A single camera inspection system to detect and localize obstacles on railways based on manifold Kalman filtering

Federica Fioretti
TeCIP Institute
Scuola Superiore Sant'Anna
Pisa, Italy
federica.fioretti.90@gmail.com

Emanuele Ruffaldi
MMI S.p.A.
Calci, Italy
emanuele.ruffaldi@mmimicro.com

Carlo Alberto Avizzano
TeCIP Institute
Scuola Superiore Sant'Anna
Pisa, Italy
c.avizzano@santannapisa.it

***Abstract*—Railway line surveillance is important for providing safe and smooth travel of trains under effects of environmental or human-generated damages to the railway. This work presents a Structure from Motion pipeline specifically designed with the aim of supporting the monitoring operations of the railway infrastructure using a monocular camera mounted on the train's tractor. Within this work we developed a dynamical reconstruction instrument based on the mathematics of the projective geometry for handling the problem of localization, by triangulation techniques of points, lines, whole objects and of other known elements. Exploiting the a-priori knowledge of the scene structure (known track gauge) and the camera intrinsic parameters it is possible to reconstruct in metric dimension the trajectory of the train and the position of the detected object. The approach proposed here combines Computer Vision techniques to detect the significant elements and to classify a set of features with Bayesian filtering. Algorithms for this specific purposes have been developed in order to identify the rail track geometry, and a line-based approach has been adopted to assess the camera poses. Starting from these first estimates, a manifold Unscented Kalman Filter operates on the set of robustly matched features, fusing heterogeneous cues about the camera orientation and using RANSAC to find the best solution. Consequently, the detected objects can be triangulated and localized. An analysis using real captures is reported to prove the quality of the results obtained.**

I. INTRODUCTION

Due to the raising need to enforce the surveillance of the railroad, a variety of systems have been developed in order to guarantee the safety of the vehicles and its users. State-of-the-art technology has been employed such that control systems may rely on the usage of train on-board sensors (Locomotive-based systems) as well as on the sensory equipment settled within the infrastructure, allowing communication with the rail vehicles (Infrastructure-based system). The use of Unmanned Rail Vehicles (URV) represents an innovative solution in this framework, bringing the twofold advantage of being a self powered base, and allowing for the installation of a larger number of sensors, thus reducing the disruption of the rail traffic [1]. In this scenario the environment analysis and reconstruction is strongly affected by a proper localization of detected objects and of the sensing vehicle.

The issue of estimating the location of a vehicle from the images captured by its sensors has been widely dealt in literature, under different conditions [2]. However, only fewer works address this problem with the usage of a monocular camera. Lately [3] it is also possible to use inertial data together with the camera captures for an online mapping as well as for elaborating precomputed maps.

Among the filtering based approaches, the Extended Kalman Filter (EKF) represents the most used instrument to assess camera poses. An example of EKF-based monocular Simultaneous Localization And Mapping (SLAM) can be found in Civera et al. [4] where they present an inverse parametrization of the image depth for point features and the relative uncertainty. When the presence of non-linearity becomes considerable, difficult to model, or the gradient is computationally complex, an Unscented Kalman Filter (UKF) is preferred, like in the SLAM implemented by Chekhlov [5] to address the unreliability of feature detection and matching, modeling the observation as the expected image projection of the features. Also, the UKF on manifold [6] can be considered to handle sensor fusion [7].

In the field of autonomous driving the detection and localization of static objects in urban environments, e.g. traffic lights, signs, has seen a continuous evolution. By collecting a large amount of images to train Convolutional Neural Networks (CNN), it is possible to accurately recover the 3D positions of the recognized objects, given the camera poses. For revisited areas, results can be even enhanced in terms of hit rate and position accuracy [8].

Song and Manmohan [9] proposed an alternative strategy for motion detection and dynamic object localization in a moving car. The algorithm makes use of the large KITTI dataset [10] to detect common objects in a road scenario and associate relevant 3D bounding boxes. A subsequent constrained optimization is applied between the road plane and to the densely tracked boxes.

Previous contributions to obstacle detection within a less structured environment such as railways consist in tracking the area occupied by the railroad in the image [11], eventually

employing thermal cameras [12].

Wohlfeil [13] presented a vision based approach to assess an automatic detection of rail switches for determining which course of the rail track is taken by a train.

In the present work we address a Structure-from-Motion (SfM) and obstacle detection problem aimed at an URV for railway surveillance. In more detail our system is based on the visual-input, gathered by a monocular calibrated, undistorted, camera.

The proposed solution consists of an intelligent vision system mounted on the train tractor, able to localize itself and the objects detected in proximity of the rails, while moving within a rail track environment. The importance of providing the most complete information as possible on the inspected rail road is to better contextualize possible detected anomalies.

The recovery of the camera pose is addressed by exploiting topological information, such as image line-features, and by applying an algebraic SfM algorithm. This process is based on the projective geometry mathematics and requires the conversion of frames into spherical images, following the approach of Ly et al. [14]. In order to obtain for each frame an estimate of the camera pose, an UKF on manifold has been implemented. The filter allows to refine previous estimates, while directly taking in the input the matched point features. This behavior is useful especially when the presence of lines in the environment is difficult to extract from a frame, thus when the algebraic algorithm may experience failure.

Models trained to recognize specific objects in a frame allow for the tracking and the localization of known elements of the infrastructure, whose presence in proximity of the railroad is nominal. Due to the lack of a dataset for railway's elements we extracted our own dataset to classify poles and signs.

In addition, the robustness of reconstruction is further improved by combining UKF results with other optical flow analyses, which include the position of the projective Focus of Expansion (FoE) and the displacement of images above the horizon line.

The algorithm has been tested by using a video collected with an on-board camera. The results demonstrated the proper operation of the UKF in a real scenario, when the features are collected from a noisy environment and motion innovation is combined with motion cues computed by the image flow analysis.

The paper is structured as follows: first we provide a listing of nomenclature, then we briefly present the overall architecture. Section IV presents the detection, followed by pose estimation of Section V. Finally we provide evaluation in Section VI and conclusions in Section VII.

II. Nomenclature

The manuscript merges concepts from computer vision with state reconstruction techniques, derived from control and automation. Wherever it is possible we tried to use the nomenclature that is common in the respective background, but, in order to avoid misunderstanding for terms that are named with the same symbols, some of them have been renamed. This section clarifies the notation used to represent each term and describes the associated symbols, chosen in order to avoid any ambiguity, though the nomenclature may differ from the convention.

Parameters regarding the camera:

- K: the $\mathbb{R}^{3\times3}$ intrinsic camera parameters matrix encoding the focal distance(s) and the camera (X, Y) centre in pixel coordinates;
- $\mathcal{E}(t, R) \in SE(3)$: the extrinsic camera parameters matrix, describing the camera pose, containing translation $t \in \mathbb{R}^3$ and orientation $R \in SO(3)$;
- $\Pi = K\mathcal{E}(t, R)$: the camera matrix;

Quantities involved in reconstruction:

- ${}^{A}T_{B}$: the transformation from the reference frame B to frame A;
- ${}^{F}A_i$: the 3D location of a point feature in frame F. When F is not specified the world frame is considered;
- FoE: the image Focus of Expansion expressed in homogeneous coordinates;
- ${}^{F}L_i$: any line feature detected within an image frame;
- ${}^{F}\pi_{ij}$: the plane identified by the line ${}^{F}L_i$ and the camera center ${}^{F}C_j$;

Quantities involved in the Kalman filtering:

- Δt: the fixed sample time between frames, corresponding to 30Hz in this work;
- k: the k-th time step, associated to the image capture;
- x: the Kalman filter state = $[\mathcal{E}(t, R), \omega, u, \alpha, a]^T$: the system state vector with the following components:
  - $u \in \mathbb{R}^3$: the linear velocity;
  - $a \in \mathbb{R}^3$: the linear acceleration;
  - $\omega \in \mathbb{R}^3$: the angular velocity;
  - $\alpha \in \mathbb{R}^3$: the angular acceleration;
- $y = [z, \Delta\psi]$: the system observation variables, where
  - $z \in \mathbb{R}^2$: is the (X, Y) image position of a matched image feature between two frames;
  - $\Delta\psi$: the yaw angle between two frames estimated from the horizontal motion of the FoE;
- $w \in \mathbb{R}^6$: the process noise acting on a and α with associated covariance matrix Q;
- $v \in \mathbb{R}^3$: the measurement noise with associated covariance matrix V;
- $\mathcal{M}$: manifold, e.g. Euclidean, Lie Group or combination
- $\mathcal{N}_{\mathcal{M}}(\mu, \Sigma)$: multivariate Gaussian defined over the manifold $\mathcal{M}$
- χ: sigma points used in the Unscented Filter
- $\mathcal{G}_{\mathcal{K}}$: the Kalman Gain
- P: covariance matrix of the state estimation error;

III. Approach Overview

Our interactive SLAM technique is structured into a sequence of stages, shown in Fig. 1. Two different pieces of information are extracted from images. First, the image flow is processed to detect relevant point and line features. Secondly, the Focus of Expansion position and the horizon slide are computed to estimate the yaw rate.

The camera pose estimation module employs this data, by applying first an algebraic algorithm based on line features detected in a triplet of images and located using multiple view geometry. The result is refined through UKF on manifold which combines filter prediction with actual feature detection and estimated yaw rotation.

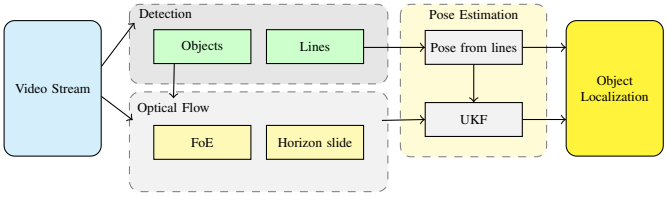
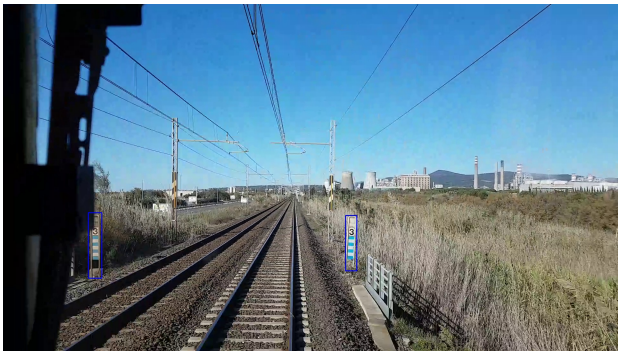


Fig. 1. Pipeline of the proposed approach.

IV. Detection

The identification of a suitable detection and matching algorithm represents a crucial choice to get the most complete information about the scene. Our detection module is not only concerned on general point features matched across a sequence of frames, but includes in the process also the lines, defining the tri-dimensional geometry of the environment, e.g. rails and railway sleepers. Furthermore, the railway signs, poles and small poles, along the railway can be recognized using dedicated HAAR Cascade models as shown in Fig. 2.



(a)

(b)

Fig. 2. HAAR Cascade-based detection of the kilometer signs (a) and of the poles indicating the turn (b).

A. Objects detection

A dedicated dataset representing the common objects in Italian Railway network has been generated. Object recognition has been performed using the Viola-Jones learning algorithm [15] in place of alternative Deep Learning approaches since the objects have no significant variation from one to another. Moreover, there are not sufficient images to train more complex classifiers. Possible isolated false alarms can be easily recognized as such and can be discarded a-posteriori.

B. Line detection

The extraction of the rail track lines from real images is done employing a combination of Hough transform and template matching techniques as explained in Algorithm 1. The rails are constantly viewed by the camera. Even during turns, rails have low curvature, and in the lower portion of the image they can be approximated as straight lines. In order to locate these lines, each frame has been split into horizontal strips. And the 15 closer to the image bottom are used to search for straight lines using the Hough transform. The rest of the tracks is then searched by operating a template matching operation between strips, starting from the bottom, and using the results of previous match as template for the above strip.

The detection of rail sleepers is more unstable since they are often covered by rocks. To improve the success rate we introduced an automatic approximation method, which traces a line passing through a feature in the area between the rail. Its slope is flat in a condition of straight track and, during turns, it has a value proportional to the slope of the detected rail lines and to the y-image coordinate of the feature. Note that the system is able to discriminate straight from turn, by considering the displacement of a small frames placed at the central part of the image.

Algorithm 1 Railway extraction

$frame \leftarrow$ first frame
$iter \leftarrow 30$
loop: extract each frame of the video
while $frame$ is not empty **do**
    evaluation of the lowest stripe of the image
    $leftX[0] \leftarrow$ Hough line search
    $rightX[0] \leftarrow$ Hough line search
    iterative evaluation of the upper stripes of image
    for $i \leftarrow 1$ to $iter$ **do**
        template matching technique
        $leftX[i] \leftarrow$ Find position of the template closer to $leftX[i-1]$
        $rightX[i] \leftarrow$Find position of the template closer to $rightX[i-1]$

C. Focus of Expansion tracking

The Focus of Expansion represents the epipole of all the images. When a camera z-axis is aligned with the frontal direction of motion, its position corresponds to the projection of the optical centre onto the image sensor, i.e. the last column

of K. If the camera has an offset rotation w.r.t. this axis (said R_c), we may estimate the new position of the focus of expansion as:

$$FoE(R_c) = \begin{bmatrix} c_x \\ c_y \\ 0 \end{bmatrix} + R_c \begin{bmatrix} 0 \\ 0 \\ f_{xy} \end{bmatrix} \quad (1)$$

where c_x, c_y represent the optical center of the camera, and f_{xy} the focal distance of the lens measured in pixels.

Since this position only depends on constant information, such as the camera intrinsic parameters and the camera-train orientation when the initial orientation is not known, the FoE can be estimated by locating the track convergence point during a straight motion (e.g. when the tracks are fully linear).

From perspective geometry we know that the FoE does not change during linear motion. However its position can be altered when an angular velocity is over-imposed during the motion. During the train motion an online localization of FoE point is achieved by computing the intersection point of rails and, when possible, using the optical flow of recognized objects (signs or poles).

By tracking the dynamic location of FoE, additional motion information can be extracted to evaluate the consistency of the matched features, to estimate the yaw rate between frames, and to improve the self localization task. Since the motion of the train is mainly planar, the distance of the FoE w.r.t. its ideal position provides information about the yaw rate.

D. Points Detection

Video frames are searched to detect relevant features which can be used for SLAM operation. Each feature is identified with an associated image point (${}^F A_i$). Image points tracking is based on the ORB detection algorithm [16] and the brute-force matcher as provided by OpenCV. In addition, the optical flow helps assessing the quality of matched features. Feature motion is ideally radial from the FoE. Let δA_i be the displacement variation of features ${}^F A_i$ w.r.t the previous frame (${}^{F-1} A_i$), we expect relative angle (θ_{A_i}) being small:

$$\theta_{A_i} = \frac{\delta A_i \times ({}^F A_i - FoE)}{\|\delta A_i\| \|({}^F A_i - FoE)\|} \approx 0 \quad |\theta_{A_i}| < thresh \quad (2)$$

This constraint has been relaxed setting $thresh = 0.02$, to preserve feature matches in case of image discretization noise. Points that do not comply with equation (2) check are rejected before SLAM operation.

V. Pose Estimation

After obtaining the different features and the yaw rate we proceed with pose estimation as follows.

A. Localization using line features

The recovery of relative poses for calibrated cameras with (partially) overlapping fields of view is addressed as a line-based SfM expressed in a unitary spherical space [14]: inside the 3D scene we consider a set of unitary spheres centered at the optical centers of each camera as shown in Fig. 4.

Fig. 3. Tracking of the rails and railroad ties in order to achieve their line fitting. Small rail elements within a frame have been subsequently detected using a template matching algorithm to obtain the whole rail profile. For each template the blue markers indicate the position of lower left corner. The lines approximating the rails are gathered by a polynomial fitting of the lowest 15 templates.

Relevant environment features will then be projected onto these spheres. According to this mapping, each line feature (L_i) generates a diametrical circle identified by the intersection of the sphere with a plane (π_{ij}) passing through the camera optical center C_j and the 3D line itself. All lines having the same vanishing direction intersect in two antipodal points, thus either of the two points can be used to represent the corresponding vanishing point.

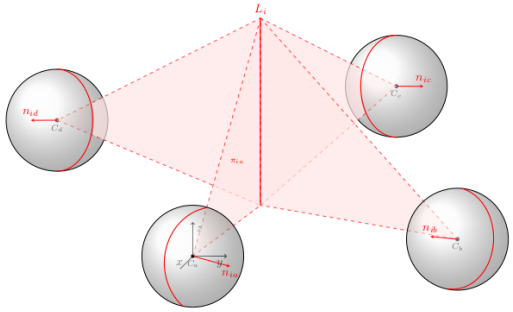


Fig. 4. Line projection to spherical images. The 3D line L_i is projected to diametrical circles with corresponding unit normals n_{ij}.

The projection of distant points from the camera onto the image plane is only due to the camera orientation. This assert still holds after the projection from a frame to a sphere and can be exploited to map the geometry of the environment to the local camera frame. The vanishing frontal, vertical and transversal directions in a railway infrastructure can be estimated from the rails, the poles and the railway sleepers. These directions define three vectors on the unit sphere, and can be used to identify the rotation of each camera: for each j-th camera, the rotation matrix M_j maps the coordinate system defined by vanishing points to the camera frame.

Fig. 4 also shows that the location of camera centers changes the diametrical circle projected onto the unit spheres. Using such property it is possible to recover relative displacement associated to different frames. In particular we proceed by

computing the n_{ij} vectors orthogonal to π_{ij}. Choosing one of the cameras as the origin of the global coordinate system $\{W\}$, e. g. the a-th camera, the translation directions can be computed for a triplet of cameras viewing the same set of lines.

Let M_a the rotation matrix associated to the camera placed at the origin of the global frame, then the rotation matrix from j-th local camera frame to global ${}^W R_{C_j}$, can be computed for each camera as:

$$ {}^W R_{C_j} = M_a^T M_j, \quad \{W\} \equiv \{C_a\} $$

This leads to the following definition for the extrinsic camera parameters:

$$ [{}^W R_{C_j} | {}^W t_j] = \begin{cases} [I|0], & j = a \\ [R_j|t_j], & j \neq a \end{cases} $$

Having obtained the relative rotations R_j for all cameras, for each pair of cameras, e.g. (b, c), we can express and use the line constraint as in equation (3). The equation embeds the two unknown translations directions $\tilde{t}_b$ and $\tilde{t}_c$ in a 6-by-1 matrix, which can be obtained computing the null space of the following:

$$ [n_{ia}]_\times R_b^T n_{ib} n_{ic}^T \tilde{t}_c - [n_{ia}]_\times R_c^T n_{ic} n_{ib}^T \tilde{t}_b = 0 \qquad (3) $$

Equation (3) allows us to fully reconstruct the relative position of each camera with the exception of one parameter that matches to the global (image to world) scaling factor that cannot be recovered with only monocular-based information. To recover this quantity we used prior knowledge: the rails distance computed with the intersection points lying on the edges of a railway sleeper line, can be locally reconstructed imposing their known distance to 1.435 m.

Taking one of these points A, matched between two frames j and k, imposing their equality in global frame accordingly to the parameters previously computed $[R_j, \tilde{t}_j]$, $[R_k, \tilde{t}_k]$, the unknown projective scale $\mathcal{S} \in \mathbb{R}$ can be recovered using the following equation:

$$ R_j({}^{C_j}A) - \mathcal{S}\tilde{t}_j - [R_k({}^{C_k}A) - \mathcal{S}\tilde{t}_k] = 0 \qquad (4) $$

B. UKF for camera poses improvement

When the number of lines viewed by a triplet of cameras is not sufficient or if the displacement of the matched lines is too small, the previously introduced algebraic algorithm may not find a solution for obtaining the camera poses. To handle these cases, an additional module for the camera orientation and location assessment has been developed as an UKF-based refinement.

We defined a camera state-vector as $x = [\mathcal{E}, \omega, u, \alpha, a]^T$, where all terms have been defined in the Nomenclature section. User input and rail shapes were modeled as process additive-noise w acting on the rotational and translational acceleration α and a. The diagram in Fig. 5 and the Algorithm 2 present an overview of our filtering process.

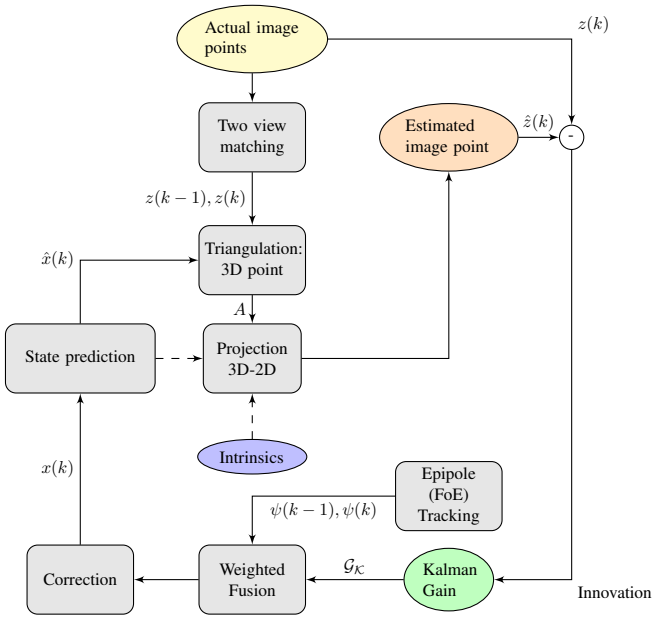


Fig. 5. The manifold UKF allows integrating 2D matched points with other motion cues obtained by the visual analysis. The diagram presents the information flow originating from the image point at time k and the state and previous step. Multiple matches are fused using RANSAC obtaining a weighted fusion among contributions.

The transformation $\mathcal{E}(k-1) \rightarrow \mathcal{E}(k)$, computed with the algebraic algorithm for two consecutive frames, is used to initialize the state estimates $\hat{x}(1)$ and $\hat{x}(2)^-$. with the assumption that the first camera position is the global origin.

The UKF state prediction provides two camera poses $\mathcal{E}(k-1)$, $\mathcal{E}(k)$ that combined with camera intrinsic parameters allow triangulating the 3D position of any point-feature (z) identified in the associated frames.

The backward projection of ${}^k A_i$ to the frame at time instant k produces an estimate for the 2D inhomogeneous position $\hat{z}(k)$. The difference of $\hat{z}(k)$ with the actual inhomogeneous coordinates $z(k)$ gives the innovation of the filtering process, which is employed to correct the state estimate.

After iterating this process for each matched point-feature between a couple of frames, a set of possible values for the state at time instant k is available and a RANSAC algorithm has been implemented to filter outliers. The fusion of the inlier estimates gives the corrected estimate of the state $\hat{x}(k)$.

C. Algebra for Kalman filter over manifold

The state of our Kalman filter contains different variables among which the transformation of the camera $\mathcal{E}(k)$ that belongs to the special Lie Group $SE(3)$. For dealing with Kalman filter with variables belonging to a manifold that is locally homeomorphic to the Euclidean space we chose to use a manifold formulation [7]. In particular the structure of the manifold is encapsulated using two operators boxplus $\boxplus$ and boxminus $\boxminus$ as follows:

$$ \boxplus : \mathcal{M} \times \mathbb{R}^n \rightarrow \mathcal{M} \qquad (5) $$

$$ \boxminus : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^n \qquad (6) $$

Algorithm 2 Vision-based UKF

for $k \leftarrow 2$ to *number of frames* **do**
  $\mathcal{E}_{k-1} \leftarrow \hat{x}_{k-1}$
  $\mathcal{E}_k \leftarrow \hat{x}_k^-$
  for $i \leftarrow 1$ to *number of matched points* **do**
    $A \leftarrow$ Triangulation$(\mathcal{E}_{k-1}, \mathcal{E}_k, K, z_{i,k-1}, z_{i,k})$
    $\tilde{z} \leftarrow K\mathcal{E}_k(A - t_k)$
    Reprojected point A onto k-th frame:
    $\hat{z}_{i,k} \leftarrow \tilde{z}/\tilde{z}(3)$
    Innovation:
    $I \leftarrow z_{i,k} - \hat{z}_{i,k}$
    UKF CORRECTION, obtaining $\hat{x}_{k_i}$
  inlier estimates $\leftarrow$ RANSAC$(\hat{x}_{k_i})$
  $\hat{x}_k, P_k \leftarrow$ FUSION of the *inlier estimates*
  $\hat{x}_{k+1}^-, P_{k+1}^- \leftarrow$ UKF PREDICTION

The former operation performs a motion over the manifold from the given point along the direction specified in tangent space. The latter computes the motion in the tangent space that brings one point in the manifold to another. In the euclidean case these operations map directly to plus and minus respectively, while in the Lie Group case $\boxplus$ is the composition of transformation expressed by the algebra, and $\boxminus$ is the distance between two elements in the group expressed in the algebra:

$$\boxplus(X, v) = exp(v)X \tag{7}$$

$$\boxminus(X, Y) = log(XY^{-1}) \tag{8}$$

Given these two operations it is possible to introduce a concentrated parameters Gaussian distribution over the manifold $\mathcal{N}_{\mathcal{M}}(\mu, \Sigma)$ in which the Gaussian has a mean belonging to an element of the group, and the covariance is expressed in the tangent space. Sampling an element from the manifold Gaussian corresponds to sampling the tangent space with the given covariance Σ and then applying the $\boxplus$ operator with the mean μ.

D. Unscented Transformation

A fundamental element of Kalman filter over manifold is the Unscented Transformation (UT) that allows to apply a function from one manifold to another when the input is a multivariate Gaussian $\mathcal{N}_{\mathcal{M}}$. The UT follows a quadrature approach, that evaluates the function in a set of points around the mean that depends on the covariance of the variable, called sigma points. In this formulation $2M + 1$ sigma points are used, where M is the dimension of the tangent space, that is the dimension of the covariance matrix. The UT comprises three steps: (1) computation of sigma points, (2) evaluation of each point with the function, (3) reconstruction of the output Gaussian in the output manifold. The adaptation of the UT from Euclidean space to manifolds requires to replace the plus operator with the $\boxplus$ one in the first stage.

The extraction of $2M + 1$ sigma points is guided by the covariance of the state estimation error P:

$$\chi^{(i)} = \hat{x} \boxplus \left(\sqrt{(M + \lambda)P}\right)_i, \quad i = 1, \ldots, 2M \quad \lambda \in \mathbb{R} \tag{9}$$

where $(\sqrt{(\cdot)})_i$ is the i-th column of the square root of the covariance matrix typically obtained by Cholesky decomposition and λ is a weight that controls the importance of the mean with respect to the surrounding sigma points. Given the function $f : \mathcal{M} \to \mathcal{S}$ we transform the sigma points $\chi_X^{(i)} \in \mathcal{M}$ into $\chi_Y^{(i)} \in \mathcal{S}$.

The reconstructed Gaussian has a mean $\mu_Y \in \mathcal{S}$ computed as the weighted average of each transformed sigma point $\chi_X^{(i)}$. Due to the nature of the manifold this operation cannot be computed in closed form but only in iterative form. The covariance of the Gaussian is straightforward and uses $\boxminus$:

$$P_{YY} = \sum_{i=0}^{2M} W_i^{(c)} (\chi_Y^{(i)} \boxminus \mu_Y)(\chi_Y^{(i)} \boxminus \mu_Y)^T \tag{10}$$

Finally, it will be useful to compute the covariance between the input an the output of the function as a matrix M by S:

$$P_{XY} = \sum_{i=0}^{2M} W_i^{(c)} (\chi_X^{(i)} \boxminus \mu_Y)(\chi_Y^{(i)} \boxminus \mu_Y)^T \tag{11}$$

The weights $W_i^{(m)}$ and $W_i^{(c)}$ have been computed as Wan et al. [17].

Thanks to the Unscented Transformation over the manifold it is possible to express both the Prediction and the Correction steps of the filter.

E. UKF Prediction

In the prediction step the state $X = [SE(3), \mathbb{R}^3, \mathbb{R}^3, \mathbb{R}^6]^T$ is transformed by the nonlinear function $f(\cdot)$, which regulates the state dynamics

$$\begin{cases} \mathcal{E}(k+1) = \mathcal{E}(k) \boxplus [\omega(k)\Delta t, u(k)\Delta t]^T \\ u(k+1) = u(k) + a(k)\Delta t \\ \omega(k+1) = \omega(k) + \alpha(k)\Delta t \end{cases} \tag{12}$$

In the manifold formulation the process noise is not additive meaning that we need to employ state augmentation, that is to evaluate the prediction over $x*$ that contains both the original state x and the noise.

F. UKF Correction

The Kalman gain $\mathcal{G}_{\mathcal{K}}$ uses the variance of the estimated observation P_{zz}, and the covariance between the predicted state and the observation $P_{x^- z}$:

$$\mathcal{G}_{\mathcal{K}} = P_{x^- z} P_{zz}^{-1}$$

Then the new state is obtained by using the $\boxplus$ and $\boxminus$ operators:

$$x(k) = x^- \boxplus \mathcal{G}_{\mathcal{K}}(z \boxminus z*) \tag{13}$$

$$P(k) = P^- - \mathcal{G}_{\mathcal{K}} P_{zz} \mathcal{G}_{\mathcal{K}}^T \tag{14}$$

Where $z*$ is the observation.

G. Fusion

Each matched feature provides a separate correction contribution to the state. We fuse these contributions using RANSAC to remove outliers and select a center of mass ($\hat{x}_m(k)$) as the vector with the lowest error, among the inlier estimates. A weighted mean is operated on the rest of the elements, according to their covariance in order to determine their contribute of pose $\Delta\hat{x}$.

The final state estimate, only based on the features information, is given as a combination of the center of mass $\hat{x}_m(k)$ and the contribute $\Delta\hat{x}$.

The information of angular asset given from the FoE position is reliable, thanks to its redundant tracking (rail lines intersection, optical flow in detected objects boxes) and it is worth integrating in the filtering process. Therefore, taking into account the yaw angle cues derived by the FoE sliding along the image x-axis, an additional state estimate can be obtained.

The raw data of $\Delta\psi$ gathered in this fashion have been filtered from outliers and smoothed. The estimated angular velocity $\hat{\omega}_{FoE}(k)$ has been computed as follows:

$$\hat{\omega}_{y,FoE}(k) = \frac{\Delta\psi(k,k-1)}{\Delta t} \tag{15}$$

The final state estimate is computed as a weighted fusion of $\hat{x}_{FoE}(k)$ with the one given by the matched features displacement $\hat{x}(k)$:

$$P_f(k) = \left(P^{-1}(k) + P_{FoE}^{-1}(k)\right)^{-1} \tag{16}$$

$$\hat{x}_f(k) = \left(\hat{x}_{FoE}(k)P_{FoE}^{-1}(k) + \hat{x}(k)P^{-1}(k)\right) P_f(k) \tag{17}$$

VI. Evaluation and Results

In order to evaluate the proposed method we consider a recording obtained on board a train tractor and compare the estimated trajectory first to the information obtained by the satellite image of the railway, and then to the results provided by the OpenSfM software. The recording has been made on a single track railway and it lasts for 8min and 20 seconds for 9 km at about 72 km/h.

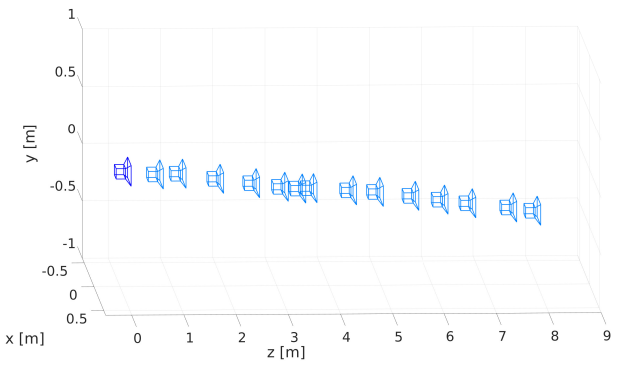


Fig. 6. Fifteen consecutive camera poses recovered from the lines matching across triplets of images during straight motion of the train. The estimated linear velocity is realistic (about 15.85 m/s).

The application of the algebraic algorithm alone hardly provides a solution for a long sequence of consecutive frames, especially during turns. Anyway, in case of a straight motion of the train a good result can be obtained with a sufficient number of matched lines (Fig. 6). For this reason, a filtering process is needed for a complete reconstruction of the camera motion.

The HAAR cascade classifier for recognizing kilometer signs, trained on a set of 60 images, reported a minimum hit rate of 0.995 and a maximum false alarm rate of 0.05. The detection of the poles, marking a curve, experienced lower performances being less textured elements. The latter classifier, trained on a dataset of 248 images, has a minimum hit rate of 0.95 and a maximum false alarm rate of 0.2. Inter frame coherence analyses can additionally improve the associated accuracy.

In order to prove the accuracy of the trajectory reconstruction and object localization, a comparison is made with a satellite picture of area. The image acquisition starts with the view shown in Fig. 7. The distance traveled by the train fits with the scale of the map. Moreover the kilometer signs along the railway have been detected and the resulting location, triangulated with the estimated camera poses, is close to its actual position. This result suggests that the linear velocity (66.74 km/h) has been successfully estimated using our pipeline. Nonetheless the performances in camera asset and angular velocity estimation was not good enough to fully reproduce the turn. Excluding outliers, the mean position w.r.t. the start camera frame is at $(X, Y, Z) = (0.7677, -6.9634, 234.7150)$, with standard deviations $(s_x, s_y, s_z) = (0.1398, 0.1657, 0.8809)$.

Among the SfM techniques at the state of the art, the same dataset of pictures has been processed through OpenSfM, an Open Source project for SfM[1], for a further verification of the results obtained. Using OpenSfM the shape of the track was correctly reproduced, with a sampled set of frames as shown in Fig. 8 (a). According to the reconstruction obtained in this framework, the train traveled for 110.72 m in 10.56 s, (37.56 km/h) which is significantly slower than expected. This points out the importance of computing the right projective scale factor to recover the metric dimension together with the topology of the environment. The comparison of the 3D reconstructions for camera displacement in Fig. 8 (b) validates the geometric results, since the behaviors are similar, especially along the y and z directions.

VII. Conclusions

The proposed approach represents a vision-based instrument which is able to reconstruct the trajectory of a railway vehicle, equipped with a monocular camera. Our method can find application in the inspection systems of railway environments, since it successfully estimates the linear velocity of a train and provides a metrical localization of the recognized object. The ability to recover both the distance traveled and the position of relevant element along the railway, such as kilometers

[1]https://github.com/mapillary/OpenSfM

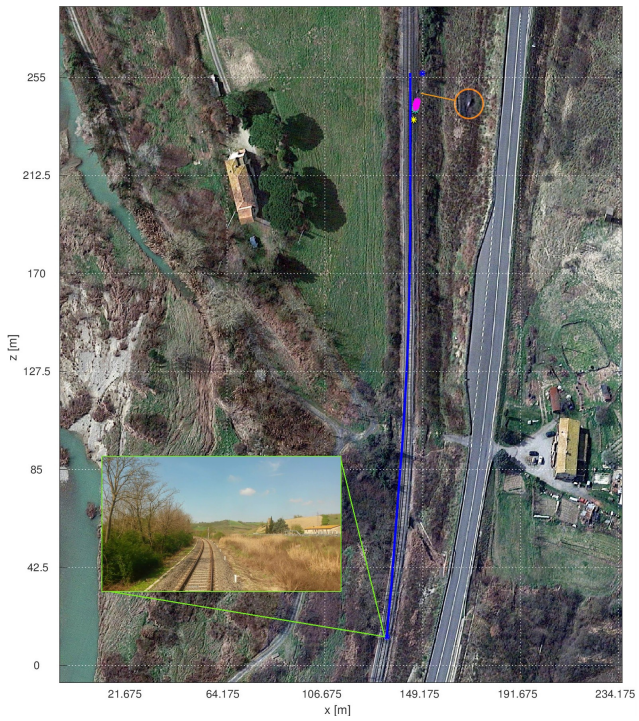


Fig. 7. Satellite picture of the rail track (Volterra, 43.3572, 10.7890). The number of pixels contained in the map resolution allowed for the definition of the scale factor of 0.0862 m/pixels. According to our reconstruction (in blue), the train traveled for 248.89 m along global z axis. The colored markers indicate the positions of the features lying within the bounding boxes of the detected kilometer sign w.r.t. the first camera pose, obtained by triangulation with the camera poses estimates. Each color stands for a different view of the sign. The actual position of this sign appears to be few meters further.

signs and turn markings, gives an important contextualization for possible anomalies detected. Exploiting prior information such as the rail gauge and a small set of images to train the objects classifiers, the proposed technique offers a promising contribution to the obstacle detection process within the rail infrastructure.

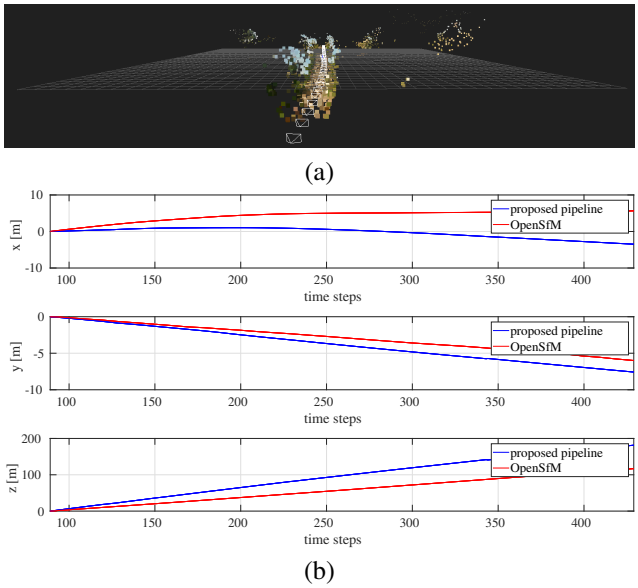


Fig. 8. OpenSfM reconstruction obtained processing 1 frame every 10 of the same dataset. (a) This result has been compared along each direction to the one obtained using our method. The analysis is done on the turn part of the trajectory (b).

References

[1] M. P. daSilva, W. Baron, *et al.*, "State-of-the-art technologies for intrusion and obstacle detection for railroad operations," tech. rep., United States. Federal Railroad Administration, 2007.

[2] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion*.," *Acta Numerica*, vol. 26, pp. 305–364, 2017.

[3] T. Schneider, M. T. Dymczyk, M. Fehr, *et al.*, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, 2018.

[4] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular slam," *IEEE transactions on robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[5] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway, "Real-time and robust monocular slam using predictive multi-resolution descriptors," in *International symposium on visual computing*, pp. 276–285, Springer, 2006.

[6] S. Hauberg, F. Lauze, and K. S. Pedersen, "Unscented kalman filtering on riemannian manifolds," *Journal of mathematical imaging and vision*, vol. 46, no. 1, pp. 103–120, 2013.

[7] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, vol. 14, no. 1, pp. 57–77, 2013.

[8] G. Dabisias, E. Ruffaldi, H. Grimmett, and P. Ondruska, "Value: Large scale voting-based automatic labeling for urban environments," in *ICRA*, IEEE, 2018.

[9] S. Song and M. Chandraker, "Joint sfm and detection cues for monocular 3d localization in road scenes," in *CVPR*, pp. 3734–3742, IEEE, 2015.

[10] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *CVPR*, 2016.

[11] F. Kaleli and Y. S. Akgul, "Vision-based railroad track extraction using dynamic programming," in *Intelligent Transportation Systems*, pp. 1–6, IEEE, 2009.

[12] A. Berg, K. Öfjäll, *et al.*, "Detecting rails and obstacles using a train-mounted thermal camera.," in *SCIA*, pp. 492–503, 2015.

[13] J. Wohlfeil, "Vision based rail track and switch recognition for self-localization of trains in a rail network," in *Intelligent Vehicles Symposium (IV)*, pp. 1025–1030, IEEE, 2011.

[14] D. S. Ly, C. Demonceaux, P. Vasseur, and C. Pégard, "Extrinsic calibration of heterogeneous cameras by line images," *Machine vision and applications*, vol. 25, no. 6, pp. 1601–1614, 2014.

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, vol. 1, pp. I–I, IEEE, 2001.

[16] E. Rublee, V. Rabaud, *et al.*, "Orb: An efficient alternative to sift or surf," in *ICCV*, pp. 2564–2571, IEEE, 2011.

[17] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing*, pp. 153–158, IEEE, 2000.