, since one would expect the point cloud to be cylindrical in shape. The observed defects are expected to be due to secondary reflections from the pipeline, lens scattering, poor camera calibration and other camera imperfections.

Inspection of the range data reveals that the data initially narrows slightly with increasing range. Note also that the density of range measurements decreases with increasing distance from the camera, limiting the useful distance range. From the camera and up to a range of 1.25 meters a conic shape can be clearly discerned.

### 5.1.2 Cylinder model versus cone model

Both the cylinder model and the cone model were fit to data captured of a pipeline with a single junction at a one meter distance from the camera. For this experiment, data points with an intensity below 3000 and a range beyond 1.5 meters were ignored. The intensity threshold was set relatively high due to a wish of including reliable data points only. The threshold $t^2$ for outlier data in the optimization criterion in Equation 2 was set to ten cm.

Illustrated in Figure 8a is the best cylinder fit for the measured data points. The cylinder fit points slightly off axis. Visual inspection of Figure 8b indicates that the cone model is a better fit for the region of interest

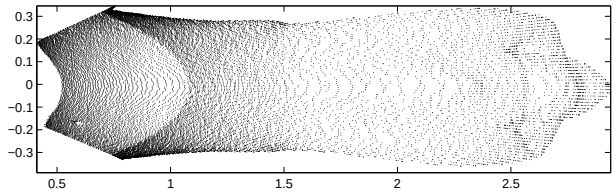
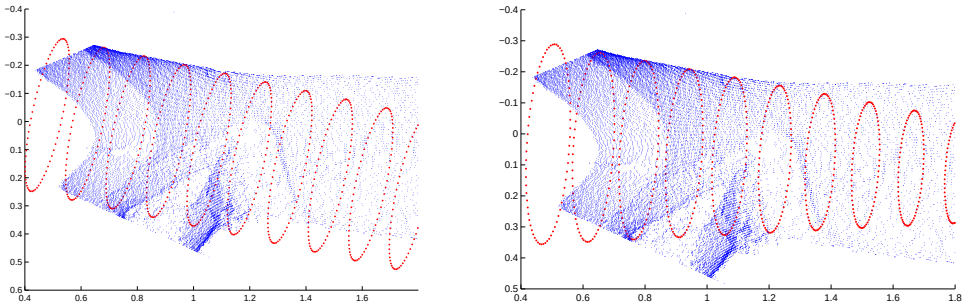


Figure 7: TOF camera range image of a pipeline without junctions or obstacles, seen from above. Axes in meters. The density of range measurements decreases markedly at a distance of 1.5 meters. Note the conic shape of the range data, discernible from the camera and up to 1.25 meters. Note also that the measurements indicate a pipeline diameter of 60 cm, while the pipeline diameter used in the experiment was 40 cm.



(a) (b)

Figure 8: Cylinder (a) and cone (b) fit to the TOF range image of pipeline with at junction at one meter distance, top-view. Camera positioned at B in Figure 6, looking towards C. Opening towards pipe D visible at approximately 0.75-1.25 meter distance. All units in meters. Fitted model indicated with solid points. Only data up to one meter is used for landmark and obstacle detection. Unevenly distributed data measurement points result in a best fitting cylinder slightly off-axis. The cone model handles this better and estimates the pipeline axis more accurately.

up to one meter, and in particular it estimates the axis better. We believe this is due to the conic shape of the actual measurements combined with lack of supporting points on one side of the pipeline.

#### 5.1.3 Camera's point spread function

Observations show that the conical shape of the surrounding pipeline changes as the robot approaches a bright obstacle. As the robot gets closer to the object, the estimated distance from the robot to the cone's convergence point gets closer as well. This is illustrated with a solid line in Figure 9.

As this behavior is likely due to smearing of range measurements inherent in TOF cameras, as discussed in Section 2, we attempted correcting for this using the method of Kavli et al.[13] Results after this correction is illustrated with the dashed line in Figure 9.

The results in Figure 9 clearly shows that after correction, range data appears cylindric, correcting the geometric distortion and conic shape. Thus, it seems evident that the conic shape of the recorded data may be due to smearing of range measurements.

For scenes where there is a clear foreground object, the correction appears to work successfully. However, for other pipeline scenes with no prominent foreground object, we found that this correction method introduced discontinuities in the corrected range data. These discontinuities made further data analysis impractical, since they hindered good estimation of the model parameters. Hence, for further analysis we chose to use the cone model to implicitly correct for the camera effects, rather than explicitly correcting for them through applying the described estimated inverse point spread function.

### 5.2 Landmark detection

The proposed landmark detection procedure was applied to data recorded when the robot traversed paths in the pipeline system, see Figure 6. The first results on this procedure was published in our previous paper[12] in 2008.

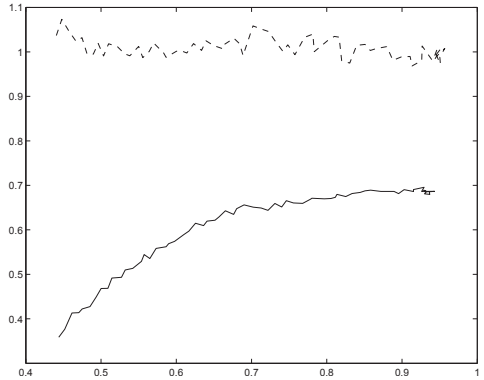



Figure 9: Shape of estimated cone as function of distance to small bright object in pipeline. $x$-axis: Distance in meters to obstacle as estimated by camera. $y$-axis: Ratio of cone radius one meter from camera divided by cone radius at camera position. Solid line/dashed line: Before/after applying smearing correction.[13] For uncorrected data the cone's shape becomes steeper as the obstacle gets closer. After correction, the pipeline is correctly estimated as a cylinder.

In this paper, a new test dataset was recorded independently of the dataset used to develop the algorithm, and no changes to the algorithm or parameters tuned to the training data were done.

As we did not have access to precise external odometry, we used a stopwatch and a tape measure to provide auxiliary speed measurements. Each junction or bend was approached twice. Separation of left-hand landmarks and right-hand landmarks was done by using the angular information associated with each landmark, using $[0.1, \pi/2]$ and $[\pi/2, \pi]$ as respective intervals.

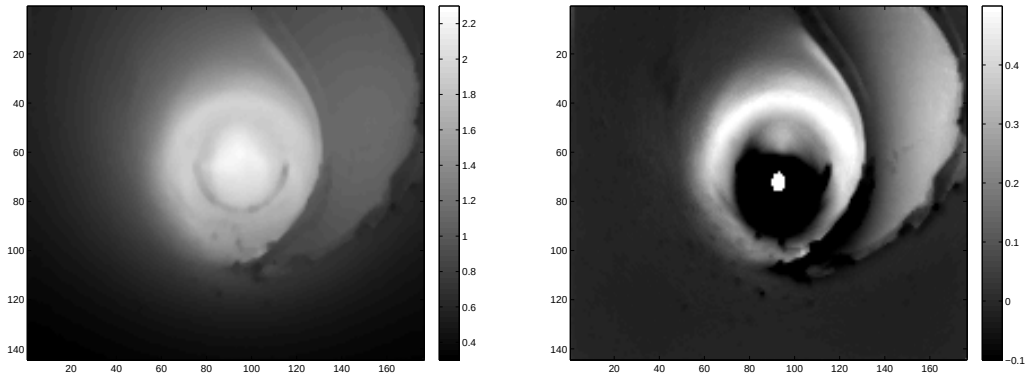
The cone model was fit to the range data in each frame in the video sequence. If the estimated cone converged to a point closer than two meters ahead of the camera, a fitted cylinder model was used instead. Each image was subsampled by a factor ten to speed up processing. As illustrated in Figure 2, deviations were measured as the projected distance of the observed range onto the fitted cone axis. Deviations exceeding three cm from the cone fit were classified as landmark candidates, and segmented out. The blob was classified as a landmark if the blob had a 3D area larger than 100 $\text{cm}^2$, an adjacent edge larger than ten cm, no internal edges exceeding ten cm, and adjacent blob edges two times larger than internal edges.

A range image showing the Y-junction can be seen in Figure 10a. In each pixel measured distance, i.e. the $h$-coordinate, is indicated in meters. The junction is clearly distinguishable in the right part of the image. By subtracting the cone fit from the observed data, a deviation measure for each pixel can be obtained. This is shown in Figure 10b. For pixels where we observe the pipeline walls, deviations would be expected to be close to zero.

Figure 11a illustrates the range image of the T-junction as seen from a distance of approximately 0.7 meters. The crossing pipe wall can be seen as a large area of having roughly a distance of one meter to the camera. Comparison of the cone model to observed data gives the deviation measure image illustrated in Figure 11b. The left and right turns of the pipe are easily discernible, and can be detected and tracked in the same way as the Y-junction. Note the occurrence of negative deviation in the center of the image since the cone model penetrates the crossing pipe wall.

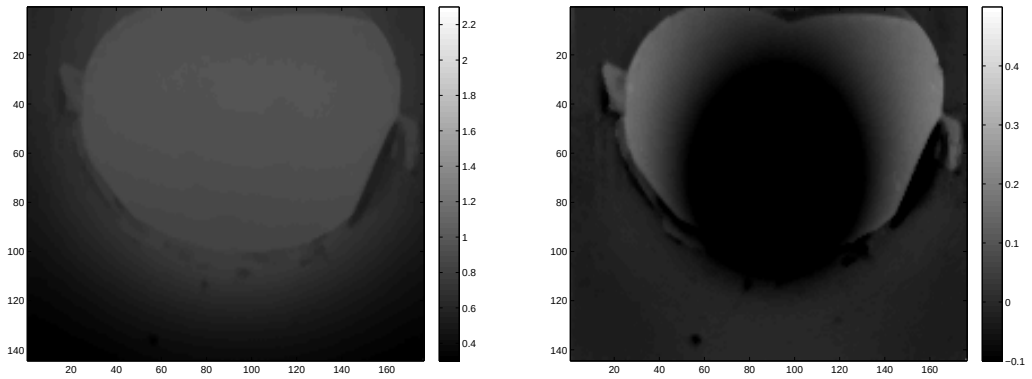
Table 1 summarizes our results of tracking landmarks over time on the acquired test dataset, compared to measurements from odometry. Odometry is found using a tape measure and stop watch. Speed results from camera is computed as the slope of the curve obtained by linear regression of the distance over time measurements. Deviation is calculated as the absolute deviation between camera and odometry measurements, relative to odometry. We overestimate the speed of approaching landmarks by 30% in average, and have a 16% standard deviation in our speed estimates. There is little variation in the manually estimated robot speed, indicating that these measurements were quite precise.

Figure 12 shows some examples of tracking performance. We see that the rate of approach is close to linear. However, the landmark is not detected and disappears from some frames. Manual inspection of the recording indicates this to be due to the blob classification process, in which the used thresholds appear to be too strict.



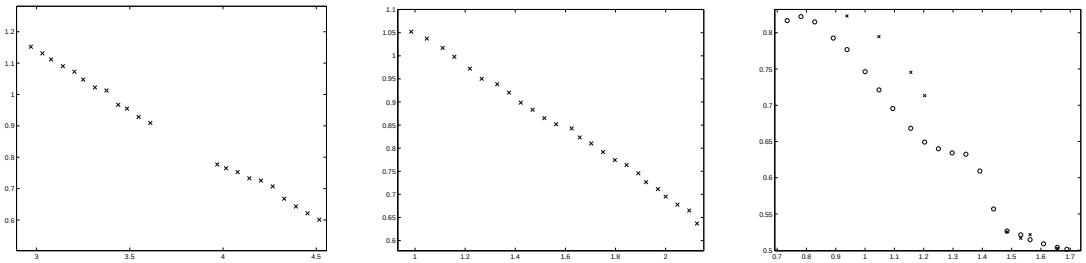
(a) (b)

Figure 10: Distance and deviation measure images of the Y-junction. (a) Raw TOF range image of Y-junction, indicating measured distance, i.e. the $h$-coordinate, in each pixel in meters. Observe that the landmark is clearly discernible in the right of the image at approximately one meter distance. (b) Deviation measure image of Y-junction obtained by subtracting expected range measurements of a synthesized cone from the raw TOF range image measurements. The Y-junction in the right of the image clearly deviates from the fitted cone.



(a) (b)

Figure 11: Distance and deviation measure images of the T-junction. (a) Raw TOF range image of T-junction, from a distance of roughly 0.7 meters, indicating measured distance, i.e. the $h$-coordinate, in each pixel in meters. Note that the crossing pipe wall can be observed as a large area with a distance of roughly one meter in the center of the image. (b) Deviation measure image of the T-junction, obtained by subtracting expected range measurements of a synthesized cone from the raw TOF range image measurements. The left and right turns of the pipe are easily discernible. Note the occurrence of negative deviation in the center of the image, due to the cone model penetrating the crossing pipe wall.



(a) Landmark 2–3 (b) Landmark 3–2 (c) Landmark 3–4

Figure 12: Estimated distance to landmarks as function of time. X-axis: Time since start of recording in seconds. Y-axis: Estimated distance to landmark in meters. Crosses/circles: Right/left hand landmarks. Landmark numbers indicate traversal path in pipeline sketched in Figure 6.

| Section | Speed odometry (m/s) | Speed camera (m/s) | Deviation (%) |
|---|---|---|---|
| 1-2 | 0.33 | 0.38 | 15 |
| | 0.34 | 0.42 | 23 |
| 2-3 | 0.30 | 0.39 | 32 |
| | 0.30 | 0.34 | 14 |
| 3-4 (left) | 0.30 | 0.37 | 24 |
| | 0.33 | 0.42 | 27 |
| 3-4 (right) | 0.30 | 0.49 | 65 |
| | 0.33 | 0.52 | 57 |
| 3-2 | 0.30 | 0.35 | 16 |
| | 0.29 | 0.38 | 31 |
| 8-3 (left) | 0.30 | 0.36 | 21 |
| | 0.27 | 0.41 | 52 |
| 8-3 (right) | 0.30 | 0.40 | 35 |
| | 0.27 | 0.38 | 40 |
| 4-3 | 0.33 | 0.38 | 14 |
| | 0.31 | 0.36 | 15 |

Table 1: Results of estimated speed based on the tracking of detected landmarks as compared to external speed measurements (odometry). Section column indicates landmark where the robot started followed by landmark that the robot tracked. Two separate recordings were made for each landmark, these are given as consecutive lines in the table. External speed measurements was estimated manually using a stop-watch and tape measure. Camera speed was estimated by performing linear regression on distance versus time estimates.

### 5.3 Obstacle detection

In order to evaluate the proposed obstacle detection procedure, we placed objects of various sizes into the pipeline, and let the robot drive towards them, recording the data. Our main goal with this experiment was to determine whether the along-axis view, i.e. the obstacle location relative the pipe instead of relative the camera, is generated correctly and thus can be used for path planning.

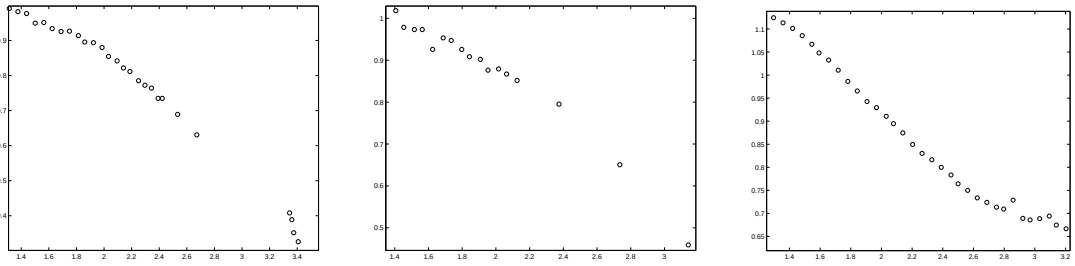
The obstacle position was found by measuring the distance to the closest object larger than $5 \times 5$ cm within the along-axis view of the image. Illustrated in Figure 13 are the results plotted as tracking plots for the obstacles over time. From these plots we see that the obstacle disappears from the graph when it gets 0.6-0.7 meter the camera. This is due the relatively narrow field-of-view (FOV) of the TOF camera, since the obstacle can no longer be positioned relative to the pipe when only small fractions of the pipe is within the camera field-of-view. Thus an along-axis view is not possible to create and the obstacle disappears from the graph when it gets too close to the camera. For distances above 0.7 meter, we see that the distance-to-obstacle is decreasing linearly as expected.

In Table 2 the estimated robot speed based on these measurements is summarized. Speed from camera measurements is found using linear regression on distance-to-obstacle versus time. Again we compare to externally measured speed, measured using stopwatch and tape measure. We see that the the two methods of estimating robot speed provide similar results, although the robot speed when approaching obstacles is underestimated by about 14% in average, with a standard deviation of 11%. Hence, the along-axis mapping provide useful results and can be used for path planning, like the procedure illustrated in Figure 5.

## 6. CONCLUSIONS AND OUTLOOK

This paper has considered the use of TOF cameras as a visual navigation aid for pipe inspection robots. Inherent flaws in the camera and the active illumination of the TOF camera coupled with cramped surroundings and reflective surfaces influence the range measurements in a detrimental way. This leads to a systematic error in range estimates, and the pipeline images take a rather conic than cylindrical shape. Hence, a method for scene segmentation based on a model that estimates deviation from the best fitting cylindrical shape has been developed, and results extended from our previous paper have been presented.

The shape distortion appears more significant in the presence of obstacles. The defects are attempted corrected for using a method based on an estimated inverse point spread function. Results from Figure 9 show



(a) Obstacle of size appr. 10x30 cm in bottom of pipe.

(b) Pipe filled up completely by obstacle.

(c) Minor obstacle of size appr. 8x8 cm size in bottom of pipe.

Figure 13: Distance to obstacles in pipeline versus time. X-axis: Time in seconds. Y-axis: Distance to obstacle in meters.

| **Obstacle** | **Speed odometry (m/s)** | **Speed camera (m/s)** | **Deviation (%)** |
|---|---|---|---|
| Box | 0.31 | 0.32 | 3 |
| | 0.31 | 0.31 | 0 |
| Small | 0.28 | 0.25 | 11 |
| | 0.30 | 0.24 | 20 |
| Blocking | 0.30 | 0.23 | 23 |
| | 0.32 | 0.23 | 28 |

Table 2: Results of estimating speed based on distance-to-obstacle as compared to external speed measurements. First column indicate three different obstacles placed into the pipeline, a box sized approximately 10x30 cm, a minor 8x8 cm obstacle, and a blocking obstacle filling the pipe. Two recordings per obstacle, presented in consecutive lines. Odometry speed indicates speed measured with stopwatch and tape measure. Camera speed indicates speed computed from linear regression on distance-to-obstacle versus time.

that after correction, the pipeline is correctly estimated as a cylinder. This indicates that the conic shape of the recorded pipeline data results from the inherent imaging system defects, and can be explained by smearing inherent in the TOF cameras, where less-bright pixels, in the presence of a bright object close to the camera, will appear closer to the camera than they actually are.

A procedure for detecting obstruction of the pipeline has been presented, and a simple approach for routing around the obstacle is discussed. As with the landmark detection, the main flaw is precise estimate of distance. Whereas speed estimates for landmark detection are overestimated, the estimated relative speed of the robot seems to be consistently underestimated for obstacles. A likely explanation for these misestimates may be the sensor specific challenges of the camera, combined with the cramped acquisition environments.

The estimated range to detected landmarks or obstacles is yet to be verified with better precision. Secondary sensors, such as accelerometer, gyroscope, contact sensors and odometry data reported from the robot motors, can provide additional information. Fusion of such sensor data for model verification and improvement would be an interesting topic for further studies. A more sophisticated algorithm for differentiation and classification of obstacles from landmarks should also be developed.

The relatively narrow field-of-view of the TOF camera poses a slight challenge for navigational tasks. As discussed in Section 5, we lose view of the pipeline when it is closer than about 0.5 meter. Using TOF cameras as the single source of visual navigation information thus means that the robot needs to operate blind when traversing obstacles or navigating junctions. We envision supplementing the information from the TOF camera with range scanners and possibly contact sensors to alleviate this. Another possible solution would be to use several inexpensive TOF cameras in an array configuration.

To summarize, TOF cameras significantly simplifies the task of scene segmentation and interpretation, and is a useful sensor as navigational aid for pipe robots. However, to obtain absolute distance estimates of detected scene objects, information from additional sensors is necessary. Further research on camera error correction should however be pursued to alleviate this requirement.

## REFERENCES


[1] R. Bostelman, T. Hong, and R. Madhavan, "Obstacle detection using a time-of-flight range camera for automated guided vehicle safety and navigation," *Integrated Computer-Aided Engineering* **12**(3), pp. 237–249, 2005.

[2] A. Prusak, O. Melnychuk, H. Roth, I. Schiller, and R. Koch, "Pose estimation and map building with a time-of-flight-camera for robot navigation," *International Journal of Intelligent Systems Technologies and Applications* **5**(3–4), pp. 355–364, 2008.

[3] F. Yuan, A. Swadzba, R. Philippsen, O. Engin, M. Hanheide, and S. Wachsmuth, "Laser-based navigation enhanced with 3d time-of-flight data," in *IEEE International Conference on Robotics and Automation*, 2009.

[4] C. Munkelt, M. Trummer, P. Khmstedt, G. Notni, and J. Denzler, *View Planning for 3D Reconstruction Using Time-of-Flight Camera Data*, vol. 5748 of *Lecture Notes in Computer Science*, pp. 352–361. Springer Berlin/Heidelberg, 2009.

[5] A. Kolb, E. Barth, and R. Koch, "Tof-sensors: New dimensions for realism and interactivity," in *CVPR Workshops*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008.

[6] A. Kolb, E. Barth, R. Koch, and R. Larsen, "Time-of-flight sensors in computer graphics," in *Eurographics 2009 - State of the Art Reports*, M. Pauly and G. Greiner, eds., pp. 119–134, Eurographics Association, Eurographics, (CH-1288 Aire-la-Ville), 2009.

[7] A. Ahrary, L. Tian, S.-I. Kamata, and M. Ishikawa, "Navigation of an autonomous sewer inspection robot based on stereo camera images and laser scanner data," *International Journal on Artificial Intelligence Tools* **16**(4), pp. 611–625, 2007.

[8] E. Rome, J. Hertzberg, F. Krichner, U. Licht, and T. Christaller, "Towards autonomous sewer robots: the MAKRO project," *Urban Water* **1**(1), pp. 57–70, 1999.

[9] F. Kirchner and J. Hertzberg, "A prototype study of an autonomous robot platform for sewerage system maintenance," *Autonomous Robots* **4**(4), pp. 319–331, 1997.

[10] Z. Jia, A. Balasuriya, and S. Challa, "Recent developments in vision based target tracking for autonomous vehicles navigation," *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE* , pp. 765–770, 2006.

[11] S. Thrun, *Robotics and Cognitive Approaches to Spatial Mapping*, ch. Simultaneous Localization and Mapping, pp. 13–41. Springer Verlag, 2008.

[12] J. T. Thielemann, G. M. Breivik, and A. Berge, "Pipeline landmark detection for autonomous robot navigation using time-of-flight imagery," in *CVPR Workshops*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008.

[13] T. Kavli, T. Kirkhus, J. T. Thielemann, and B. Jagielski, "Modelling and compensating measurement errors caused by scattering in time-of-flight cameras," in *Two- and Three-Dimensional Methods for Inspection and Metrology VI*, **7066**, p. 706604, SPIE, 2008.

[14] J. Mure-Dubois and H. Hugli, "Optimized scattering compensation for time-of-flight camera," in *Two- and Three-Dimensional Methods for Inspection and Metrology V*, P. S. Huang, ed., **6762**, p. 67620H, SPIE, 2007.

[15] P. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Journal of Computer Vision and Image Understanding* **78**(1), pp. 138–156, 2000.

[16] J. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal* **7**, pp. 308–313, 1965.

[17] S. Yanowitz and A. Bruckstein, "A new method for image segmentation," *Pattern Recognition, 1988., 9th International Conference on* , pp. 270–275 vol.1, 1988.

[18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, 2005.

[19] W. Burgard and M. Hebert, "World modeling," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, eds., pp. 853–869, Springer, 2008.

[20] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik* **1**(1), pp. 269–271, 1959.