Monocular Visual Odometry for Robot Localization in LNG Pipes

Peter Hansen, Hatem Alismail, Peter Rander and Brett Browning

Abstract— **Regular inspection for corrosion of the pipes used in Liquified Natural Gas (LNG) processing facilities is critical for safety. We argue that a visual perception system equipped on a pipe crawling robot can improve on existing techniques (Magnetic Flux Leakage, radiography, ultrasound) by producing high resolution registered appearance maps of the internal surface. To achieve this capability, it is necessary to estimate the pose of sensors as the robot traverses the pipes. We have explored two monocular visual odometry algorithms (dense and sparse) that can be used to estimate sensor pose. Both algorithms use a single easily made measurement of the scene structure to resolve the monocular scale ambiguity in their visual odometry estimates. We have obtained pose estimates using these algorithms with image sequences captured from cameras mounted on different robots as they moved through two pipes having diameters of 152mm (6") and 406mm (16"), and lengths of 6 and 4 meters respectively. Accurate pose estimates were obtained whose errors were consistently less than 1 percent for distance traveled down the pipe.**

I. INTRODUCTION

Corrosion of the pipes used in the sour gas processing stages of Liquified Natural Gas (LNG) facilities can lead to failures that result in significant damage to the infrastructure, loss of product, and most importantly fatalities and serious injuries to humans. Detailed inspection of these pipes to monitor corrosion rates is therefore a high priority.

In current industry practice, inspection is performed using Non-Destructive sensors located external to the pipe to measure wall thickness, and/or by inserting sacrificial metal samples into the pipe and measuring their loss of material rate. Example sensors include Magnetic Flux Leakage (MFL), ultrasound, and radiography (e.g. [1]). In either case, measurements can only be made in reachable locations. Pipes in LNG facilities are often densely packed and difficult to access, making complete coverage of the pipe network expensive and time consuming.

An alternative approach is to use a pipe crawling robot to measure all of the pipe surface thereby avoiding the need for extensive, and potentially unreliable, predictive models. This approach has proven highly successful for inspecting gas pipelines, e.g. Pipe Inspection Guages (PIGs)[1] and downstream networks [2], where wheel odometry and/or expensive Inertial Motion Units (IMUs) are used when pose estimates are needed.

This paper was made possible by the support of an NPRP grant from the Qatar National Research Fund. The statements made herein are solely the responsibility of the authors.

Hansen and Browning are with the Qri8 lab, Carnegie Mellon University, Doha, Qatar phansen@qatar.cmu.edu. Alismail, Rander & Browning are with the Robotics Institute/NREC, Carnegie Mellon University, Pittsburgh PA, USA, {halismai,rander,brettb}@cs.cmu.edu.

[1] http://www.geoilandgas.com/businesses/ge_oilandgas/en/prod_serv/serv/pipeline/en/inspection_services/index.htm

In our work, we aim to improve on the current technologies used for pipe inspection. Our goal is to produce a reliable, and low-cost, visual perception system that can support building high resolution registered appearance maps of the interior pipe wall which can later be used for corrosion detection and/or augmenting existing sensors. Concretely, our current system operates with a monocular camera mounted on a vehicle which moves through a pipe network. The output of the system is a multiple degree of freedom estimate of the vehicle pose at each time step. A second output of the system is a 2D metric appearance map of the inner pipe surface. Figure 1 illustrates a small section of the appearance map (stitched image) generated for two different pipes using the algorithms developed in this work. Single degree of freedom pose estimation methods (e.g. wheel odometry) would not be sufficient for producing consistent appearance maps.

In this paper, we explore and evaluate two classes of monocular visual odometry algorithms, *dense* and *sparse*, which can be used to achieve high accuracy position estimates. To date, both algorithms are designed for use only with straight cylindrical pipes (i.e. having no longitudinal curvature) with a constant radius. As with all visual odometry algorithms (e.g. [3], [4], [5]), the positions are found relative to a starting position by integrating incremental changes in camera pose. The dense algorithm estimates a change in pose using a translational motion model (image space), which is converted to a change in translation in the pipe coordinate frame using a measured scale factor ζ. The image translation between adjacent images is estimated using all overlapping pixels. The sparse algorithm measures camera egomotion using sparse scene point correspondences found between key-frames, and their estimated position on the interior surface of the pipe. This estimated position is found using the manually measured radius of the pipe. Unlike most visual odometry algorithms, the sparse method estimates the six degree of freedom camera poses incrementally to obtain robust estimates. This approach is suited for our constrained operating environment, as discussed in more detail in the following sections. The reference scale measurement ζ and the measured radius provide the mechanism for removing the monocular scale ambiguity in the egomotion estimates. We demonstrate through experiments that both can provide accurate visual odometry estimates with errors for distance traveled in a pipe consistently less than 1 percent.

This paper is organized as follows. In section 2 we provide an overview of our datasets and coordinate conventions. This information is used when describing the dense and sparse visual odometery algorithms in sections 3 and 4 respectively. In section 5 we present our results and discussion, and in section 6 we detail our conclusions and future work.

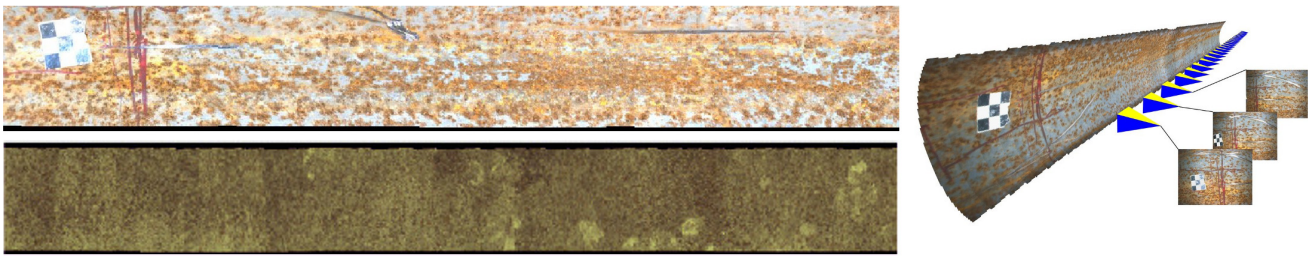

Fig. 1. Stitched images generated from approximately one hundred individual images captured in pipe 1 (top) and pipe 2 (bottom) using the sparse visual odometry algorithm (see table I). The x, y image coordinates correspond, respectively, to axial and circumferential distances in each pipe. Low resolution versions are shown which cover a small region of each pipe (approximately 0.4 meters in length down the axis of the pipe). The right image illustrates the 3D rendering of the pipe 1 stitched image with a subset of the camera frustums displayed.

TABLE I

SUMMARY OF DATASETS.

	Pipe 1 (Pittsburgh)	Pipe 2 (Qatar)
Material	Carbon steel	
Length	6m	4m
Outer diameter	152.40mm (6")	406.40mm (16")
Inner diameter	153.32mm	387.56mm
Camera	PGR Dragonfly2 1024 × 768, 7.5fps	PGR Firefly 640 × 480, 30fps
Horiz. FOV	70°	25°
LEDs	4 × 3.5W (280 lumen)	
Ground truth	5844.4mm	3391.0mm
Sequences (# images)	a: forward (4256) b: forward (4176) c: forward & rev. (3369)	forward & rev. (2392)

II. DATASETS AND COORDINATE CONVENTIONS

We have collected datasets using two pipes; pipe 1 (located in Pittsburgh), and pipe 2 (located in Qatar). A summary of the datasets is given in Table I. For each pipe, we retrofitted a small robotic platform with a camera and lens assembly and four high intensity Light Emitting Diodes (LEDs). Figure 2 shows the robot platforms and an example image. Each camera was positioned so that the uppermost interior surface of each pipe, directly above the robot, was observed. This configuration enables high resolution images of the surface to be obtained, which are necessary for both estimating motion and producing the appearance maps. Robot motion in each pipe is primarily along the h direction (see figure 3), although non-negligible high frequency motion occurs in all other dimensions. Images from each camera are logged and processed off-line.

A. Gain-mask correction

High intensity LEDs provide the only light source in each pipe. To account for their non-uniform lighting distribution and vignetting, we pre-process each image by dividing it with a *gain mask*. The gain mask is the mean of 50 images taken by the camera observing a white piece of card affixed to the interior surface of the pipe with the LEDs operating at their specified power. Figure 4 shows the gain mask for pipe 1, and the gain corrected version of the image in figure 2.

B. Ground truth

For each pipe, we manually augment the uppermost interior surface at each end with a fine mark. The ground truth measurement for each pipe is the measured distance δh between the two marks. Since these marks are visible in the first and last image in each dataset, we can compare visual odometry estimates for distance traveled in the h direction (see figure 3) to the ground truth measurement.

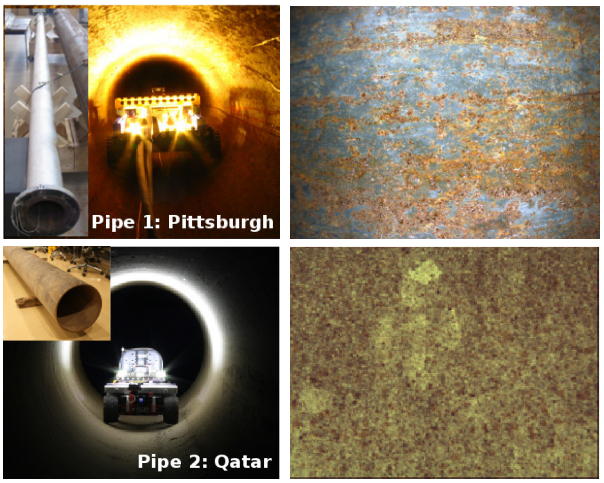

Fig. 2. The two pipes and robotic platforms used in this work (left), and example images from each camera (right).

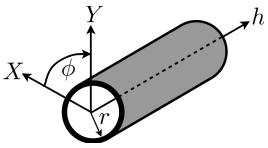

Fig. 3. The coordinate system for a straight cylindrical pipe with constant internal radius r.

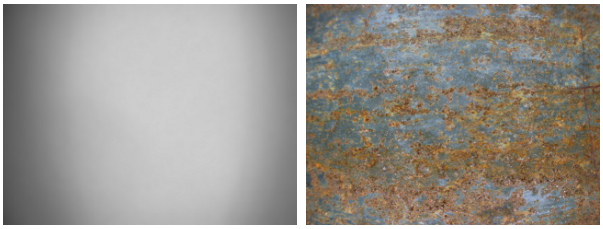

Fig. 4. The gain mask image (left), and gain corrected image (right) for pipe 1. The corrected image shows dramatically improved intensity uniformity compared to the original image (upper right image of figure 2).

C. Reference Measurements: Monocular Scale Ambiguity

Without any prior knowledge, monocular visual odometry algorithms are unable to estimate translational motion with

metric units. The dense and sparse algorithms both use a reference scale measurement to resolve this ambiguity. The dense algorithm uses a scale measurement ζ (pixels/meter), which relates a change in pixel coordinates in undistorted[2] perspective images to a metric change in position in the pipe coordinate frame. The scale factor ζ is obtained by imaging a reference pattern fixed to the surface of the pipe, and then finding the ratio of the pixel distance between points in the pattern to their known metric distance. This process is completed before collecting datasets. The reference scale measurements that were obtained for pipe 1 and pipe 2 are $\zeta = 10.47 \times 10^3$ and $\zeta = 7.16 \times 10^3$ respectively.

The sparse monocular algorithm enforces that all world points observed in the images lie on the interior surface of a straight cylindrical pipe with constant radius. Hence, we manually measure the inner diameters of each pipe (see Table I, and [6] for details).

D. Coordinate Conventions

Referring to figure 3, we define $\mathbf{X} = (X, Y, h)^T \in \mathbb{R}^3$ as the coordinate of a world point in the pipe's coordinate frame. The coordinate of a scene point, constrained to lie on the interior surface of the pipe, can be parameterized as:

$$\mathbf{X} = \begin{pmatrix} r\cos\phi & r\sin\phi & h \end{pmatrix}^T, \tag{1}$$

where $\phi = \tan^{-1}(Y/X)$. The pose P_t of the camera relative to the pipe at time t is written as:

$$P_t = \begin{bmatrix} R & -R\,\mathbf{C} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{2}$$

where R is a 3×3 rotation matrix, and $\mathbf{C}$ is a 3×1 Euclidean position vector of the camera with respect to the origin of the pipe. The pose P_t defines the transform of a world point with coordinate $\mathbf{X}_i$, in the pipe coordinate frame, to coordinate $\mathbf{Y}_i$ in the camera coordinate frame:

$$\begin{pmatrix} \mathbf{Y}_i^T & 1 \end{pmatrix}^T = P_t \begin{pmatrix} \mathbf{X}_i^T & 1 \end{pmatrix}^T. \tag{3}$$

III. DENSE MONOCULAR

The dense monocular algorithm uses direct, or pixel-based, image registration (alignment) to estimate camera egomotion [7], [8]. We use a two degrees of freedom parametric model [9], whereby adjacent images I_{t-1}, I_t are related by a single $\delta\mathbf{x} = (\delta x, \delta y)^T$ pixel shift (image translation):

$$I_t(\mathbf{x}) = I_{t-1}(\mathbf{x} - \delta\mathbf{x}). \tag{4}$$

We set the initial camera pose P_0 as[3]:

$$P_0 = \begin{bmatrix} R^{-1} & \mathbf{C} \\ \mathbf{0}^T & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 1 & 0 & C_X \\ 0 & 0 & 1 & C_Y \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}. \tag{5}$$

The change in pose $Q = P^{-1}$ between adjacent images, measured in the pipe coordinate frame of reference, is

$$Q = \begin{bmatrix} I_{3\times3} & \delta\,\mathbf{C} \\ \mathbf{0}^T & 1 \end{bmatrix}^{-1}, \quad \delta\,\mathbf{C} = \frac{1}{\zeta}(\delta y, 0, \delta x)^T. \tag{6}$$

[2]Using the Matlab Calibration Toolbox `http://www.vision.caltech.edu/bouguetj/calib_doc/`

[3]The camera's principal axis is in the direction of the pipe's y-axis, and the camera's x axis is in the direction of the pipe's h axis.

The scale factor ζ discussed in section 2 allows the image translation $\delta\mathbf{x}$ to be converted to a change in camera pose $\delta\mathbf{C}$, in the pipe coordinate frame, with metric units. The unknown offsets C_X and C_Y in (5) could be manually measured if desired. However, they have no influence on the estimate of the distance the robot travels in the h direction down the pipe. To estimate the image translation $\delta\mathbf{x}$, we use a *full-search* method followed by an iterative *model-based* refinement.

A. Full-search

An initial integer estimate of the image translation $\delta\mathbf{x}$ is obtained by sliding one of the images over the other on a $2D$ grid in image space. The objective is to find the alignment (image translation $\delta\mathbf{x}$) that minimizes a function of image dissimilarity. In this work, we use the Sum of Absolute Difference (SAD) normalized by the area of overlap to measure the quality of registration.

B. Iterative model-based refinement

In an attempt to improve upon the initial integer estimate of $\delta\mathbf{x}$ obtained using full-search, we proceed to use the iterative model-based registration framework proposed by [9] for sub-pixel refinement. The translational motion $\delta\mathbf{x}$ between image pairs is approximated iteratively using a Gauss-Newton refinement process to minimize the Sum of Squared Differences (SSD) between adjacent images, normalized by the area of overlap (see [9] for details). Iteration continues until convergence, or an empirically selected maximum number of times is exceeded.

Iterative registration schemes are prone to converge in a local minima if the overlap between images is small. Obtaining an initial estimate for $\delta\mathbf{x}$ using the full search method guarantees, in most cases, that the iterative refinement will converge to a more accurate estimate of the correct motion.

IV. SPARSE MONOCULAR

The sparse monocular algorithm estimates camera egomotion based on the changes in positions of sparse keypoint correspondences in adjacent images. These changes are typically referred to as the *sparse optical flow*, and are used as the basis for many monocular visual odometry algorithms [3], [10]. However, most visual odometry algorithms are designed to operate in unstructured environments where the relative Euclidean coordinates of scene points cannot be derived from a single image; they must be triangulated from pairs or sequences of images [11]. As discussed, we constrain the camera to lie within a straight cylindrical pipe with a fixed, and known, radius r. The Euclidean coordinate $\mathbf{X}_i$ of any pixel in an image can therefore be derived given the radius r and camera pose P_n. Camera egomotion is estimated using the derived world point coordinates $\mathbf{X}$ of the sparse keypoint correspondences. Using the world point coordinates enables the camera egomotion to be estimated with metric units of translation. Hence, the monocular scale ambiguity is avoided.

A. Obtaining Scene Point Correspondences

A region-based Harris corner detector [12] is used to identify keypoints in the original (distorted) gain-corrected gray-scale images[4]. Each image is divided into an equally

[4]Adjacent images contain minimal scale change, and we have found no advantages using scale-invariant keypoints (e.g. SIFT [13]).

spaced 6×8 grid, and the 20 most salient keypoints in each cell retained after non-maxima suppression of the saliency values using a 7×7 pixel wide window — the saliency is the Harris 'cornerness' score. A region-based scheme is used to ensure that there is a uniform distribution of keypoints throughout the image. Sub-pixel keypoint positions are obtained using a two-dimensional quadratic interpolation of the Harris cornerness score based on the scheme developed in [13]. A 128-dimensional SIFT descriptor [13] is then evaluated for each keypoint from the gray-scale intensity values within a fixed sized region surrounding it. Since the cameras used have been calibrated, the pixel position $\mathbf{x}_i$ of each keypoint is mapped to a spherical coordinate $\boldsymbol{\eta}_i$, $\mathbf{x}_i \mapsto \boldsymbol{\eta}_i$, which defines a ray in space originating from the camera center.

Given any two images, corresponding keypoints are found using the ambiguity metric [13] for the SIFT descriptors with a mutual consistency check. However, the correspondence between every adjacent pair of images are not used to estimate motion since many are separated by only a few pixels difference. We use a method similar to that of Mouragnon et al [14] and Tardif et al [10] to automatically select only key frames (images) that are used to obtain the visual odometry estimates. Starting with the first image I_0 in the sequence, we keep finding the correspondences between image I_0 and I_j, where I_j is the j^{th} image in the sequence. When the number of correspondences between image I_0 and I_j falls below a threshold, or the median magnitude of the sparse optical flow vectors is above some threshold, the set of correspondences between images I_0 and I_{j-1} are kept, and images I_0 and I_{j-1} are assigned a camera pose $P_{n=0}$ and $P_{n=1}$ respectively. This process is then repeated starting at image I_{j-1}, and continued for the remainder of the sequence. The images I_n used to compute the camera egomotion are the key-frames. We attempt to remove any outliers in the set of correspondences between key frames using RANSAC [15] and the five-point algorithm in [16].

B. Euclidean coordinates of scene points

Recall that the image coordinate $\mathbf{x}_i$ of any keypoint can be mapped to a spherical coordinate $\boldsymbol{\eta}_i$ using the camera calibration parameters. We parameterize a world point in the camera coordinate frame, which is constrained to lie on the interior surface of a pipe with radius r, as $\mathbf{Y}_i = \kappa_i \boldsymbol{\eta}_i$, where κ_i is a scalar. From (3), $\mathbf{X}_i = R^{-1}\mathbf{Y}_i + \mathbf{C}$, which is the coordinate of the point in the world (pipe) coordinate frame. Letting $R = [\mathbf{R_1}\,\mathbf{R_2}\,\mathbf{R_3}]$, and referring to figure 3 and (1), the coordinate $\mathbf{X}_i = (X, Y, h)^T$ must satisfy the constraint:

$$r^2 = X^2 + Y^2 \tag{7}$$

$$= (\kappa_i \mathbf{R_1}\boldsymbol{\eta}_i + C_X)^2 + (\kappa_i \mathbf{R_2}\boldsymbol{\eta}_i + C_Y)^2 . \tag{8}$$

Expanding (8) produces a quadratic in κ_i, which has one positive and one negative solution if the camera is inside the pipe. The positive solution is correct since the point must be in front of the perspective camera, and defines the Euclidean scene point coordinate $\mathbf{Y}_i = \kappa_i \boldsymbol{\eta}_i$, where $\mathbf{X}_i = R^{-1}\mathbf{Y}_i + \mathbf{C}$.

C. Visual Odometry Estimation

Six degree of freedom (6-DOF) motion estimates are obtained using a number of steps:

1) Initial one degree of freedom (1-DOF) estimation.
2) Optimization of initial camera position.
3) Six degree of freedom (6-DOF) refinement.

Before discussing each of these steps, it is important to introduce the *global* index g for each world point. Assume that the same world point is detected in multiple images, and the keypoint correspondences found between key frames enables us to identify that these keypoints do in fact belong to the same world point. If the pose P is know for each of the images, then the coordinate $\mathbf{X}$ of this point could be derived from any of the images using the procedure described in the previous section. We assign this world point a global index g, $\mathbf{X}^g$, and store each of the j estimates of its position obtained from different images, $\mathbf{X}^g_j$.

1) Initial one degree of freedom estimation: Since the camera motion in a straight cylindrical pipes was constrained primarily to a change in translation δh down the pipe, the initial estimate of each camera pose P_{n+1} is

$$P_{n+1} = \begin{bmatrix} R_n & -R_n\,(\mathbf{C}_n + \delta\,\mathbf{C}) \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{9}$$

where $\delta\,\mathbf{C} = (0, 0, \delta h)^T$, which has only a single degree of freedom $\delta\,h$. The estimate for δh is obtained using all relevant prior information in the sequence, here the coordinates $\mathbf{X}$ of all the world points derived before key-frame $n+1$.

For an estimate of P_{n+1}, we derive the coordinates of the world points using the method described previously. Denote these points $\hat{\mathbf{X}}$. A non-linear optimization (Levenberg-Marquardt) is used to find the estimate of $\delta\,h$ in equation 9 which minimizes the error

$$\epsilon = \sum_g \sum_j \left[\mathbf{X}^g_j - \hat{\mathbf{X}}^g_j\right]^2, \tag{10}$$

where $\sum_j$ is the summation for all j estimates of the world point with global index g. We could have projected the images coordinates $\mathbf{X}$ to the current image, and then minimized the distance to their observed (detected) pixel positions. However, the derived positions $\mathbf{X}, \hat{\mathbf{X}}$ all have an approximately equal uncertainty[5], and we have found the method used to provide satisfactory results.

2) Optimization of initial camera position: We perform a batch optimization, using the first ten key-frames, to improve upon the initial manually measured estimate for $\mathbf{C}_{n=0}$ (we keep $C_h = 0$ for the first key-frame).

The optimization used does not minimize the errors between the scene point coordinates $\mathbf{X}$. The reason is that the coordinates of all of these points change during a batch optimization, and it is not clear how a suitable normalization factor can be selected — moving the camera closer to the pipe surface minimizes the dispersion of the scene points and the magnitude of the errors. For this reason, the error minimized is defined in image space with respect to the coordinates of the keypoints detected in the original images.

The estimate for P_0 is set to the same initial pose used by the dense algorithm, see (5), using the manually measured values for C_X and C_Y. For each iteration, a 1-DOF estimate

[5]This applies for the camera configuration used, which has a narrow field of view, and assuming a (fixed scale) Gaussian uncertainty of the detected keypoint positions.

for each of the ten cameras $P_{0,\cdots,9}$ is obtained, and the world coordinates $\mathbf{X}$ of all the corresponding keypoints are found from their image coordinates $\mathbf{x}$. These points can then be mapped to image coordinates $\hat{\mathbf{x}}$ in any of the other images where they were found. The error ϵ' minimized is defined in image space as

$$\epsilon' = \sum_g \sum_j \left[\mathbf{x}_j^g - \hat{\mathbf{x}}_j^g\right]^2, \tag{11}$$

where, again, $\sum_j$ is the summation for all j estimates of the world point with global index g. Note that $\mathbf{x}^g$ and $\hat{\mathbf{x}}^g$ may appear in two or more images in the set.

3) Six degree of freedom refinement: The initial 1-DOF estimate is used to obtain a reasonably accurate initial estimate of camera pose. We refine the pose estimates using a 6-DOF model and a sliding window scheme. After each $a = \tau N$ key frames, where τ is any integer and $N = 50$ is a constant, the 6-DOF refinement for the previous $2N$ frames is implemented. This implementation is performed separately for each frame in the window, and no 'batch' optimization of all the frames is used.

For frame P_k, where $1 \leq k \leq a$, the coordinates $\hat{\mathbf{X}}$ for each of the correspondences in the image are found for a given six degrees of freedom estimate of P_k — these points are no longer assumed to belong to the set of world points $\mathbf{X}$. The optimized 6-DOF estimate is the one which minimizes (10). The non-linear optimization is implemented using Levenberg-Marquardt, where the camera rotation R is parameterized using quaternions. This process is implemented sequentially from the first to last image in the window. Since this method limits the degree by which the pose can change, it is run for an empirically selected number of times, which in the following experiments is two.

D. Selection of incremental addition of degrees of freedom

When an initial estimate of a camera's pose P_n is obtained, only information from the previous frames can be used. However, during the 6-DOF sliding window optimization, a camera's pose P_n is optimized using information from all frames. Attempting to estimate a camera's 6-DOF pose using only information from previous frames proved unreliable and inaccurate — this is the same reason why we don't use the five-point algorithm to obtain an initial egomotion estimate. One reason for this is the difficulty in reliably decoupling rotational and translational motion when using narrow field of view (i.e. typical perspective) cameras [17], particularly when: there are minimal depth discontinuities in the scene [18]; there are small changes in camera rotation and/or translation [19], [20]; the focus of expansion or contraction is outside the camera's field of view [17]. The last two factors in particular exist for our camera configuration (see [6] for details). Using information from all frames to estimate the 6-DOF motion enabled significantly more reliable and accurate results to be obtained.

V. RESULTS AND DISCUSSION

A. Results

Visual odometry results were obtained, for the datasets summarized in table I, using the dense and sparse algorithms. The estimates of the distance traveled down the axis of the pipe versus the ground truth measurements are summarized in table II (the errors are the ground truth values minus the absolute estimated change δh of camera pose). The absolute percentage errors are also recorded in the table. For the datasets (Pipe 1c, Pipe 2) where the robot moves forward down the pipe, and then reverses in the opposite direction, the total ('loop closure') error is $\epsilon = \text{fwd. err} - \text{rev. err}$. The total absolute percentage error is then

$$\epsilon(\%) = 100 \times |\text{fwd. err} - \text{rev. err}| / \left(2 \times \text{ground truth}\right). \tag{12}$$

TABLE II

DENSE AND SPARSE ALGORITHM RESULTS. ALL PERCENTAGES ARE ABSOLUTE VALUES. SEE TABLE I FOR GROUND TRUTH.

Dataset	Metric	Dense		Sparse	
Pipe 1a	error mm	-9.0	(0.154%)	17.9	(0.306%)
Pipe 1b	error mm	42.4	(0.725%)	16.9	(0.289%)
Pipe 1c	Fwd. error mm	8.7	(0.149%)	7.5	(0.128%)
	Rev. error mm	-21.1	(0.361%)	3.5	(0.060%)
	Tot. error mm	29.8	(0.255%)	4.0	(0.034%)
Pipe 2	Fwd. error mm	28.4	(0.838%)	7.1	(0.209%)
	Rev. error mm	-99.0	(2.919%)	20.7	(0.619%)
	Tot. error mm	127.4	(1.879%)	-13.6	(0.201%)

B. Discussion

The results in table II show that the dense and sparse algorithms are capable of finding accurate visual odometry estimates with respect to distance traveled down the pipe, with errors consistently less than 1 percent. Note that neither of the pipes used contained any significant geometric deviations from the straight cylindrical, uniform radius pipe assumption used by both the dense and sparse algorithms (e.g. longitudinal curvature, change in radius, non-circular profile). We would expect less accurate visual odometry results to be obtained if they were introduced and the experiments repeated. Overall, the sparse algorithm performs marginally better than the dense algorithm. A number of factors which influence the accuracy of the visual odometry estimates are discussed here.

1) Gain-correction: Non-uniform light distribution in the images severely impacts the accuracy of the dense egomotion estimates since the pixels themselves (intensity and gradient) are used to estimate the image translation $\delta\mathbf{x}$. When using the original (not gain-corrected) images, we observed the full search estimates of $\delta\mathbf{x}$ incorrectly converging on solutions which align the non-uniform light patterns in the images. Our gain correction procedure is only approximate, and cannot account for specularities in the images — we only observed small specularities in some of the images, and have since used polarizing materials to limit them. As a result the gain corrected images may contain some non-uniform lighting which limits the accuracy of the dense egomotion estimates. Any remaining non-uniform lighting in the gain corrected images will negatively impact the accuracy of keypoint localization, and as a result the accuracy of the sparse egomotion estimates. However, non-uniform lighting effects the sparse egomotion estimates far less than that by which it effects the dense egomotion estimates.

2) Inherent motion model assumptions: The dense monocular algorithm assumes that image pairs are related by an image shift $\delta\mathbf{x}$, which corresponds to a 2-DOF change in camera position (i.e. translation) in the pipe. Referring to

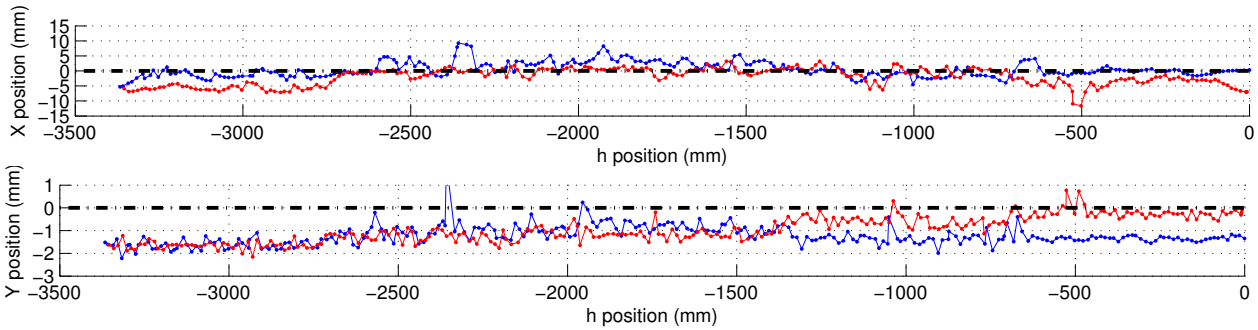

Fig. 5. The estimate of the camera position $\mathbf{C} = (X, Y, h)^T$ in pipe 2 obtained using the sparse algorithm: robot traveling forward (blue), and robot traveling in reverse (red). The high frequency change in position X is due primarily to the robot design.

figure 5, the Y coordinate of the camera changes significantly when traveling in the reverse direction – the robot 'climbed' up the side of the pipe by rotating about the pipe's axis. The dense monocular algorithm is unable to correctly model this motion. As a result, the pose estimates for pipe 2 (rev) obtained using the dense algorithm are poor in comparison to the 6-DOF estimate found using the sparse algorithm.

3) Pipe diameter and camera angle of view: Referring to table I, pipe 1 has a smaller diameter than pipe 2, and the horizontal angle of view of the lens used with pipe 1 is greater than that for pipe 2. This means that, compared to pipe 2, the images in the pipe 1 datasets exhibit a greater degree of foreshortening, and the distance of the world points from the camera in pipe 1 are less.

Foreshortening negatively impacts the dense monocular egomotion estimates since the algorithm assumes that the camera is observing a planar scene (translational model). Although foreshortening presents challenges during keypoint detection and matching using the sparse algorithm, it has little impact when finding the sparse egomotion estimates.

The ability to image scene points at a small distance from a high resolution camera, using a large angle of view lens, is desirable for visual odometry applications [17], [18], [20]. Therefore, we would expect the accuracy of the sparse visual odometry estimates obtained for pipe 1 to be better than those obtained for pipe 2. The results obtained support this claim.

VI. CONCLUSIONS AND FUTURE WORK

We have investigated two monocular visual odometry algorithms, dense and sparse, that are designed to estimate camera pose in a straight cylindrical pipe. This is a first step towards a visual perception system for LNG pipes inspection. Importantly, knowledge of the scene structure (i.e. a straight cylindrical pipe with constant radius) is used by both to resolve the monocular scale ambiguity in their visual odometry estimates. The algorithms were evaluated on different datasets taken in different pipes. In the experiments presented, both were able to estimate camera pose within 1% accuracy of the ground truth. However, the more sophisticated sparse algorithm was able the outperform the dense algorithm due to its ability to model 6-DOF motion, and its relative invariance to non-uniform image lighting. In ongoing work we are exploring different stereo camera camera configurations which may be used to produce high resolution, and high accuracy, estimates of the internal 3D structure of pipes.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of Mohamed Mustafa (CMQ) and Joey Gannon (RI-NREC) in developing the dataset collection hardware.

REFERENCES

[1] J. Nestleroth and T. Bubenik, "Magnetic flux leakage (MFL) technology for natural gas pipeline inspection," Battelle, Report Number GRI-00/0180 to the Gas Research Institute, Tech. Rep., 1999.
[2] H. Schempf, "Visual and nde inspection of live gas mains using a robotic explorer," *JFR*, vol. Winter, 2009.
[3] D. Nistér, O. Naroditsky, and J. Bergend, "Visual odometry for ground vehicle applications," *JFR*, vol. 23, no. 1, pp. 3–20, January 2006.
[4] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *JFR*, vol. 24, no. 3, pp. 169–186, March 2007.
[5] A. Levin and R. Szeliski, "Visual odometry and map correlation," in *CVPR*, June 2004, pp. 611–618.
[6] P. Hansen, H. Alismail, P. Rander, and B. Browning, "Towards a visual perception system for pipe inspection: Monocular visual odometry," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-10-22 / CMU-CS-QTR-104, July 2010.
[7] R. Szeliski, "Image alignment and stitching: A tutorial," in *Foundations and Trends in Computer Graphics and Vision*. Now Publishers Inc., 2006.
[8] A. Ardeshir, *2-D and 3-D Image Registration*. Wiley & Sons, 2005.
[9] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *ECCV'92*, 1992, pp. 237–252.
[10] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *Proceedings IROS*, 2008, pp. 2531–2538.
[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.
[12] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings Fourth Alvey Vision Conference*, 1988, pp. 147–151.
[13] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[14] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3D reconstruction," in *CVPR*, 2006.
[15] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comms. of the ACM*, pp. 381–395, 1981.
[16] D. Nistér, "An efficient solution to the five-point relative pose problem," *PAMI*, vol. 26, no. 6, pp. 756–770, June 2004.
[17] J. Gluckman and S. Nayar, "Ego-motion and omnidirectional cameras," in *CVPR*, 1998, pp. 999–1005.
[18] K. Daniilidis and H.-H. Nagel, "The coupling of rotation and translation in motion estimation of planar surfaces," in *CVPR*, June 1993, pp. 188–193.
[19] D. Nistér, "Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors," in *ECCV*, 2000, pp. 649–663.
[20] J. Neumann, C. Fermüller, and Y. Aloimonos, "Eyes form eyes: New cameras for structure from motion," in *Proceedings Workshop on Omnidirectional Vision (OMNIVIS)*, 2002.