A vision-based system for internal pipeline inspection

Author 1, Author 2, Author 3, Author 4

***Abstract*—The internal inspection of large pipeline infrastructures, such as sewers and waterworks, is a fundamental task for the prevention of possible failures. In particular, visual inspection is typically performed by human operators on the basis of video sequences either acquired on-line or recorded for further off-line analysis. In this work, we propose a vision-based software approach to assist the human operator by conveniently showing the acquired data and by automatically detecting and highlighting the pipeline sections where relevant anomalies could occur.**

I. Introduction

The regular internal inspection of pipelines is a fundamental maintenance task to guarantee the correct functionality of the infrastructure and to prevent incidents. Even if the pipeline is designed to last for a long period of time, it can be affected by a wide range of problems, such as cracks, collapses, defective junctions, obstructions, etc. Ignoring such defects could lead to system inefficiency, economic losses, environmental issues, or even potential dangers for human beings. As an example, Figure 1 was taken in downtown Florence, Italy, on May 25, 2016. In this case, an undetected waterworks leakage weakened the soil structure, causing a ground collapse, roughly 200×7 meters wide, and damaging dozens of parked vehicles. Subsequent investigations showed that the leakage was due to an old underground pipe, dated back to the 1950s, whose conditions were not regularly monitored [1].

Fig. 1. A ground collapse due to a waterworks leakage.

Visual inspection of the internal surface of pipelines is usually done by means of robotic rovers or floating platforms, mounting proper camera sensors and lights. The inspection can be done either on-line or off-line. In the first case, a human operator can see the acquired video as the inspection is being executed, and he can monitor the internal surface by means of a motorized camera head, which can be manually oriented. The advantage of on-line approaches is that defects can be detected in real-time, however it is often unpractical in long pipelines because of the lack of a reliable communication between the sensor and the display base. In the off-line approaches, a video sequence is recorded on-board, and it is later retrieved when the sensor is extracted from the pipe. These approaches are more common for long pipelines, however the camera cannot be oriented toward the relevant areas to be monitored. To overcome this issue, wide-angle sensors (e.g. hemispheric cameras) could be employed, in order to acquire images of the whole internal pipe surface surrounding the camera. However, the final video sequence must still be analyzed off-line by a human operator. This is an extremely long and tedious task, since on average a trained operator can approximately inspect 100 meters per hour, as reported by the company funding this project.

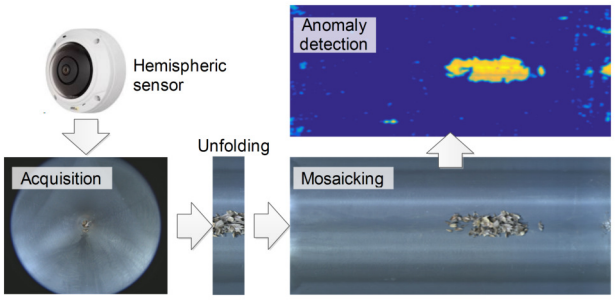


Fig. 2. System architecture.

The aim of the proposed work is to define the computer vision and machine learning algorithms that can be employed to ease the task of human operators. The system acquires video sequences with a hemispheric camera, unfolds them to ease the visual inspection, and automatically highlights the zones where a potential defect could be detected (Figure 2). We assume that a proper support to move the camera inside the pipeline exists, and it can keep the camera with its optical axis parallel to the pipe axis. We also assume a circular pipe section. The optimal scenario involves empty pipes, where the internal surface is fully visible. If this is not the case, as in many real-world applications like waterworks, the system can perform visual inspection only on the off-water surface.

In section II we give a brief review on the state of the art on pipeline inspection systems. Sections III and IV discuss how the images acquired by a hemispheric camera can be used to create unfolded views of the pipe internal surface, while in section V these images are automatically analyzed to detect anomalous image portions. In section VI experimental results are shown and discussed. Finally, section VII reports conclusions and a possible idea for future developments.

Affiliation here

II. Related work

Most of the recent works in literature concerning pipeline inspection are focused on hardware specifically designed for this task, in particular inspection robots [2]–[4]. Several works also deal with the different inspection technologies that could be adopted. As discussed in [5], [6], there are two main categories of inspection methods: direct methods and indirect methods. Direct methods include all the automated and manual visual inspection approaches, such as closed-circuit television (CCTV) [7], laser approaches [8], [9], as well as non-destructive testing approaches such electromagnetic methods [10], [11], acoustic methods [12], [13] and ultrasound methods [14]–[16]. Indirect methods, instead, include all the approaches concerning water audit, flow testing and measurement of soil resistivity [17]–[19] to determine the risk of deterioration. CCTV systems are slow, costly and have limited accuracy usually caused by human or environmental factors, while laser, acoustic and ultrasound methods suffer from lack of resolution and an inability to detect water inflow [20].

The research on automated vision-based approaches is more limited. Some works focus on finding the position of the inspection robot inside the pipe. This is achieved through Visual Simultaneous Localization and Mapping (VSLAM) [21] or visual odometry [22] algorithms, which are also used to perform a 3D reconstruction of the pipe [23]. Usually, these systems combine RGB cameras with other sensors such as Time of Flight cameras and lasers in order to obtain more accurate measurements.

The detection of anomalies in traditional systems is typically left to human operators. However, some works exist trying to extract visual information about specific anomalies such as cracks, corrosion, holes, obstructions or gaps. For example, Huynh et al. [20] use stereo vision to reconstruct a depth map of the observed pipe section. The map allows to detect all those defects that can be identified by means of their discrepancy with the 3D model of the pipe, such as the presence of debris or roots. Ting et al. [24] instead use a catadioptric omnidirectional sensor to acquire images that are then unfolded using an approach similar to the one proposed here. They also propose a set of geometric image properties to explicitly identify pipe cracks. More generally, image features can be combined with pattern recognition or machine learning approaches in order to explicitly recognize specific defects [25]. For example, Wu et al. [26] use Maximum Response filter banks to train ensemble classifiers, while Mashford et al. [27] use Support Vector Machines for the classification task.

All the mentioned works try to detect and recognize pipe defects using an explicit labeling approach: visual features are used to train a classifier to recognize cracks, holes, etc. This approach however is generally prone to errors because of the large intra-class variability: for example, there are many variables that influence the visual aspect of a crack (e.g. shape, length, depth, pipe material, etc.) and thus it is non-trivial to train a classifier to recognize all the possible cracks. At the same time, the nature of anomalies is extremely wide, ranging from surface defects to the presence of occlusions or misaligned joints. It is thus difficult to take in consideration all the possibilities. The method proposed in this work tries to overcome all these limitations by adopting an anomaly detection strategy. By analyzing the visual aspect of the pipe internal surface, a normality model is built, allowing the detection of any image patch that is discordant with the model. The system thus renounces to give an explicit label to each defect, focusing on the detection of visual anomalies that are later submitted to human evaluation. The system can thus bypass intra-class variance, and it can actually handle any type of unknown defects, as long as they are visually detectable. To the best of our knowledge, no other works adopted this approach yet in the field of pipeline inspection.

III. Image unfolding

The first module of the proposed system is the image unfolding block. This module acquires a video sequence and unfolds it to show the internal surface of the pipe on a planar view, as shown in Figure 3. The unfolding equations highly depend on the type of camera used, thus a comment on the hardware architecture is needed. As mentioned in section I, the system requires a simultaneous view of the entire internal surface of the pipe surrounding the camera. This can be achieved by either catadioptric optics or wide-angle (hemispheric) optics. Catadioptric optics are composed of a standard lens set and a frontal mirror, they can achieve good resolution on the side areas, but on the other hand the image unfolding is complex and highly dependent on mirror shape (parabolic, ellipsoid, etc. [28]). Catadioptric systems are also typically less widespread and more expensive than traditional systems. Hemispheric optics, on the other hand, can have lower spatial resolution near image borders, but their market availability is higher and are typically cheaper. Moreover, to the best of our knowledge, the image unfolding procedure is the same for almost all the optics currently available, thus allowing an higher flexibility in system design. The following sections describe how image unfolding can be achieved using hemispheric optics.

A. Camera calibration

Figure 4 shows a typical image acquired by a hemispheric camera. The optic acquires a circular image, however the imaging sensor has a rectangular shape, resulting in a final image with black areas on the sides. Hemispheric image unfolding requires the knowledge of some basic image properties, such as the center of the circular region (which not necessarily coincides with the image center) and its radius. The process of camera calibration aims at identifying the required parameters.

In order to calibrate the camera, we require that a uniform, white, bright area is observed, so that the circular region can be segmented from the side black patches. An image thresholding is then performed, in order to transform a single-channel acquired image I in a binary image B defined as:

$$B(x,y) = \begin{cases} 1 & \text{if } I(x,y) \geq t \\ 0 & \text{if } I(x,y) < t \end{cases} \quad (1)$$

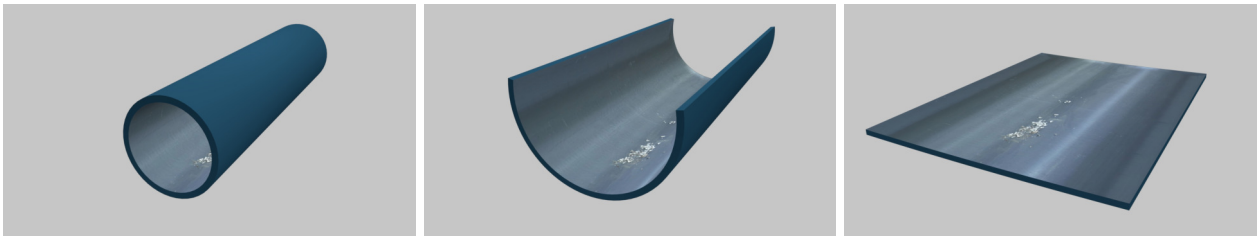


Fig. 3. The process of pipe unfolding.

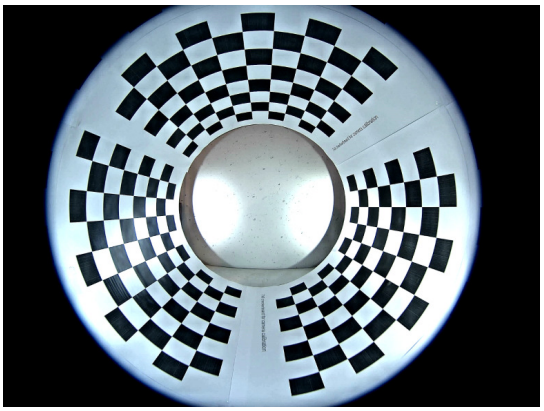


Fig. 4. Image acquired by a hemispheric camera inside a pipe whose internal surface has been covered by a checkerboard test pattern.

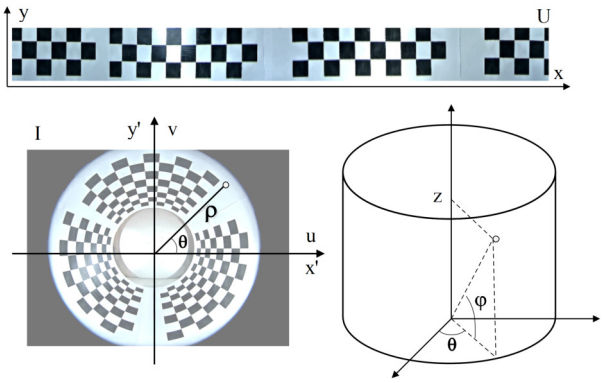


Fig. 5. The unfolded image U, the hemispheric image I and the correspondence between cylindrical and polar reference systems.

where t is a threshold, defined as $t = (\min_{x,y} I(x,y) + \max_{x,y} I(x,y))/2$. The image center (x_c, y_c) can now be defined as the mean position of pixels set to 1 in the binary image:

$$\begin{aligned} x_c &= \sum_x \sum_y x B(x,y) / \sum_x \sum_y B(x,y) \\ y_c &= \sum_x \sum_y y B(x,y) / \sum_x \sum_y B(x,y). \end{aligned} \tag{2}$$

The radius R is defined as the minimum distance of a black pixel from the center:

$$R = \min_{(x,y)|B(x,y)=0} \sqrt{(x-x_c)^2 + (y-y_c)^2}. \tag{3}$$

B. Unfolding

Here we assume that the camera is placed at the center of the pipe, the off-center case is described in the next subsection. Given an image I acquired by a hemispheric camera and the desired unfolded image U, the unfolding procedure requires the knowledge of the function $f : (x', y') \in I \mapsto (x, y) \in U$. Actual unfolding however uses $f^{-1} : U \mapsto I$ since it allows to find a (possibly interpolated) value for each pixel of the unfolded image. The mapping can be split in three main steps:

- map pixel coordinates in the unfolded image to cylindrical coordinates representing points on the pipe surface;
- convert cylindrical coordinates to spherical coordinates centered on the hemispheric optics;
- map spherical coordinates to pixel coordinates in the hemispheric image.

The reference systems are shown in Figure 5. Here, (θ, z) are the cylindrical coordinates, while (θ, φ) are the spherical ones. Without loss of generality, the cylindrical radius is arbitrarily set to 1. In the spherical system, the distance from the origin is ignored, since (θ, φ) alone are sufficient to uniquely define a point on the pipe surface.

As a first step, cylindrical coordinates (θ, z) must be defined starting from the pixel coordinates (x, y) in the unfolded image $U_{w \times h}$. The azimuth angle θ is obtained by rescaling the x coordinate, from the range $[0, w]$ to $[0, 2\pi]$:

$$\theta = \frac{2\pi x}{w}. \tag{4}$$

To define the coordinate z, we assume that the region depicted in U lies between two boundary elevation angles φ_{min} and φ_{max}, such that $0 \leq \varphi_{min} < \varphi_{max} < \pi/2$, and thus:

$$z = \tan \varphi_{min} + \frac{y}{h}(\tan \varphi_{max} - \tan \varphi_{min}). \tag{5}$$

Moving from the cylindrical system to the spherical one is straightforward, since they share the same azimuth value θ, and the elevation angle is defined as:

$$\varphi = \arctan z. \tag{6}$$

For the final step, let us consider the polar coordinate system in the hemispheric image, as shown in Figure 5. Here, the azimuth angle θ is the same of the spherical and cylindrical systems, while the definition of $\rho \in [0, 1]$ depends on the camera architecture. In the vast majority of hemispheric cameras, there is a linear correspondence between ρ and φ, however it is possible that some optics acquire distorted

images to give higher resolution to specific image regions. Assuming that the linear correspondence hypothesis holds, it follows that:

$$\rho = 1 - \frac{2\varphi}{\pi}. \tag{7}$$

The polar coordinates (θ, ρ) can be transformed to Cartesian coordinates in the $[-1, 1] \times [-1, 1]$ range:

$$\begin{aligned} u &= \rho \cos\theta \\ v &= \rho \sin\theta \end{aligned} \tag{8}$$

which can be converted to real pixel coordinates:

$$\begin{aligned} x' &= uR + x_c \\ y' &= vR + y_c \end{aligned} \tag{9}$$

where R, x_c, y_c are the parameters obtained by camera calibration with equations (2) and (3).

C. Off-center case

The equations given in section III-B work for the ideal case of a camera placed at the center of the pipe. In practical applications, this requirement could not be satisfied, especially if the pipe is partially filled with liquid. We thus provide corrective equations to compensate for the off-center displacement of the camera, which give the true (θ_r, φ_r) camera spherical coordinates to observe the same point that would be observed by an ideal camera, placed at pipe center, at coordinates (θ_i, φ_i). The equations are defined as:

$$\theta_r = \arctan\left(\frac{A\sin\beta - \sin\theta_i}{A\cos\beta - \cos\theta_i}\right) \tag{10}$$

$$\varphi_r = \arctan\left(\tan\varphi_i \frac{\sin(\beta - \theta_r)}{\sin(\beta - \theta_i)}\right). \tag{11}$$

Full proof and definition for A and β are given in Appendix.

The unfolding procedure in the off-center case is thus computed by the following steps:

1) Use eq. (4)–(6) to get the ideal spherical coordinates (θ_i, φ_i);
2) Use eq. (10), (11) to get the real spherical coordinates (θ_r, φ_r);
3) Use eq. (7)–(9) to get the pixel coordinates (x', y').

IV. Mosaicking

The second module of the proposed system is the image mosaicking block. It acquires the unfolded video frames coming from the previous module and provides a comprehensive planar view of the whole internal surface of the pipe, as shown in Figure 6. Mosaicking can be considered a main prerequisite of many systems dealing with the automatic analysis of planar video sequences. In fact, on one side, it allows us to obtain a single image representing the entire area of interest, thus facilitating the application of proper analysis algorithms. On the other side, the use of a mosaic prevents multiple processing of overlapping areas in adjacent frames.

In this paper, the implementation of the mosaicking is based on the architecture presented in [29]. The following sections describe how unfolded images can be used to build the mosaic.

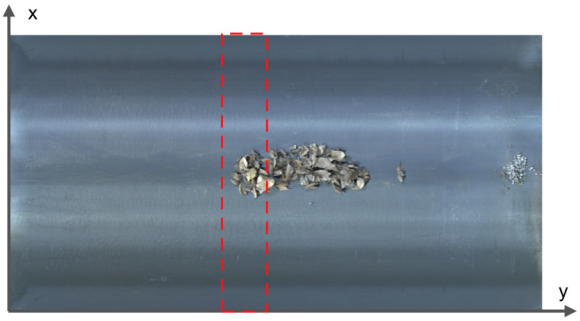


Fig. 6. A mosaic of the internal surface of a pipe. The dashed rectangle highlights one of the unfolded images used to build the mosaic.

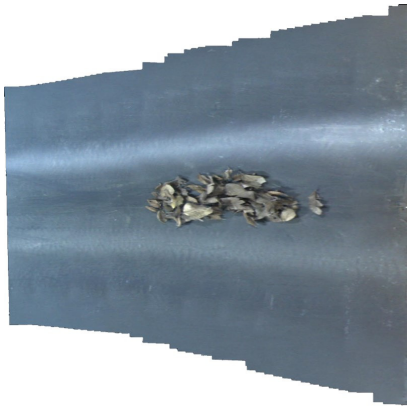


Fig. 7. Mosaic for the same sequence of Figure 6, created with Microsoft Image Composite Editor.

A. Feature extraction and matching

Without loss of generality, let M_t be the mosaic built up to time t, and let U_{t+1} represent the next image, at time $t+1$, that must be added to it. The main stages to build the new mosaic, $M_t \cup U_{t+1}$, are the feature extraction and matching processes. Notice that initially the mosaic is composed of the first unfolded image acquired at time $t = 1$, i.e., $M_1 \equiv U_1$.

During the first step, a set of local image features (i.e. keypoints) is extracted from both M_t and U_{t+1}. The set provides a limited collection of well-localised anchor points, which can be consistently identified even in presence of illumination changes or noise. In the second step, the keypoints are used to compute the overlapping region between M_t and U_{t+1}. The latter is a crucial aspect for mosaicking process, since only images with a certain degree of overlapping can be stitched together to form a mosaic.

Following the results presented in [30], the proposed mosaicking approach uses the SURF [31] algorithm to extract the features and the Brute Force Matcher (BFM) algorithm [32] to perform the matching process. Formally, the two sets of features extracted from M_t and U_{t+1} are defined as $\mathcal{X}_{M_t} = \{\alpha_1, \ldots, \alpha_h\}$ and $\mathcal{X}_{U_{t+1}} = \{\beta_1, \ldots, \beta_s\}$, where $h, s \in \mathbb{N}$. The BFM algorithm generates two sub-sets $\hat{\mathcal{X}}_{M_t} = \{\alpha_{h_1}, \ldots, \alpha_{h_m}\} \subseteq \mathcal{X}_{M_t}$ and $\hat{\mathcal{X}}_{U_{t+1}} = \{\beta_{s_1}, \ldots, \beta_{s_m}\} \subseteq \mathcal{X}_{U_{t+1}}$, where for each $k \in \{h_1, \ldots, h_m\}$ a single $j \in \{s_1, \ldots, s_m\}$ exists such that α_k matches β_j . The two sub-

sets have the same cardinality.

Feature matching performance could be degraded by visually uniform surfaces, where no anchor points can be robustly detected. In order to mitigate this effect, a contrast-enhancing preprocessing step is applied. It works by mapping the intensity values of each image to new values such that, by default, 1% of the data is saturated at low and high intensities of the input data (inspired by MATLAB function `imadjust`). Moreover, the SURF feature detector has a tunable parameter (the Hessian threshold) which controls the sensitivity of feature detection process. This parameter is dynamically adapted so that the number of detected features is always more than 200 and less than 400. Despite the preprocessing being just a heuristic (no features can be detected if the pipe surface is perfectly uniform), it worked extremely well with all the real pipes it was tested on, even if the pipe was never used before such as the one shown in Figure 6 (before the insertion of debris).

B. Homography and stitching

Once the corresponding features $\hat{\mathcal{X}}_{M_t}$ and $\hat{\mathcal{X}}_{U_{t+1}}$ are computed, the mosaicking approach evaluates the geometrical transformation by which the features of the current image U_{t+1} are fitted with those of the present mosaic, M_t, within the reference system of the latter. The transformation is subsequently applied to each pixel of the image to stitch it over the mosaic. The standard practice to perform this task is through a homographic matrix H whose elements are defined as follows [29], [30]:

$$H = \begin{bmatrix} R_a & R_b & T_x \\ R_c & R_d & T_y \\ W_a & W_b & 1 \end{bmatrix} \tag{12}$$

where $H_R = \begin{bmatrix} R_a & R_b \\ R_c & R_d \end{bmatrix}$ is the *Rotation Matrix*, $M_T = \begin{bmatrix} T_x & T_y \end{bmatrix}^T$ is the *Translation Vector*, and $M_W = \begin{bmatrix} W_a & W_b \end{bmatrix}$ is the *Warping Vector*. However, in the proposed case, the warping vector is null, since unwarping has already been performed by the image unfolding module. Moreover, in section I, we assumed that the camera optical axis coincides with the pipe central axis, thus no rotations are expected. The problem is thus reduced to the estimation of the translation part only, where T_y is due to the camera moving along the pipe and T_x is due to camera roll, if present.

The problem can theoretically be solved by finding just one correspondence between $\hat{\mathcal{X}}_{M_t}$ and $\hat{\mathcal{X}}_{U_{t+1}}$, however the results could be unreliable because of noise or matching errors. For this reason, after the feature matching process, a step to distinguish the *inlier* points (i.e., keypoints correctly matched) from the *outlier* ones (i.e., keypoints wrongly matched) is necessary. The RANSAC algorithm is one of the most used methods for distinguishing these points, it provides robust estimations even with a high number of outliers [29], [30]. Once the outliers are removed, the translation vector is estimated as the mean of all the translations between matching features, and a new mosaic M_{t+1} is obtained.

The entire mosaicking is thus explicitly tuned for the special case of pipe internals, especially because of the constraints on the expected motion. As a proof of the goodness of the proposed approach, compare the results from the proposed system (Figure 6) with one obtained with Microsoft Image Composite Editor (Figure 7), a state-of-the-art general-purpose mosaicking software based on DAISY image features [33] and Interactive Digital Photomontage algorithm [34]. Despite ICE was fed with several hints on the nature of the mosaic (planar motion, relative image positions, estimated overlap) the absence of specific constraints led to an incorrect result.

V. Anomaly detection

Explicit defect recognition is a complex task, because of the large number of possible defects and their wide intra-class variance. In the proposed work, rather than explicitly classifying image regions as defects, we adopt an anomaly detection approach: image regions are given an anomaly score based on their dissimilarity from surrounding areas, thus highlighting the parts which do not look like the "regular" pipe internal surface.

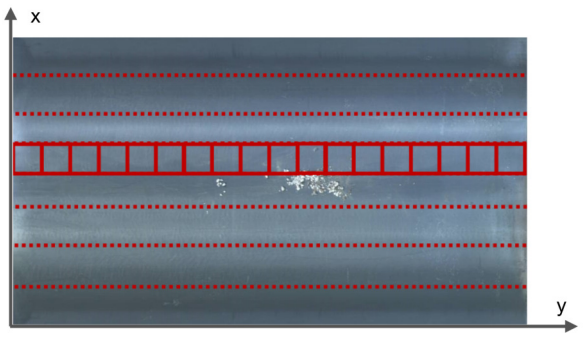


Fig. 8. Mosaic images are divided in stripes along the y axis. Anomaly detection is performed independently on each stripe by cutting it in patches where the LBP feature descriptor is computed.

Anomaly detection is thus based on visual similarity. Within a pipe section, no similarity is expected between different radial sectors (e.g. the top surface will probably be very different from the bottom surface, where water flows). Mosaic images are thus divided in stripes along the direction of the pipe symmetry axis, and each stripe is analyzed independently from the other ones.

In order to measure visual similarity, each stripe is cut in patches as shown in Figure 8, and on each patch a Local Binary Pattern (LBP) texture feature is computed [35]. LBP are visual features that describe the appearance of an image patch based on local intensity differences, and have been widely adopted in many application fields [36], [37].

The proposed approach to automatically detect image patches with anomalous LBP descriptors is inspired by one-class Support Vector Machines (SVM) theory [38] and it is a variant of the method described in [39], [40]. Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a set of n LBP descriptors, so that each $x_i \in \mathbb{R}^m$ is a vector with pre-defined, fixed size. We define a kernel k such that:

$$k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{13}$$

where $\Phi : \mathcal{X} \mapsto \mathcal{H}$ is a function mapping the descriptors to a new feature space $\mathcal{H}$. According to kernel methods theory, there is no need to explicitly know Φ, as any function k satisfying the Mercer's theorem is a dot product in some feature space [41]. We adopted the popular Radial Basis Function kernel, defined as:

$$k(x_i, x_j) = \exp\left(\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{14}$$

which is a normalized kernel since $k(x, x) = 1 \ \forall x \in \mathcal{X}$. It follows that $\|\Phi(x)\|^2 = \Phi(x) \cdot \Phi(x) = k(x, x) = 1$, thus all the vectors $\Phi(x_i)$ lie on the surface of an hypersphere with radius 1 in $\mathcal{H}$.

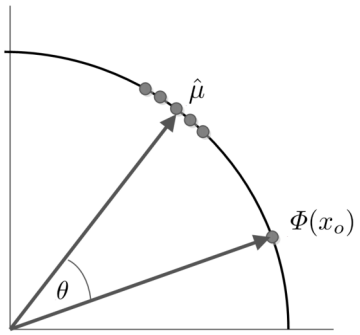


Fig. 9. The angle between an element $\Phi(x_o)$ and the normalized data mean $\hat{\mu}$ can be interpreted as an anomaly measure.

One-class SVMs identify outliers by proving that the "normal" data all lie on a hyperspherical cap in $\mathcal{H}$, and use proper optimization techniques to identify the best hyperplane that separates this cap from the rest of the hypersphere where anomalous data (the outliers) reside [38]. However, this relies on a tunable parameter (e.g. ν in [38]) dependent on the number of expected outliers, which is generally unknown. We thus propose a technique to automatically identify outliers without knowing their amount. Let μ be the mean of all the data projected in $\mathcal{H}$:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \Phi(x_i) \tag{15}$$

and $\hat{\mu}$ be the mean normalized to unitary length:

$$\hat{\mu} = \frac{\mu}{\|\mu\|} = \frac{\sum_{i=1}^{n} \Phi(x_i)}{\|\sum_{i=1}^{n} \Phi(x_i)\|}. \tag{16}$$

Given a generic candidate outlier $\Phi(x_o)$, we can define its dot product with $\hat{\mu}$ as:

$$\hat{\mu} \cdot \Phi(x_o) = \frac{\sum_{i=1}^{n} \Phi(x_i) \cdot \Phi(x_o)}{\|\sum_{i=1}^{n} \Phi(x_i)\|} \tag{17}$$

$$= \frac{\sum_{i=1}^{n} k(x_i, x_o)}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} k(x_i, x_o)}}. \tag{18}$$

The dot product of two unitary vectors is the cosine of the angle between the two vectors, thus we can compute the angle θ_o between a generic element $\Phi(x_o)$ and the normalized data mean $\hat{\mu}$ as:

$$\theta_o = \arccos\left(\frac{\sum_{i=1}^{n} k(x_i, x_o)}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} k(x_i, x_o)}}\right). \tag{19}$$

In their seminal paper on one-class ν-SVM [38], Schölkopf et al. prove that the non-outlier data reside in a spherical cap whose center converges to $\hat{\mu}$ as $\nu \to 1$. It is thus safe to assume that the non-outlier data are close to each other around the data mean, thus having a small θ_o, while outliers will lie far from it. In other words, the angle θ_o can be interpreted as a measure of the anomaly degree of a given element (Figure 9). This consideration motivates the following procedure to identify outliers.

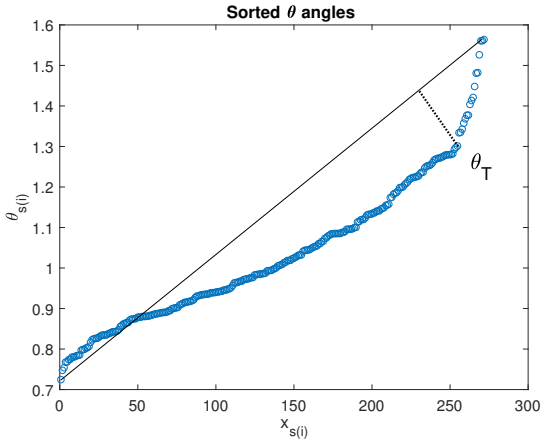


Fig. 10. Plot of Θ_s and the graphical identification of the anomaly threshold θ_T.

Let s be a permutation of $[1 \dots n]$ such that $\Theta_s = \{\theta_{s(1)} \dots \theta_{s(n)}\}$ is the list of all the angles θ computed with eq. (19) and sorted by ascending values. We expect that values in Θ_s will initially have a slow, constant increasing rate, corresponding to non-outlier data which all lie close to the data mean, while the last elements will have an abrupt increment due to outliers lying far from the majority of the data. This motivates a graphical way to identify the point separating the two trends, defined as the farthest point in the plot of Θ_s from the line merging the first and last element, as shown in Figure 10. This way is possible to automatically define a threshold value θ_T such that any element x_i whose corresponding θ_i is larger than θ_T is considered an outlier. This technique does not require any initial estimation on the amount of expected outliers, in contrast with traditional one-class SVM classification tools.

By applying the proposed approach to LBP descriptors of image patches within each image stripe, visually-anomalous areas can be detected, and an anomaly score can be assigned to each portion of the image. An anomaly map can thus be built, helping the human operator to focus only on potentially relevant areas and thus reducing the inspection time.

VI. Experimental results

In order to evaluate the performances of the proposed system, we used an Axis M3027-PVE hemispheric color

camera. The image resolution is 2592x1944 pixel (including the lateral black regions, as in Figure 4) at 12 frames per second. The camera was mounted in a waterproof enclosure, also containing lights and a battery pack providing the required power via PoE interface. Acquired video sequences are stored on an internal microSD card, thus the entire system is totally autonomous.

The calibration procedure found the circular image center at position $(1261, 960)$ with a radius $R = 1041$ pixels. It is worth noting that the computed center is 37 pixels far from the geometric one at $(1296, 972)$, thus justifying the need for a calibration routine. The angular boundaries described in section III are set to $\varphi_{min} = 0.124$ rad, $\varphi_{max} = \pi/6$ rad. In particular, φ_{min} has been chosen in order to avoid the unfolding procedure to process non-existent image areas: the camera in fact cuts off a small portion of the top and bottom parts of the circular image, as it can be seen in Figure 4.

The unfolded image size, used in equations (4) and (5), is in principle arbitrary. However, the maximum sensible width is $w = 2\pi R'$ pixels, where R' is the image radius at φ_{min} elevation, computed as $R' = (1 - 2\varphi_{min}/\pi)R$. Larger values would be useless, since unfolded images would have a resolution higher than any portion of the hemispheric image. To compute the height, we consider the unitary cylinder shown in Figure 5: here, the height of the unfolded area is $\tan(\varphi_{max}) - \tan(\varphi_{min})$, while the cylinder perimeter is 2π. This allows to find the correct image ratio and thus image height is set as $h = w\frac{\tan(\varphi_{max})-\tan(\varphi_{min})}{2\pi}$. According to these considerations, the unfolded image size is set to 6024x434 pixels. In the following experiments, however, we scaled the unfolded images by a factor of 2 in both width and height to improve the processing speed: experimental results showed that such a rescaling factor does not lead to noticeable performance degradation, while reducing the off-line processing time by 4 times.

In the mosaic building phase, the feature detector is automatically tuned to extract at least 200 features. Empirical experiments showed that 200 features are enough to achieve a correct alignment. Finally, in anomaly detection, we fixed the cell size where to compute LBP features to 16x16, thus obtaining 188 independent stripes.

A. Laboratory results with a test pipe

Figure 11 shows some of the laboratory results obtained on a short pipe segment, with radius 20 cm and length 500 cm, which was manually filled with different types of waste. The actual tests considered 10 different scenarios, with the original video lengths varying from 32 s to 63 s. The left column shows the obtained mosaics, each one is 3012 pixel wide and roughly 5000 pixel high (exact mosaic height depends on the start and stop positions where the sequence has been acquired). The mosaics do not have stitching glitches and correctly depict the pipe internals. The right column shows the anomaly maps after a median filtering, where blue represents normality, while other colors show different degrees of anomaly, with bright yellow being the most anomalous areas. As it can be seen, the areas with waste were correctly detected as different from the remaining pipe appearance. In order to give a quantitative measure of the results, ground truth masks were manually created for each mosaic, highlighting the areas with waste. This mask was compared with the detected anomalous areas in order to compute the precision and recall of the achieved results. The average precision over the 10 test sequences is 0.54, while the average recall is 0.93. The proposed algorithm is thus efficient in detecting the real image anomalies, although it also identifies some false positives. This behavior is acceptable if we consider that the system aim is not to perform a fully automated processing, but to highlight potentially relevant image areas to a human operator for further evaluation and classification.

B. Results in a real waterworks pipeline

The system was tested also on real-world scenarios, in particular regarding three different public waterworks pipelines, here called pipeline 1, 2, and 3, since full details as the exact location or the public entity in charge of them cannot be disclosed. Table I shows detailed data about the pipelines. All scenarios involved concrete pipes with large diameters, from 2325 to 3100 mm, partially filled with water. Inspection of course can be applied only to the off-water internal pipe surface.

In order to use the system in partially filled pipelines, the hardware structure (camera, batteries, lights) was mounted over a floating platform. The platform is moved by the water flow and the speed is controlled using a retention cable. The water flow naturally keeps the camera optical axis parallel with the pipe axis, however the platform can oscillate sideways. In order to avoid this horizontal drift, the platform is kept roughly at the center of the pipe using side spacers. The vertical offset respect to the pipe center, due to water filling the pipes, is compensated using the off-center unfolding equations described in section III-C by settings camera offset (x_c, y_c) to $(0, -0.2)$, $(0, 0.2)$ and $(0, 0.2)$ for pipelines 1, 2, and 3 respectively. With the described setup, several video sequences, listed in table II, were acquired, covering segments from 78 to 1134 meters long. In order to test the system under different conditions, sequences have been acquired at different speeds, for example sequence 3.2 covers only 684 meters, much less than sequence 3.1 (1134 meters) despite the temporal length is almost doubled.

TABLE I
INSPECTED WATERWORKS PIPELINES

Pipeline	Material	Diameter	Fill
1	Concrete	2600 mm	Water 60%
2	Concrete	3100 mm	Water 40%
3	Concrete	2325 mm	Water 40%

The acquired video sequences were processed to create mosaics such as the one shown in Figure 12. In order to limit the memory footprint of the system, each mosaic has a maximum height of 10,000 pixels, thus each sequence is split in several mosaics. In total, 50 mosaics were built, each one with size 2847×10000 pixels, depicting roughly 50 meters of pipeline each. The mosaicking procedure never failed since

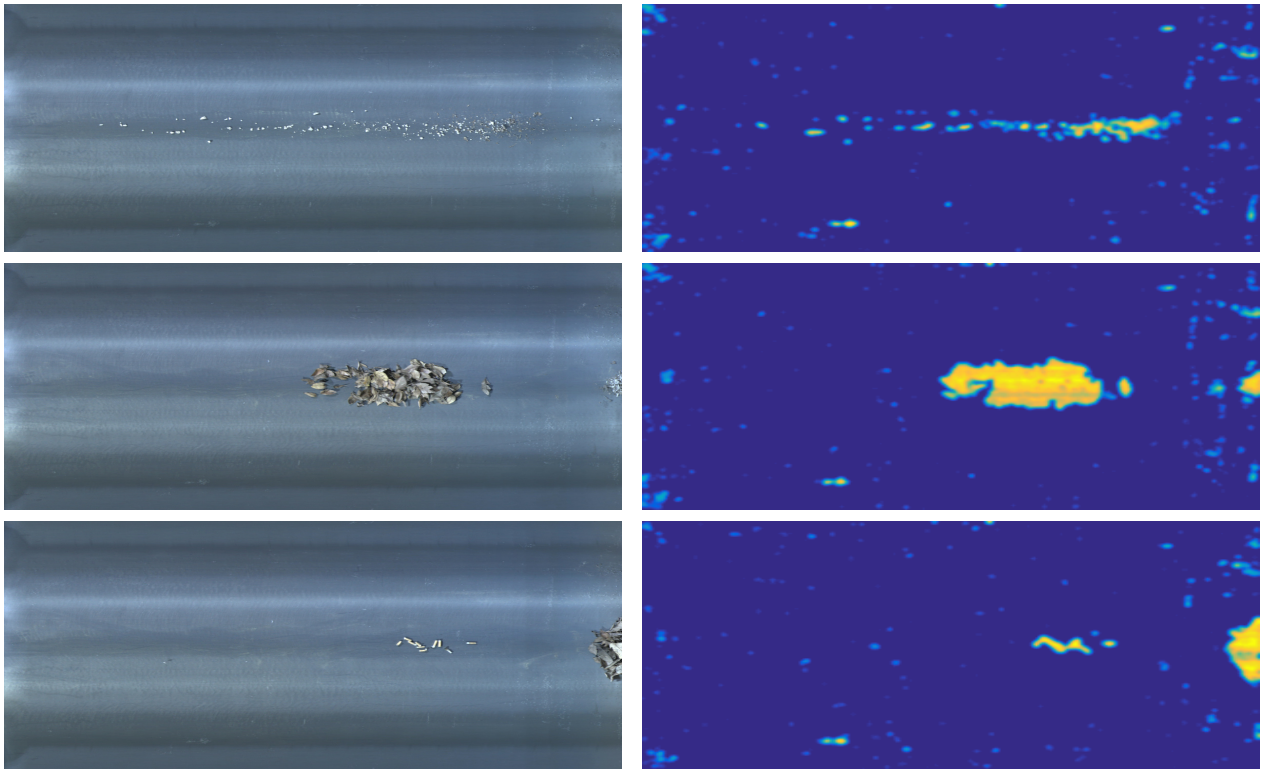


Fig. 11. Experimental results with pebbles (first row), leaves (second row), and cigarette butts (third row). Left column: mosaics of the pipe internal surface. Right column: anomaly maps, yellow denotes detected anomalies.

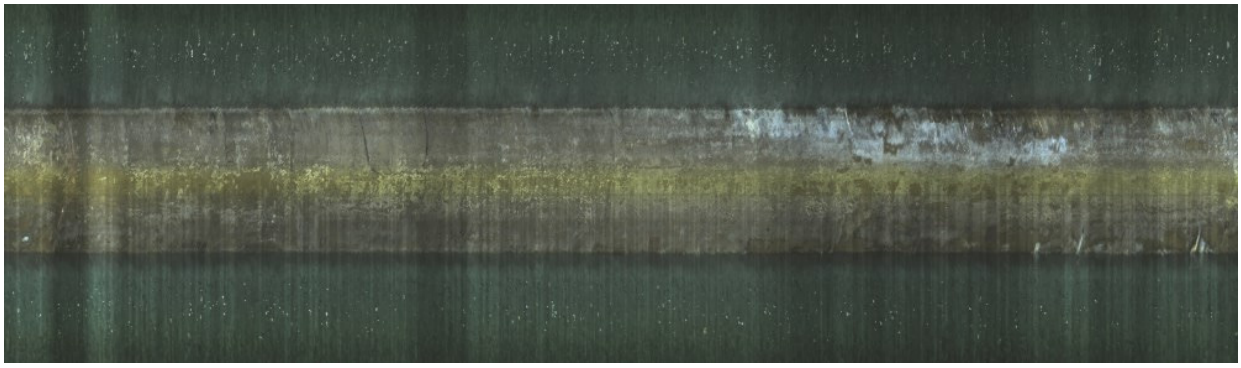


Fig. 12. One of the mosaics obtained from sequence 1.1. Unfolding is done in order to keep the visible pipe surface in the central part of the mosaic. Vertical darker bands are due to a defective light; they do not have an excessive impact on the system thanks to LBP texture features being invariant for intensity changes.

TABLE II
ACQUIRED VIDEO SEQUENCES

Sequence	Pipeline	Length (time)	Length (meters)
1.1	1	32m 24s	405
2.1	2	07m 34s	78.5
2.2	2	07m 17s	78.9
3.1	3	1h 12m 08s	1134
3.2	3	2h 22m 25s	683.9
3.3	3	19m 46s	327.6

the pipe surface has enough visual features to guarantee an optimal feature extraction and matching. The mosaics were then processed by the anomaly detection module, in order to identify visually anomalous areas, such as the one shown in Figure 13. There, an anomalous leak of external water into the pipe left a visually evident white stripe on the pipe surface. The Figure shows how the system clearly detected the leak as a visual anomaly, thus allowing the human operator to perform further investigation.

Anomaly detection performance was measured by comparing the results with the output of a traditional human-based visual inspection. Rather than creating pixel-wise ground truth masks as in section VI-A we just filtered out the anomaly areas smaller than a given threshold (here fixed to 1000 pixels), and counted the remaining ones. The system detected in total 121 anomalies over all the test sequences, while the human operator identified 17 real anomalies, see full details reported in table III. The overall recall of the system is 0.88, meaning

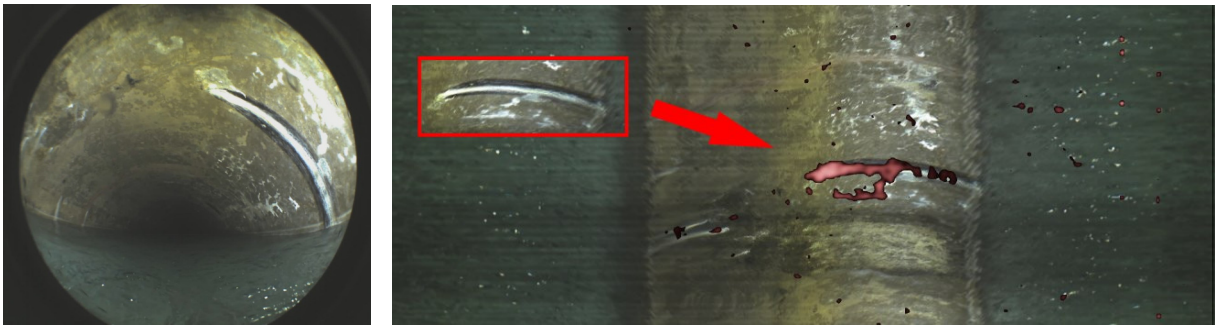


Fig. 13. Left: visual appearance of an anomalous leak, as seen from the camera. Right: unfolded view, with highlighted anomalous areas detected by the system. The actual leak, covered by the anomaly map, is shown in the red box.

TABLE III
SYSTEM PERFORMANCES. SECOND AND THIRD COLUMNS REPORT THE NUMBER OF ANOMALIES DETECTED BY A HUMAN OPERATOR AND THE PROPOSED SYSTEM RESPECTIVELY. THE REMAINING COLUMNS SHOW THE NUMBER OF TRUE POSITIVES, FALSE NEGATIVES AND FALSE POSITIVES, WITH THE CORRESPONDING PRECISION (TP/(TP+FP)) AND RECALL (TP/(TP+FN)).

Seq.	Real	Sys	TP	FN	FP	Prec	Rec
1.1	3	19	3	0	16	0.15	1
2.1	1	5	1	0	4	0.20	1
2.2	0	3	0	0	3	0	n.a.
3.1	6	49	5	1	44	0.10	0.83
3.2	4	28	3	1	25	0.10	0.75
3.3	3	17	3	0	14	0.17	1
Total	17	121	15	2	106	0.12	0.88

that the majority of real anomalies were correctly detected. The precision is 0.12, meaning that several detections were actually false positives. This result is worse than the one obtained in laboratory tests, mostly because a brand new pipe has a much higher textural uniformity than a real one, and thus detecting anomalies is easier. The high recall is anyway a positive result, since it implies that the human operator can focus only on the detected areas to search for real anomalies. Considering that, in each analyzed sequence, the total area of detected anomalies is approximately 1% of the total mosaics area, the human effort is greatly reduced. In fact a second human operator, focusing only on the detected areas, managed to get the same results of the first operator in 1/10 of the time.

Regarding the computational requirements, the system was implemented in C++ with OpenCV 3.1 libraries and tested on a PC mounting an Intel i7-4790 CPU @ 3.60 GHz and 8 GB RAM. On average, the process of image acquisition, unfolding and mosaicking required 94 ms per frame. Considering that the camera acquires at 12 fps, this means that the video processing up to mosaicking can almost be done in real-time, provided that a computational unit is embedded in the system. Off-line anomaly detection required on average 19.9 s to process a 2847×10000 mosaic with 16×16 LBP cell size.

VII. CONCLUSIONS

In this work, we presented a novel technique to support human operators in the task of visual pipeline inspection. The system relies on video sequences acquired by a hemispheric camera in order to build unfolded views of the pipe's internal surface. The views are then processed with a kernel-based anomaly detector to highlight areas which are visually different from the surroundings, and thus possibly denoting a defect that should be reported to the human operator. The achieved results are promising and show that the presence of anomalies inside the pipe can be reliably detected. As a future development, we plan to augment the system capabilities with laser and sonar scanners. The new sensors could help both in detecting the true position of the camera inside the pipe, thus improving the visual processing, and in finding shape or size anomalies, e.g. due to corrosion, which could be not visually detectable.

ACKNOWLEDGMENTS

[Currently omitted to guarantee paper anonymity]

APPENDIX

A. Derivation of eqs. 10 and 11

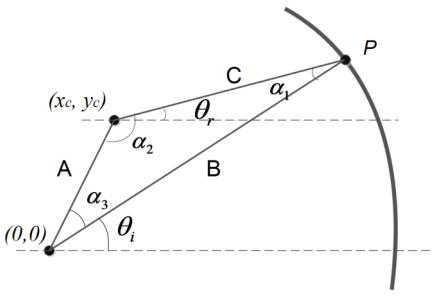


Fig. 14. Pipe section showing the pipe center at $(0, 0)$ and the camera at (x_c, y_c). The goal is to find the real azimuth angle θ_r given the ideal angle θ_i to observe point P.

Figure 14 shows a pipe section, where the pipe center is located at the origin and camera is positioned at (x_c, y_c). The pipe radius can be arbitrarily assumed to be $B = 1$. The camera displacement is defined as $A = \sqrt{x_c^2 + y_c^2}$. We also define $\beta = \theta_1 + \alpha_3 = \arctan(y_c/x_c)$. Basic geometric considerations allow to find $\alpha_2 = \pi - \beta + \theta_r$ and $\alpha_1 = \pi - \alpha_2 - \alpha_3 = \theta_i - \theta_r$.

According to the law of sines, we have:

$$\frac{A}{\sin \alpha_1} = \frac{1}{\sin \alpha_2} \tag{20}$$

and thus:

$$A = \frac{\sin\alpha_1}{\sin\alpha_2} \tag{21}$$

$$= \frac{\sin(\theta_i - \theta_r)}{\sin(\beta - \theta_r)} \tag{22}$$

$$= \frac{\sin\theta_i\cos\theta_r - \cos\theta_i\sin\theta_r}{\sin\beta\cos\theta_r - \cos\beta\sin\theta_r}. \tag{23}$$

By rearranging terms we get

$$\sin\theta_r(A\cos\beta - \cos\theta_i) = \cos\theta_r(A\sin\beta - \sin\theta_i) \tag{24}$$

and thus

$$\tan\theta_r = \frac{A\sin\beta - \sin\theta_i}{A\cos\beta - \cos\theta_i} \tag{25}$$

which leads to eq. (10).

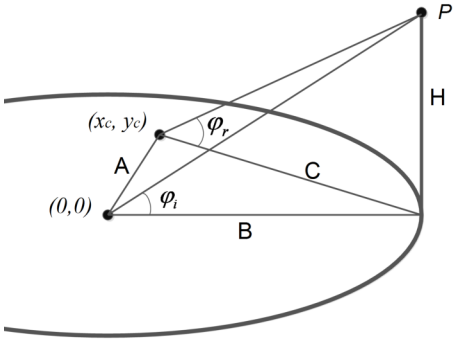


Fig. 15. Same configuration of Figure 14, but highlighting the distance H of point P from the camera plane. The goal is to find the real elevation angle φ_r given the ideal angle φ_i.

In order to find the equation linking ideal and real elevation angles, let us consider Figure 15. By using again the law of sines we get:

$$\frac{B}{C} = \frac{\sin\alpha_2}{\sin\alpha_3} \tag{26}$$

$$= \frac{\sin(\beta - \theta_r)}{\sin(\beta - \theta_i)}. \tag{27}$$

Basic trigonometric considerations give that:

$$\tan\varphi_r = \frac{H}{C} = \tan\varphi_i\frac{B}{C} \tag{28}$$

$$= \tan\varphi_i\frac{\sin(\beta - \theta_r)}{\sin(\beta - \theta_i)} \tag{29}$$

which leads to eq. (11).

REFERENCES

[1] M. Gasperetti, "Così il lungarno è sprofondato – paura a Firenze," *Corriere della Sera*, no. 26 May, 2016.

[2] Y. Zhang and G. Yan, "In-pipe inspection robot with active pipe-diameter adaptability and automatic tractive force adjusting," *Mechanism and Machine Theory*, vol. 42, no. 12, pp. 1618–1631, 2007.

[3] Y.-S. Kwon and B.-J. Yi, "Design and motion planning of a two-module collaborative indoor pipeline inspection robot," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 681–696, 2012.

[4] C. Walter, J. Saenz, N. Elkmann, H. Althoff, S. Kutzner, and T. Stuerze, "Design considerations of robotic system for cleaning and inspection of large-diameter sewers," *Journal of Field Robotics*, vol. 29, no. 1, pp. 186–214, 2012.

[5] Z. Liu and Y. Kleiner, "State of the art review of inspection technologies for condition assessment of water pipes," *Measurement*, vol. 46, no. 1, pp. 1 – 15, 2013.

[6] Q. Feng, R. Li, B. Nie, S. Liu, L. Zhao, and H. Zhang, "Literature review: Theory and application of in-line inspection technologies for oil and gas pipeline girth weld defection," *Sensors*, vol. 17, no. 1, p. 50, 2017.

[7] H. Choi and S. Ryew, "Robotic system with active steering capability for internal inspection of urban gas pipelines," *Mechatronics*, vol. 12, no. 5, pp. 713 – 736, 2002.

[8] N. Avdelidis, E. Delegou, D. Almond, and A. Moropoulou, "Surface roughness evaluation of marble by 3d laser profilometry and pulsed thermography," *NDT & E International*, vol. 37, no. 7, pp. 571 – 575, 2004.

[9] O. Duran, K. Althoefer, and L. D. Seneviratne, "State of the art in sensor technologies for sewer inspection," *IEEE Sensors Journal*, vol. 2, no. 2, pp. 73–81, 2002.

[10] Y. Shi, C. Zhang, R. Li, M. Cai, and G. Jia, "Theory and application of magnetic flux leakage pipeline detection," *Sensors*, vol. 15, no. 12, pp. 31 036–31 055, 2015.

[11] M. R. G. Hazelden, G. Ragula, "The use of broadband electromagnetic technology for integrity inspection of a 760 mm (30 in.) cast iron and steel line," in *Proceedings of the 22nd World Gas Conference, Tokyo, Japan*, 2003.

[12] D. Sack and L. Olson, "Impact echo testing of in situ precast concrete cylinder pipe," in *Proceedings of the 1998 Pipeline Division Conference: Pipelines in the Constructed Environment, American Society of Civil Engineers, San Diego, CA, USA*, 1998, pp. 250–259.

[13] M. Cong, X. Wu, and C. Qian, "A longitudinal mode electromagnetic acoustic transducer (emat) based on a permanent magnet chain for pipe inspection," *Sensors*, vol. 16, no. 5, p. 740, 2016.

[14] M. Siqueira, C. Gatts, R. da Silva, and J. Rebello, "The use of ultrasonic guided waves and wavelets analysis in pipe inspection," *Ultrasonics*, vol. 41, no. 10, pp. 785 – 797, 2004.

[15] J. L. Rose, J. Mu, and Y. Cho, "Recent advances on guided waves in pipe inspection," in *Proceedings of the 17th World Conference on Non-Destructive Testing, Shanghai, China*, 2008, pp. 25–28.

[16] J. Bosch, A. Hugger, J. Franz, S. Falter, and Y. Oberdorfer, "Phase array technology for automated pipeline inspection," *Business Briefing: Exploration & Production: The Oil & Gas Review*, pp. 1–4, 2004.

[17] Y. Kleiner, B. Rajani, and R. Sadiq, "Fuzzy-based method to evaluate soil corrosivity for prediction of water main deterioration," *Journal of Infrastructure Systems*, vol. 10, no. 4, pp. 149–156, 2004.

[18] H. Najjaran, R. Sadiq, and B. Rajani, "Fuzzy expert system to assess corrosion of cast/ductile iron pipes from backfill properties," *Computer-Aided Civil and Infrastructure Engineering*, vol. 21, no. 1, pp. 67–77, 2006.

[19] H. Najjaran, B. Rajani, R. Sadiq, and Z. Liu, "Exploring the relationship between soil properties and deterioration of metallic pipes using predictive data mining methods," *Journal of Computing in Civil Engineering*, vol. 24, no. 3, pp. 289–301, 2010.

[20] P. Huynh, R. Ross, A. Martchenko, and J. Devlin, "Anomaly inspection in sewer pipes using stereo vision," in *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2015, pp. 60–64.

[21] D. Krys and H. Najjaran, "Development of visual simultaneous localization and mapping (vslam) for a pipe inspection robot," in *2007 International Symposium on Computational Intelligence in Robotics and Automation*, 2007, pp. 344–349.

[22] H. H. Aghdam, H. A. Kadir, M. R. Arshad, and M. Zaman, "Localizing pipe inspection robot using visual odometry," in *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, 2014, pp. 245–250.

[23] P. Hansen, H. Alismail, P. Rander, and B. Browning, "Visual mapping for natural gas pipe inspection," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 532–558, 2015.

[24] T. Wu, S. Lu, and Y. Tang, "An in-pipe internal defects inspection system based on the active stereo omnidirectional vision sensor," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015, pp. 2637–2641.

[25] D. Avola, G. L. Foresti, L. Cinque, C. Massaroni, G. Vitale, and L. Lombardi, "A multipurpose autonomous robot for target recognition in unknown environments," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 766–771.

[26] W. Wu, Z. Liu, and Y. He, "Classification of defects with ensemble methods in the automated visual inspection of sewer pipes," *Pattern Analysis and Applications*, vol. 18, no. 2, pp. 263–276, 2015.

[27] J. Mashford, P. Davis, and M. Rahilly, "Pixel-based colour image segmentation using support vector machine for automatic pipe inspection," in *AI 2007: Advances in Artificial Intelligence: 20th Australian Joint Conference on Artificial Intelligence, Gold Coast, Australia, December 2-6, 2007. Proceedings*, M. A. Orgun and J. Thornton, Eds., 2007, pp. 739–743.

[28] Y. Zhang, L. Zhao, and W. Hu, "A survey of catadioptric omnidirectional camera calibration," *International Journal of Information Technology and Computer Science*, vol. 5, no. 3, p. 13, 2013.

[29] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vision*, vol. 74, no. 1, pp. 59–73, 2007.

[30] C. Piciarelli, C. Micheloni, N. Martinel, M. Vernier, and G. L. Foresti, "Outdoor environment monitoring with unmanned aerial vehicles," in *Proceedings of the 17th International Conference on Image Analysis and Processing*, 2013, pp. 279–287.

[31] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vis. and Image Unders.*, vol. 110, no. 3, pp. 346–359, 2008.

[32] B. C. Song, M. J. Kim, and J. B. Ra, "A fast multiresolution feature matching algorithm for exhaustive search in large image databases," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 5, pp. 673–678, 2001.

[33] S. Winder, G. Hua, and M. Brown, "Picking the best daisy," in *Computer Vision and Pattern Recognition*. IEEE Computer Society, June 2009.

[34] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 294–302, 2004.

[35] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[36] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, *Computer Vision Using Local Binary Patterns*. Springer, 2011.

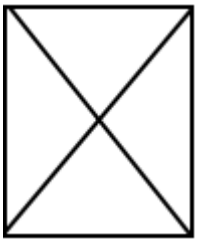
[37] D. Avola, G. L. Foresti, N. Martinel, C. Micheloni, D. Pannone, and C. Piciarelli, "Aerial video surveillance system for small-scale uav environment monitoring," in *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[38] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt *et al.*, "Support vector method for novelty detection." in *Neural Information Processing Systems*, vol. 12, 1999, pp. 582–588.

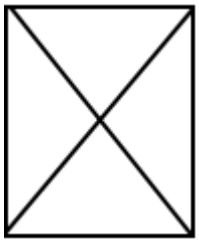
[39] C. Piciarelli and G. Foresti, "Anomalous trajectory detection using support vector machines," in *Advanced Video and Signal Based Surveillance (AVSS), 2007 IEEE International Conference on*, 2007, pp. 153–158.

[40] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008.

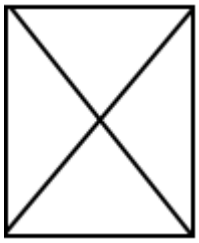
[41] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.



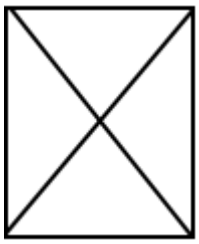
Author 1 Biography text here.



Author 2 Biography text here.



Author 3 Biography text here.



Author 4 Biography text here.