DOES: A Deep Learning-based approach to estimate roll and pitch at sea.

Fabiana Di Ciaccio[a], Paolo Russo[b] and Salvatore Troisi[a]

[a] *International PhD Programme/UNESCO Chair "Environment, Resources and Sustainable Development", Department of Science and Technology, Parthenope University of Naples, Centro Direzionale Isola C4, 80143, Naples, Italy; (fabiana.diciaccio, salvatore.troisi)@uniparthenope.it.*

[b] *Department of Computer, Control and Management Engineering "Antonio Ruberti", University of Rome La Sapienza, via Ariosto 25, 00185, Rome, Italy; paolo.russo@diag.uniroma1.it.*

ABSTRACT

The use of Attitude and Heading Reference Systems (AHRS) for orientation estimation is now common practice in a wide range of applications, e.g., robotics and human motion tracking, aerial vehicles and aerospace, gaming and virtual reality, indoor pedestrian navigation and maritime navigation. The integration of the high-rate measurements can provide very accurate estimates, but these can suffer from errors accumulation due to the sensors drift over longer time scales. To overcome this issue, inertial sensors are typically combined with additional sensors and techniques. As an example, camera-based solutions have drawn a large attention by the community, thanks to their low-costs and easy hardware setup; moreover, impressive results have been demonstrated in the context of Deep Learning. This work presents the preliminary results obtained by DOES, a supportive Deep Learning method specifically designed for maritime navigation, which aims at improving the roll and pitch estimations obtained by common AHRS. DOES recovers these estimations through the analysis of the frames acquired by a low-cost camera pointing the horizon at sea. The training has been performed on the novel ROPIS dataset, presented in the context of this work, acquired using the FrameWO application developed for the scope. Promising results encourage to test other network backbones and to further expand the dataset, improving the accuracy of the results and the range of applications of the method as a valid support to visual-based odometry techniques.

CRediT authorship contribution statement

Fabiana Di Ciaccio: Conceptualization, methodology, formal analysis, software, resources, data curation, writing, review and editing. **Paolo Russo:** Conceptualization, methodology, formal analysis, software, resources, data curation, writing, review and editing. **Salvatore Troisi:** Formal analysis, methodology, resources, supervision, review and editing.

1. Introduction

The pose estimation problem consists in estimating the position and orientation of a vehicle, device, human or robot with respect to a reference frame, through the use of different kinds of internal or external sensors. The accurate measurement of the orientation plays in fact a critical role in a wide range of activities, e.g., robotics and human motion tracking, bio-logging for animal behaviour research, aerial vehicles and aerospace, gaming and virtual reality applications, medicine and biotechnology, indoor and outdoor pedestrian navigation, maritime and/or autonomous navigation. When Global Navigation Satellite Systems (GNSS) are not able to provide correct information about the position and attitude of a vehicle, navigation and localization operations are generally performed through the integration

ORCID(s): 0000-0002-4271-2255 (F.D. Ciaccio); 0000-0002-1886-3491 (P. Russo); 0000-0002-4311-6156 (S. Troisi)

of different kind of sensors: inertial, odometry, laser and sonar ranging sensors, underwater positioning systems, etc. (Alatise and Hancke, 2017).

In the last years the use of low-cost technologies is becoming widely spread in numerous applications: this means that the accuracy of the pose obtained by these systems can be affected by even more disturbing factors than the traditional high-performing methods. In these circumstances, the development of accurate and reliable orientation estimation algorithms can still be considered a very challenging task, being at the basis of the localization process and of the consequent performances of the device employed for any specific task. This finds particular application in the context of the navigation, be it aerial, maritime or pedestrian, underwater/underground or in surface, autonomous, remotely operated or traditionally performed. In the specific case of maritime navigation, the information of position and orientation of a vessel is of great interest for seafarers in different operations and scenarios (e.g., open sea, congested harbours and waterways) as it is strictly related to the safety of the navigation at any level (Del Pizzo et al., 2018). The same goes for Unmanned Surface Vehicles (USVs), which are mainly employed in environmental monitoring, safety or navigation support and research operations. In this case, a non accurate estimation of the orientation can severely compromise the ultimate success of the mission, especially when paired to low-cost sensors and poor GNSS support.

The Inertial Measurement Unit (IMU) gives the instantaneous speed and position of the vehicle without the need for external references by integrating the measures of angular velocity and linear acceleration obtained through its three orthogonal rate-gyroscopes and –accelerometers respectively. Unfortunately, several problems are associated with these sensors; among the others, measurements are noisy and biased and the errors increase over time due to the drift of the sensors. Micro Electro-Mechanical Systems (MEMS) Attitude Heading Reference Systems (AHRS) integrate to this configuration a magnetometer which measures the variation of the Earth's magnetic field: this allows to instantly calculate an improved estimation while benefitting from lighter weight, smaller sizes and lower prices. The great potential of these devices makes them suitable for several applications exploiting the pure orientation estimation, like geomatics, surveys, augmented reality, etc.

Vision-based methods are also frequently employed for the scope: these techniques allow to understand the surrounding environment by detecting its visual features through a camera; captured color data with its high resolution contains in fact several information, and the sensors are generally low-costs and with an easy hardware setup. In this context, the detection of the horizon line is an important attribute for the maritime image processing, as it allows to estimate the camera's orientation with respect to the sea surface other than restricting the object search region when detection is performed, thus reducing the processing time and the false detection problem. Several approaches have been proposed to solve this task, however the accuracy and the processing time of the horizon line detection on high-resolution maritime image still face some issues (Ganbold and Akashi, 2020).

In the last decade, Visual Odometry (VO) and Visual Simultaneous Localization and Mapping (VSLAM) techniques have been successfully developed; however, their application can be challenging too, especially when deployed in non-textured environments or poor-light conditions. Visual Inertial Odometry (VIO) systems are proposed to eliminate these limitations, combining IMU and camera to improve motion tracking performance (Huang, 2019). The current VIO systems heavily rely on manual interference to analyze failure cases and refine localization results, other than requiring careful parameters tuning procedures for the specific environment they have to work in. In recent years, Deep Learning (DL) has drawn significant attentions due to its potential in learning capability and its robustness to camera parameters and challenging environments. These data-driven methods have successfully learned new features representations from images that are used to further improve the motion estimation (Han et al., 2019).

With the aim of providing further enhancements in the orientation estimation methodologies, this paper presents DOES, Deep Orientation (of roll and pitch) Estimation at Sea, a new supportive DL model which can be combined to the actual low-cost IMU-based configuration. This approach is not intended to substitute the current systems, but aims at improving the robustness of traditional methods when some limitations occur: the unavailability of GPS signals in indoor and under-surface environment, the undesirable high drift of inertial sensors in case of extended GPS outages and the issues of possible confusion with nearby robots for SONAR & RADAR are some of the limitations associated with these navigation systems. Visual-based methods help in this sense, since they constitute a powerful tool to estimate the pose of a camera through which the motion information is further recovered. These techniques can be classified as geometric or learning based: in the first case the camera geometry is explored to estimate the motion, whereas in the latter the model is fed with labeled data and then trained to accomplish the same task. The advantage of the learning-based methods is that they do not require the knowledge of the camera parameters and can estimate the orientation with correct scale even for monocular cases (Poddar et al., 2018). Moreover, visual methods can be further integrated with traditional, IMU-based orientation estimation algorithms to obtain a robust and reliable visual-inertial odometry system (Forster et al., 2016). The work presented in this paper develops an affordable visual, learning-based backbone which estimates the attitude of a monocular camera which will be mounted on a vehicle.

The idea behind DOES is in fact to train a DL model able to output the vehicle attitude (in terms of roll and pitch angles) by processing the sea horizon view recorded by a low-cost camera. In particular, the latter needs to be mounted on the surface of an autonomous robot (or, similarly, on the bridge of traditional ships) with its axis parallel to the vehicle longitudinal axis, to correctly frame the horizon line. A similar approach could be further tested on Unmanned Aerial Vehicles (UAVs) too. To lay the foundation for this task, preliminary intensive tests have been conducted to verify the validity of the approach. Different DL architectures have been tested for the processing of the images acquired through an Android smartphone's camera.

In this context, the lack of datasets specifically designed for DL-based orientation estimation at sea has been evidenced. While tackling this issue, the need of acquisition methods assuring the synchronism of the measurements for a reliable Ground Truth (GT) has been addressed too. For this reason, this paper presents also the first release of the ROll and PItch at Sea (ROPIS) dataset (Fig. 1), which has been created through FrameWO, an Android application developed for the scope. The choice of employing low-cost sensors meets the necessity to develop affordable and smart tools to enhance the orientation estimation; for this reason, the first deployment of the dataset has been acquired using open-source libraries and software. In this preliminary release, the operating user acquires the data in the proximity of the seashore trying to simulate the real behaviour of a ship in navigation.

The main contribution of this work stands in the provision of a supportive low-cost technology aiming at improving the accuracy of the attitude estimation results in different approaches, without the need to configure camera models or considering related issues; the obtained results are promising and strongly encourage to work for further improvements.

The paper is organized as follows: Section 2 gives a brief overview on the existing literature on the orientation estimation task exploited through different traditional, visual and DL-based methods; Section 3 gives a theoretical foundation to the subject, introducing the attitude estimation problem to further describe the DL architectures which best fit the task. In Section 4 the ROPIS dataset will be presented, highlighting the issues and solutions encountered during the app creation and the data acquisitions. Section 5 details the experiments performed on DOES while the obtained results will be presented and discussed in Section 6; final considerations and future objectives will conclude the work in Section 7.

2. Related works

The accurate measurement of the orientation plays a critical role in a wide range of activities. AHRS sensors (i.e. accelerometers, gyroscopes and magnetometers) provide reliable measurements whose integration gives accurate information about the pose (position and attitude) of any object they are rigidly attached to. In the last decade, traditional methods have seen a huge improvement due to the integration with different kind of sensors, aiming at reducing the inertial-related error accumulation and the costs whilst enhancing the robustness of the methodology. As previously mentioned, one of the most effective integration is made through visual-based method, leveraging the potential of visual features and the low-cost of the devices. The following paragraphs give a concise review of the existing literature in the field of orientation estimation.

2.1. Inertial-based methods

There exists a large amount of literature on the use of inertial sensors for position and orientation estimation. The reason for this is related to their robust algorithms and their accurate solutions which makes them suitable for being used in several fields. Interestingly, relatively simple position and orientation estimation algorithms work quite well in practice, even if the model choice can sensibly affect the accuracy of the estimates (Kok et al., 2017).

There is a large and ever-growing number of application areas for inertial sensors, as for example robotics and human motion tracking (Avci et al., 2010; Luinge and Veltink, 2005), bio-logging for animal behavior research (Fourati et al., 2010), aerial vehicles and aerospace (Adler et al., 2015; De Marina et al., 2011), gaming, virtual reality and indoor pedestrian navigation (Vertzberger and Klein, 2021; Renaudin and Combettes, 2014; Harle, 2013), etc. In fact, the use of accurate inertial sensors and magnetic compasses was first introduced in the navigation field, but along with the development of MEMS technology, low-cost and small-size inertial and magnetic compass sensors appeared in various kinds of consumer electronics, game consoles, virtual reality applications and so on. The orientation representations and sensor fusion still remain the challenges to overcome (Phuong et al., 2009). Real-time orientation estimation algorithms based on low-cost IMU are analyzed by Kim and Golnaraghi (2004), where the approach is based on the relationships between the quaternion representing the platform orientation and the measurements of the sensors and the integration is performed through an Extended Kalman Filter (EKF). Baerveldt and Klang (1997) developed a low-cost and low-weight attitude estimator for autonomous helicopters based on an inclinometer and a gyroscope, while fusing the data coming from the sensors through a classic complementary filter; Gebre-Egziabher et al. (2000) proposed a gyro-free, quaternion-based attitude determination system which exploits low cost sensors. Valenti et al. (2015) implemented a complementary filter able to infer Micro Aerial Vehicle (MAV) attitude from observations of gravity and magnetic field, with the final algorithm able to work with both IMU and MARG sensors. De Marina et al. (2011) exploited an AHRS device together with a Unscented Kalman Filter algorithm to perform attitude estimation on UAVs. The same filter has been used by Allotta et al. (2016), which developed a novel navigation system for autonomous underwater vehicles that works without the presence of a GPS device, not available in underwater scenarios. Li and Wang (2013) proposed an Adaptive Kalman Filter which is able to provide pose estimations based on low-cost AHRS devices, while Di Ciaccio et al. (2019) and Michel et al. (2017) investigated the use of AHRS in smartphones as cheap but reliable devices for angles estimation. A novel error-state Kalman filter is presented by Vitali et al. (2020), which yields highly accurate estimates of IMU orientation that are robust to poor measurement updates from fluctuations in the local magnetic field and/or highly dynamic movements. An indoor pedestrian navigation method based on shoe-mounted MEMS IMU and ultra-wideband is discussed by Wen et al. (2020), which used a quaternion-based Kalman Filter to integrate the data and to reduce the complexity of the method. Aligia et al. (2021) presented a new orientation estimation strategy for a non-accelerated platform: it is based on a low-cost IMU and the orientation angles are obtained through a nonlinear Luenberger observer, while the common magnetometer offsets are calibrated by a recursive least-square algorithm. Schnee et al. (2020) utilized common bicycling motions to calibrate the 2D- and 3D-mounting orientation of a MEMS IMU on an electric bicycle. The method is independent of sensor biases and requires only a very low computation expense, so the estimation can be realized in real-time.

2.2. Vision-based methods

The possibility to employ visual data to perform orientation and in general pose estimation has been widely deepened in the past decades. Many researches have been focused on the horizon line detection, due to its relevance for visual geo-localization, port security, etc. However, some special features in real marine environments (e.g., clouds clutter, sea glint and weather conditions) frequently result in different kinds of interference in optical images. Wang et al. (2016) proposed a Sea-Sky Line (SSL) detection method for USVs based on the computation of the gradient saliency, through which the line features of the SSL are effectively enhanced while other disturbances are attenuated. The SSL identification is achieved according to regions contrast, line segment length and orientation features, and optimal state estimation of SSL detection is implemented by a cubature Kalman filter. Jeong et al. (2018) presented a fast method for detecting the horizon line in maritime scenarios. It combines a multi-scale approach and a region-of-interest (ROI) detection, which is an efficient way to reduce the amount of required processing information. The results are then combined to produce a single edge map on which the Hough transform and a least-square method are sequentially applied to accurately estimate the horizon line. The Hough transform is also used by Yongshou et al. (2018), which proposed a sea-sky line detection system based on the local Otsu segmentation; similarly, Sun and Fu (2018) recognize the horizon line in maritime images through a two-phase, coarse-fine detection algorithm which increases the overall method robustness. Another quick horizon line detection method is proposed by Praczyk (2018), which extracts the horizon line in real maritime image with improved reliability and faster execution with respect to other competitors. The horizon detection through vision sensors is also frequently exploited to obtain redundant orientation information in the field of unmanned aerial navigation. For example, Carrio et al. (2018) proposed two attitude estimation methods: the first one searches for the best line fitting the horizon in thermal images, which allows to further estimate the pitch and roll angles using an infinite horizon line model. The second method exploits a Convolutional Neural Network (CNN) which predicts the angles on the basis of the raw pixel intensities from the same kind of images.

However, these methods alone cannot be considered totally robust and reliable, since the position and slope of the horizon are strictly related to the camera intrinsic (i.e., focal length, optical center, pixel aspect ratio and skew) and extrinsic (rotation and translation) parameters and to the model used to parametrize them. Ligorio and Sabatini (2013) surveyed a plethora of methods which perform pose estimation by fusing visual, inertial and magnetic measurements, integrating them through the use of an EKF. The combined use of IMU and vision information has been explored by Alatise and Hancke (2017), which exploits SURF visual features together with accelerometer and gyroscope data to retrieve the robot pose in an indoor setting. A comprehensive analysis of the behaviour of these features when used for visual odometry can be found in the work of Chien et al. (2016).

VO, VIO and SLAM algorithms have recently received much attention for their efficient and accurate ego-motion estimation in robotics. A VIO algorithm for the estimation of the motional state of UAVs with high accuracy is presented by Hong and Lim (2018). It is based on the fusion of visual data and pre-integrated inertial measurements in a joint optimization framework and the on a stable initialization of scale and gravity using relative pose constraints. To account for the ambiguity and uncertainty of VIO initialization, a local scale parameter is adopted in the online optimization.

The use of stereo camera sensors for VO is a low-cost and effective way to estimate attitude, but may encounter problems in underwater setting due to poor imaging condition and inconsistent motion caused by water flow. Zhang et al. (2018) proposed a robust and effective stereo underwater VO system that can overcome the aforementioned difficulties and accurately localize the AUV. In the context of underwater robotics, another VO method designed to be robust to these visual perturbations is presented by Ferrera et al. (2019): it demonstrated to outperform state-of-the-art SLAM methods under many of the most challenging conditions. A novel keyframe-based SLAM system with loop-closing and relocalization capabilities targeted for the underwater domain is proposed by Rahman et al. (2019). This paper addresses drift and loss of localization by providing a robust initialization method to refine scale using depth measurements and a fast preprocessing step to enhance the image quality. Quan et al. (2019) presented a tightly coupled monocular VI-SLAM algorithm, which provides accurate and robust motion tracking at high frame rates on a standard CPU. A visual-inertial EKF is exploited to track the motion, then a globally consistent map is constructed to feed it back to the EKF state vector and reduce the drift. In a parallel thread, a global map is constructed to perform a keyframe-based visual-inertial bundle adjustment to optimize the map, together with a correction module to further eliminate the accumulated drift. ORB-SLAM3 (Campos et al., 2021) is another worth mentioning method, as it is the first system able to perform visual, visual-inertial and multi-map SLAM with monocular, stereo and RGB-D cameras, using pin-hole and fisheye lens models. It uses a feature-based tightly-integrated VI-SLAM system that fully relies on Maximum-a-Posteriori estimation, even during the IMU initialization phase, resulting in a system that operates robustly in real time, in small and large, indoor and outdoor environments, which is 2 to 5 times more accurate than previous approaches.

The rise of Deep Learning, with powerful architectures able to tackle complex tasks such as classification (Huang et al., 2017), detection (He et al., 2017), segmentation (Russo et al., 2019), denoising (Russo et al., 2021), super resolution (Wang et al., 2018), has definitely changed the way vision data is exploited for pose estimation. Instead of relying on engineered, fixed features (e.g. SIFT (Lowe, 1999), SURF (Bay et al., 2006)), recent algorithms exploit deep networks as powerful features extractors or by directly estimating the pose vector in an end-to-end model, from input images to the output prediction. For example, in order to estimate camera orientation, Rambach et al. (2016) exploited a LSTM deep network together with a linear Kalman Filter to combine IMU and camera data, while in DeepVIO (Han et al., 2019) the authors fused 2D optical flow features together with standard inertial data, obtaining state of the art results on KITTI (Geiger et al., 2013) and EuRoC (Burri et al., 2016) datasets. The combination of a traditional IMU with a LIDAR laser scan has been proposed by Li et al. (2019), which built a recurrent CNN to perform this aggregation on a scan-to-scan basis. (Li et al., 2018) proposed a method to estimate a camera six degrees of freedom and absolute scale by exploiting unsupervised data, getting good results in terms of pose accuracy on KITTI benchmark. In the more recent work of Almalioglu et al. (2019), the authors developed a generative framework able to exploit a GAN (Goodfellow et al., 2014) model on unlabelled RGB images for 6-DoF pose camera motion prediction, demonstrating the efficacy of their approach both on KITTI and Cityscapes (Cordts et al., 2016) datasets. The former method has been improved by Feng and Gu (2019) with a stack of GAN layers which demonstrated to be effective on ego-motion estimation tasks. A comprehensive review of the state of the art deep models for pose estimation can be found in the work of Zhao et al. (2020).

3. Method

This section aims at providing a theoretical background to fully understand the fundamentals of the proposed work. In particular, a general overview on the orientation estimation process is given in subsection 3.1, with some details on the sensors embedded in an AHRS and on the coordinate frame to which the smartphone device (and the related measures) is referred. Subsection 3.2 presents in a concise but detailed way the deep architecture models analysed and tested during the work.

3.1. Orientation estimation overview

The orientation of a rigid body is usually expressed by a transformation matrix in which the elements are generally parameterized in terms of Euler angles, rotation vectors, rotation matrices, and unit quaternions (Bernal-Polo and Barberá, 2017). The Euler angles are the most intuitive expression as they allow a simple analysis of the body orientation in the 3D space. These angles are defined as follows:

- ϕ represents the rotation around the x axis (*roll* angle);
- θ defines the rotation around the y axis (*pitch* angle);
- ψ is related to the rotation around the z axis (*yaw* angle).

The integration of high-rate raw data acquired by the IMU sensors or of the more cost-effective AHRS is at the basis of the orientation estimation process. The accelerometer measures the acceleration in m/s^2 applied to a device, including the force of gravity: velocity is determined if the linear acceleration component is integrated once and position if the integration is performed twice. The results can be of poor accuracy due to the extensive noise and accumulated drift from which it suffers. The gyroscope measures the device rate of rotation (i.e. the angular velocity) in rad/s, from which the rotation angle can be calculated by integration. Gyroscopes run at a high rate, allowing them to track fast and abrupt movements, but they suffer from serious drift problems caused by the accumulation of measurement errors over long periods. Therefore, the fusion of both an accelerometer and gyroscope data is suitable to determine the pose of an object and to make up for the weakness of one over the other. The magnetometer measures the Earth's magnetic field in μT, which is helpful in heading determination; the drawback is that the presence of metallic objects within the environment could influence data collected through measurements. The drift introduced by the sensors causes errors accumulation: this means that the navigation information provided by the INS can be considered reliable and accurate only within short times, while it is still impossible for a pure inertial navigation system to maintain the high-precision level throughout a mission. For this reason, the integration of the measurements provided by the three sensors aims at reducing the errors accumulation caused by the single one; this is generally made through filtering techniques and fusion methods. Moreover, information provided by external devices can considerably improve the accuracy of the estimations, especially when low-cost sensors could facilitate the process and make it more practical.

In this context, the objective of the present work is to provide a supportive mean to improve the attitude estimations obtained by common AHRS: DOES is a low-cost DL architecture developed to recover orientation information from the view of a camera pointing the horizon at sea, which will be placed on the bow of a navigating vehicle in future experiments. The training has been performed on the ROPIS dataset, acquired using an application developed for the scope on an Android smartphone which simultaneously collects the frames and calculates the corresponding Ground Truth data using the AHRS sensors.

The IMU-AHRS measurements of the smartphones are generally expressed in a custom body reference frame. The Android developer website defines its frame relative to the device's screen when the device is held in its default orientation (see Figure 2, Android). In particular, the frame originates in the center of the device with the horizontal x axis pointing to the right, the vertical y axis pointing up and the z axis points toward the outside of the screen face, so that the the coordinates behind the screen have negative Z values. The related attitude information is then referred to the same coordinates.

During the ROPIS dataset acquisition the smartphone has been kept in landscape mode, recording the horizon view. It has to be noticed that the coordinate frame does not change its definition, so in this setting the z axis points in the user direction, the y axis to his/her left and the x upwards.

3.2. Deep Learning architectures

DOES model is composed of a pre-trained backbone CNN and two additional Fully Connected (FC) layers to output the roll and pitch estimates. Several, well established architectures have been tested as backbone for the final network, as for example the VGG16-19 (Simonyan and Zisserman, 2014) and ResNet18-50-152 (He et al., 2016); the resulting numerical comparison will be reported in Section 6, Tab. 3.

The VGG-16 and VGG-19 networks are based on the popular VGG architecture. They are composed of several convolutional layers followed by a Rectified Linear Unit (ReLU) activation function and interspersed by max pooling layers. Two FC layers are concatenated in order to produce the final features which are fed to a classification layer. These two networks differ only by the quantity and dimension of the convolutional layers employed, with a total number of parameters equal to $138M$ and $144M$ respectively. Despite being among the first developed deep architectures, with a huge amount of trainable parameters making them prone to overfitting, VGG models are still incredibly widespread, thanks to their ease of use for fine-tuning purposes on different tasks (He et al., 2019; Long et al., 2015).

ResNet is a family of deep models based on the *residual* architecture. Differently from the VGG, the ResNet is made of a series of residual blocks in which the feature maps calculated by the convolutional layers are added to the input, so that each residual block calculates an *update* (hence residual) of the input feature maps. This approach makes the network resilient to the vanish gradient problem (Veit et al., 2016), improving convergence speed and the final accuracy result. Moreover, all the ResNet models avoid the use of the FC layers after the convolutional blocks, reducing the total number of trainable parameters and thus lessening the overfitting effect on training data. Authors of ResNet developed three versions with different number of layers (18, 50, 152) and with different number of visual features before the classification step (512 for the former, 2048 for the others). The number of free parameters for the 18, 50 and 152 layers models are $11M$, $23M$ and $60M$ respectively.

In the experiments presented in this work, all the networks have been fine-tuned on the proposed ROPIS dataset starting from the ImageNet (Deng et al., 2009) pre-trained weights. The ResNet18 has been chosen among the others as the default DOES backbone since it produced the best accuracy while keeping at the same time a fast inference speed. Figure 3 reports the DOES network with the default ResNet18 backbone.

Two additional FC layers have been added as additional branches on top of the highest set of visual features in the backbone network to separately estimate the roll and pitch angles; for example, in the case of the ResNet models, this correspond to the global average pooling layer. Some different estimation procedures have been experimented, as the one described in (Ruiz et al., 2018): it proposes to map the float angle value to a set of fixed bins, which then undergo a standard classification procedure with a final mapping back to the float value. However, in this work it has been experimentally found that this approach adds a layer of complexity without increasing the overall performances; this led to the decision to add a FC layer for each angle, which is able to accomplish the regression task with a good accuracy. Both the backbone network and the additional FC layers are jointly trained by back-propagation with the use of a standard Mean Square Error Loss (squared L2 norm). Two separated losses are calculated for each of the two angles, as reported in Eq. 1 for roll (L_{roll}) and Eq. 2 for pitch (L_{pitch}), where y and $\hat{y}$ are the GT and predicted values respectively. The final loss L_{final} is then obtained as a simple addition of the aforementioned quantities, as shown in Eq. 3. The GT roll and pitch values have undergone a prior normalization process, which subtracts to each of them the mean and divides by the variance, both calculated over the entire dataset.

$$L_{roll}(y_{roll}, \hat{y}_{roll}) = \frac{1}{n}\sum_{i=1}^{n}(y_{roll} - \hat{y}^{i}_{roll})^2 \tag{1}$$

$$L_{pitch}(y_{pitch}, \hat{y}_{pitch}) = \frac{1}{n}\sum_{i=1}^{n}(y_{pitch} - \hat{y}^{i}_{pitch})^2 \tag{2}$$

$$L_{final} = L_{roll}(y_{roll}, \hat{y}_{roll}) + L_{pitch}(y_{pitch}, \hat{y}_{pitch}) \tag{3}$$

4. ROPIS data acquisition process

The lack of datasets designed for DL-based orientation estimation at sea lead to the necessity of searching for methods to acquire a set of data for the scope. In the following section, the development of the Android application and the obtained ROPIS dataset will be described in detail.

4.1. Device internal sensors and characteristics

In order to train the model, the dataset needs to contain a large amount of images showing the horizon and the corresponding GT data in terms of roll and pitch angles. The latter needs to be given with the best possible accuracy, as the learning process results will depend on it, which is strictly related to the instrumentation employed for the acquisition. With the aim of producing a low-cost and flexible solution, in this work the authors avoided the use of costly, high-end IMU devices and developed the FrameWOAndroid application to acquire the dataset through a common smartphone. The presented ROPIS dataset in its first release has been totally collected through a OnePlus Nord smartphone, equipped with the most common sensors (Table 1) and characterized by an average price.

The OnePlus Nord mounts a BMI260 IMU, which contains a 16-bit tri-axial gyroscope (G) and accelerometer (A) providing fast, precise inertial sensing in smartphones and Human-Machine Interface (HMI) applications (i.e., advanced gesture, activity and context recognition, etc.). The IMU is characterized by a noise density of $160\mu g/\sqrt{Hz}$ (A) and $0.008dps/\sqrt{Hz}$ (G), a Zero-g/Zero-rate offset of $\pm 20mg$ (A) and $\pm 0.5dps$ (G) and an output data rate up to $1.6kHz$ (A) and $6.4kHz$ (G). Moreover, it mounts the industry's first self-calibrating gyroscope with motionless Component Re-Trimming (CRT) functionality, which compensates MEMS typical soldering drifts, ensuring post-soldering sensitivity errors down to $\pm 0.4\%$ (Bosh).

The MMC5603 is a monolithic complete 3-axis Anisotropic Magnetoresistance Effect (AMR) magnetic sensor with on-chip signal processing. It has an on-chip automatic degaussing with built-in SET/RESET function, allowing to eliminate thermal variation-induced offset error (Null field output) and to clear the residual magnetization resulting from strong external fields. It has a true frequency response up to $1KHz$ and can measure magnetic fields within the full scale range of $\pm 30Gauss$ (G) with $2mG$ total Root Mean Square (RMS) noise level, enabling heading accuracy of $\pm 1deg$ in electronic compass applications (Memsic).

Table 1
OnePlus Nord smartphone general specifics (OnePlus).

General	Main Sensors	Rear Camera - Main
OS: OxygenOS Android™ 10	IMU: Bosch BMI260	Megapixels: 48
CPU: Qualcomm® Snapdragon™ 765G	Magn: MEMSIC MMC5603	Pixel Size: 0.8 μm/48M; 1.6 μm (4 in 1)/12M
GPU: Adreno 620	Camera: Sony IMX586	Lens Quantity: 6P
RAM: 8GB/12GB LPDDR4X	Proximity sensor	Aperture: f/1.75
Storage: 256GB UFS2.1	Ambient light sensor	OIS, EIS: Yes

The Sony IMX586 stacked CMOS image sensor is mounted as the main camera of the OnePlus Nord, and features 48 effective megapixels with an ultra-compact pixel size of $0.8\mu m$. The sensor uses the Quad Bayer color filter array, where adjacent $2x2$ pixels come in the same color, making high-sensitivity shooting possible. During low light shooting, the signals from the four adjacent pixels are added, raising the sensitivity to a level equivalent to that of $1.6\mu m$ pixels (12 megapixels), resulting in bright, low noise images (Sony).

4.2. FrameWO application development

The FrameWO app has been developed in a free Open Source environment, the B4X suite (AnywhereSoftware), which supports the majority of PC, smartphones and embedding operating systems (e.g., Android, iOS, Windows, MacOS, Linux, Arduino, RaspberryPI) and uses a modern version of Visual Basic as programming language. The Android version (B4A) allows to wrap existing Java code as an external library and then to reference it from the B4A IDE, obtaining in release mode performances similar to those of Java. The size of a simple app is generally around 100 KB.

As previously mentioned, the necessary prerequisite for the dataset to meet the scope of this study is to associate to each frame the corresponding GT; however, the images size is much more larger than that of the IMU data, thus introducing a delay in their storage which affected their simultaneity. For this reason, the app captures the frames in YUV format (allowing for a better compression of the image) and converts them in JPEG only at the end of the process; this also avoids to run out of memory during the acquisition. A detailed overview on the YUV model can be found in (Podpora et al., 2014). Furthermore, several tests have been performed to determine an acquisition frequency value suitable for both the high-rate IMU data and the low-rate camera frames: the application offers in fact the possibility to set the camera acquisition frequency in *msec* to choose the best option for the needs.

As regards the GT, the API of Android (Android) has been used to work on the raw measures read by the sensors and to obtain the Euler angles of interest. The *getRotationMatrix* function takes as input the gravity and geomagnetic field in vector form to compute the inclination matrix I and the rotation matrix R, transforming a vector from the device coordinate system to the world coordinate system (defined as a direct orthonormal basis). By definition, R is the identity matrix when the device is aligned with the world coordinate system (i.e., when the device X axis points toward East, the Y axis points to the North Pole and the device is facing the sky) and I is a simple rotation around the X axis transforming the geomagnetic vector into the same coordinate space as gravity, i.e., the world coordinate space (see Eq. 4, where g is the magnitude of gravity and m is the magnitude of the geomagnetic field).

$$\begin{aligned} \begin{bmatrix} 0 & 0 & g \end{bmatrix} &= R * gravity \\ \begin{bmatrix} 0 & m & 0 \end{bmatrix} &= I * R * geomagneticfield \end{aligned} \tag{4}$$

In order to isolate the gravity vector, a discrete-time low-pass filter with a smoothing factor $\alpha = 0.2$ has been applied to the accelerometer measurements. The Euler angles are recovered through the *getOrientation* function, which calculates them from the elements of the rotation matrix R (Android; OpenSourceProject).

The measurements are updated at the fastest rate provided by the Android API, which is in the order of few milliseconds. The time sampling has been set equal to 100msec, that means that 10 times in a second the device simultaneously registers the orientation and the corresponding image. As a final result, the data is saved in a directory named with the date and time of the specific acquisition, which is further renamed to specify the scenario characteristics of the moment. This directory contains all the frames, saved as n_YYYY-MM-DD_HHMMSS.jpg, and a data.txt file which lists the frame name, its index n, and the related GT.

4.3. Dataset structure

The ROPIS dataset in its first release has been mainly acquired in Italy, in the cities of Gaeta (Lazio) and Racale (Puglia). It consists of 22173 sRGB TrueColor JPEG images, with resolution set to 2592x1168, for a total dimension of 42.3 GB. Six different subsets have been acquired in as many locations, each presenting different characteristics in terms of scenarios and meteo-marine conditions; five of them have been chosen for the training set, from which a total of 100 frames has been separated for the validation set, and the last acquisition has been used as test set. The use of a dedicated test set with images coming from a separate location allows to verify the ability of DOES to generalize to new, different scenes with respect to the training and validation set. More in the specific, in each place eight different acquisitions have been made trying to simulate the behaviour of a ship in navigation in both static and dynamic conditions: this aims at emulating the induced oscillations which resemble the true motion of the ship. To improve the generalization ability of the model, the data has been acquired at different day times and with sunny and cloudy sky; Figure 4 shows different samples of the ROPIS dataset. Some aspects of this data need to be highlighted:

- The point of view of the ROPIS images presents some differences with respect to the acquisitions taken on board the ship, since it adds parts of the land in the image foreground, such as sand, rocks, etc. However, this does not affect the learning procedure as the DL networks are able to recognize useful and useless image features, discarding the latter.
- A frame representing the real view from a navigating vehicle should depict some elements in the scene, such as the bow structures and some part of the bridge floor from a ship, or some of the USV sections. Although these specific features do not appear in ROPIS, DOES demonstrated its robustness to similar images cluttering present in the frames. Further experiments will be made to precisely assess their impact on the learning process.
- The data acquisition has been made with the camera at a roughly fixed height of 1.5m with slight oscillations around this value: this considers, among the different vehicle movements, also the linear vertical -up/down- motion along the z axis (*heave*), corresponding to the smartphone x axis. It should be remarked that the pitch estimation is strictly related to the horizon height and thus to the the camera axis and view; for this reason, the horizon line should be obviously always visible in the frame.

The ROPIS dataset is intended to be further enhanced. The use of other low-cost cameras (to take into account the differences in the camera parameters and lens distortion) and the setting of a range of different camera height values aim at considering their impact on the training phase. Moreover, the acquisitions will be made in different scenarios, which will include adverse meteo-marine conditions and locations as ships bridge and USV platforms. The heterogeneity of the data fed to the network will enhance the model capability to generalize over more complex data and realistic settings, making it invariant to these parameters.

5. Experimental setup

In this section some details on the training process will be given, together with a brief overview of the evaluation metrics used to appraise the performance of DOES. Finally, the problem related to the comparison of DOES with other methods will be discussed.

5.1. Training details

DOES has been developed in Python programming language using the Pytorch framework; the code is publicly available[1]. DOES has been trained using a standard fine-tuning procedure: the backbone convolutional kernels were

[1] https://github.com/fabidicia/does

pre-trained on ImageNet while the additional FC layers have been initialized with random values drawn upon Pytorch default uniform distribution. Both convolutional and FC layers have been trained using the Adam optimizer (Kingma and Ba, 2014) and a fixed learning rate set to 0.001. DOES has been trained on the ROPIS training set for a total of 10 epochs: it has in fact been noticed that a larger number of epochs led to an increase of the overfitting without any improvement of the accuracy.

The images have been squared to a preliminary 2592x2592 resolution by the application of a zero-padding; this operation adds black bands to the smallest dimension to obtain a squared input whilst preventing the loss of information. The images have then been resized to a final resolution of 224x224; a zero mean-unit variance normalization has been applied to both the images and the GT sets, with the corresponding mean and variance calculated over the specific training data.

The data augmentation process consisted of random changes in the colours of the images, using the *ColorJitter* transformation function of Pytorch which allows to set different values of brightness, contrast, saturation and hue: this resulted in an increase of the training dataset which further enhanced the generalization abilities of DOES. No random cropping nor image flipping have been applied during this process: in fact, the former would have caused the neglecting of the relative sea height information given by the images while the latter could have changed the correct roll angle perception of the network. The data augmentation procedure has naturally been deactivated during the testing phase, while the zero-padding and resize processes have been applied also to the test images; furthermore, the predicted roll and pitch values have been de-normalized before calculating the evaluation metrics presented in the following paragraph 5.2. The selected data augmentation values (brightness and hue equal to 0.5, contrast and saturation equal to 5), as well as all the other training hyper-parameters, have been tuned on the validation set.

5.2. Evaluation metrics

DOES has been evaluated on the basis of the regression metrics implemented by the Scikit library in the *sklearn.metrics* module, which contains the most common utility functions to measure the regression performance.

The Mean Absolute Error (MAE) computes a risk metric corresponding to the expected value of the absolute error (Eq. 5); it is the average absolute difference between the predicted and the true value, expressed in the same scale as the data being measured. Each error contributes to MAE in proportion to its absolute value.

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} |y_i - \hat{y}_i| \tag{5}$$

The Root Mean Square Error (RMSE) represents the square root of the second sample moment of the differences between predicted values and the observed values (or the quadratic mean of these differences, also called residuals).

It is a measure of accuracy and it is sensitive to outliers (Eq. 6). In fact, since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors, making it more useful when large errors are particularly undesirable. RMSE does not necessarily increase with the variance of the errors, growing instead with the variance of the frequency distribution of error magnitudes.

$$RMSE(y, \hat{y}) = \frac{1}{n}\sqrt{\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2} \tag{6}$$

The Standard Deviation (STD) is a measure of the amount of dispersion (or variation) of the samples. A low standard deviation indicates that the values tend to be close to the mean μ (also called the expected value) of the set, while a high standard deviation indicates that the values are spread out over a wider range (Eq. 7).

$$\sigma(\hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mu)^2} \tag{7}$$

Finally, the Median Absolute Error (MedAE) is calculated by taking the median of all the absolute differences between the GT and the prediction (Eq. 8). It is a non-negative floating point with best value of 0.0, robust to outliers since the median is not affected by values at the tails.

$$MedAE(y, \hat{y}) = median(|y_i - \hat{y}_i|, ..., |y_n - \hat{y}_n|) \tag{8}$$

5.3. Methodology comparison

The comparison between DOES and other state of the art methods turned out to be a non trivial task for several reasons; among the others, the Deep Learning based solutions currently developed for the estimation of roll and pitch are either released without source code (as for example in (Carrio et al., 2018)) or employed for very different tasks (e.g., head pose estimation (Ruiz et al., 2018)), thus making the comparison not properly correct or practically impossible. Generally speaking, traditional Horizon Line Detection (HLD) algorithms can be used as a proxy for this kind of estimations; the roll and pitch angles can in fact be correlated to the slope and position of the horizon line. However, as previously mentioned, this would require the correct knowledge of the intrinsic and extrinsic camera parameters and of the transformation matrix between the camera and the smartphone reference systems. To address this problem, a Linear Least Squares method has been applied to calibrate the HLD algorithms on the basis of the minimization of the squared error calculated between their output predictions and the GT values.

More in detail, given a set of measurements $M = [m_1, m_2...m_n]$ and the corresponding set of ground truth values $[G = g_1, g_2...g_n]$, the aim is to approximate the solution for the over-determined linear system (Eq. 9).

$$\begin{bmatrix} g_1 = x_2 + x_1 * m_1 \\ g_2 = x_2 + x_1 * m_2 \\ . \\ . \\ g_n = x_2 + x_1 * m_n \end{bmatrix} \tag{9}$$

This system can be expressed in matrix form as in Eq. 10, where A is the known *design matrix* defined as $A = [M^T, 1^T]$, $B = G^T$ is the known target vector and $X = [x_1, x_2]$ is the solution of the Linear Least Square method. It represents the linear transformation (Eq. 11) which better minimizes the squared norm (Eq. 12).

$$AX - B = 0 \tag{10}$$

$$g = x_2 + x_1 * m \tag{11}$$

$$\frac{||Ax - b||^2}{2} \tag{12}$$

Two of the most renowned HLD algorithms by the scientific community have been selected to perform this comparison and are briefly described in the following lines.

The **Otsu** method (Otsu, 1979) is a popular technique used to threshold the image between sky and non-sky regions. It is a reasonable fast and simple algorithm which performs fairly well on heterogeneous sets of data. The threshold value T is automatically computed by the algorithm through the assumption that the grayscale histogram of the image pixels intensities is bi-modal; the threshold is set so that the distance between the two histogram peaks is maximized.

Ettinger et al. (Ettinger et al., 2003) is a computer vision-based HLD algorithm that performs exhaustive search in the 2D line parameters space over the whole image looking at the best values which separate sky from terrain. However, being a slow algorithm on high resolution images, a modified version has been implemented that uses a two-stage objective: the *global* one searches for a narrow range of combinations of the pitch and roll horizon line angles corresponding to a half-plane that likely subdivides the sky from the rest of the image. The *local* one aims at searching exhaustively through these combinations to find the half-plane that maximizes the difference (in average intensity) of the two half-planes in their immediate vicinity. This method assumes that the sky pixels have higher intensity values than the ground pixels (higher mean), and that the sky has higher consistency of representation (lower variance).

6. Results and discussion

This section contains an assessment of the results provided by DOES. Table 2 shows DOES performances with respect to the selected horizon line detection algorithms. DOES is able to achieve sensible better results both on roll and pitch angles, with a Mean Absolute Error close to 1.5°, as opposed to the other methods which exhibit worse performance on all the indicators.

Table 2
DOES performances compared to those of the two HLD methods.

	DOES		Otsu (1979)		Ettinger et al. (2003)	
	roll	*pitch*	*roll*	*pitch*	*roll*	*pitch*
MAE [deg]	**1.65**	**1.84**	4.48	3.76	4.04	3.77
RMSE [deg]	2.27	2.45	5.44	4.75	5.01	4.78
STD [deg]	1.55	1.61	3.09	2.90	2.97	2.93
MedAE [deg]	1.14	1.41	4.04	3.19	3.44	3.15

The MAE and the RMSE can be used together to diagnose the variation in the errors in a set of predictions. The RMSE is generally higher than the MAE, and the greater is the difference between them, the greater will be the variance in the individual errors of the samples; moreover, if the RMSE is close to the MAE, then all the errors are of the same magnitude. In the case of the current comparison, the small gap between RMSE and MAE demonstrates the ability of DOES to produce fewer outliers than Otsu and Ettinger. In addition, the STD values of the three methods show that the results obtained by DOES are significantly more clustered than the others, meaning that they are closer to the mean value and as such can be considered more reliable. The good performances of DOES are further confirmed by the MedAE value, which is sensibly lower than the counterparts. These findings can be summarized in Figure 5, which shows the MAE behaviour analysing the outputs percentage belonging to different MAE intervals (Fig. 5a) together with the empirical cumulative distribution (Fig. 5b) for the roll angle. The same evaluation can be made for the pitch angle (Fig. 6), which exhibits similar performances to the roll angle. Another important consideration related to this comparison regards the inference time of DOES; the average estimation time on a single image is 100-150msec with any of the tested backbones, while Otsu and Ettinger inference time is comprised between 100 and 11000 msec, making them unsuitable for real-time applications on high-resolution images.

Table 3 shows a detailed comparison between DOES with its default proposed network and some alternative backbones: DOES is able to produce good performances with all the residual networks, while both VGG-19 and VGG-19bn struggle to produce reasonable results. More in detail, the MAE and RMSE results of ResNet18 are slightly better then the 50- and 152-layers versions, with the powerful DenseNet161 model able to produce a similar accuracy only on the roll angle. The performing results obtained by the ResNet18, together with the fastest training and inference speed (due to the smaller number of trainable parameters TP with respect to the other architectures), make ResNet18 the first choice for the deployment of DOES as long as new models specifically developed for the scope will be released. Future work will focus on the use of lighter architectures developed for the specific use on low-resources embedded hardware (e.g., MobileNet, Howard et al. (2017)); this will lay the foundation for the deployment of the proposed model on embedded devices (e.g., Nvidia Jetson, Mittal (2019)) in real-time scenarios, in accordance with the aim of making DOES a supportive smart technology to improve the attitude estimations provided by low-cost sensors.

Furthermore, the ROPIS dataset has been used for an additional test in which a $1.33x$ zoom has been applied to the frames to simulate different camera parameters. In some cases, this corresponded to a crop in the image which removed the horizon line, thus making DOES unable to correctly estimate the angles. This reflects in a slight decrease of the performances: the roll MAE is equal to 2.10°, with a RMSE of 2.81°, while the pitch angle exhibits a 2.02° MAE and a 2.90° RMSE.

Table 3
Comparative results on different DOES backbones. TP indicates the number of trainable parameters.

	ResNet18 **$TP = 11M$**		ResNet50 $TP = 23M$		ResNet152 $TP = 58M$		VGG19 $TP = 139M$		VGG19bn $TP = 139M$		DenseNet161 $TP = 26M$	
	roll	*pitch*	*roll*	*pitch*	*roll*	*pitch*	*roll*	*pitch*	*roll*	*pitch*	*roll*	*pitch*
MAE [deg]	**1.65**	**1.84**	1.77	1.88	1.82	1.92	4.67	4.11	1.91	1.99	1.63	1.87
RMSE [deg]	2.27	2.45	2.40	2.51	2.44	2.54	5.60	5.18	2.57	2.61	2.23	2.48
STD [deg]	1.55	1.61	1.63	1.66	1.62	1.66	3.09	3.14	1.72	1.69	1.55	1.63
MedAE [deg]	1.14	1.41	1.28	1.46	1.36	1.52	4.26	3.43	1.47	1.56	1.14	1.44

Finally, a separated test (with no prior training or specific tuning) has been made on a set of 191 images presenting three main variations with respect to the ROPIS train and test data:

- The device: a smartphone Huawei P9 (Huawei Device Co., 2021) has been used, with the FrameWO App, to collect the data. The mounted dual-lens Leica camera has different characteristics with respect to the OnePlus Nord Sony camera: the P9 Leica 12 MP has in fact an aperture size of $f/2.2$, a focal length of $27mm$ (wide), a sensor size of $1/2.9''$ and a pixel size of 1.25μm.
- The location: the acquisition has been made in a different area of the Racale city (LE).
- The environment setting: the data has been collected rightly after the sunset, in a low-light condition which highly reduced the contrast in the frame, resulting in a very challenging scenario.

Despite these substantial changes in the sensor and in the overall acquisition, DOES obtained remarkable results, performing a 2.17° MAE and a 2.70° RMSE for the roll angle and a 2.22° MAE and a 2.71° RMSE for the pitch angle. This demonstrates that DOES can successfully generalize over various conditions and camera parameters, confirming its potential for more challenging settings and further employment as inertial systems support and visual-based odometry tasks.

It is worth mentioning that the accuracy of the results is proportioned to the precision of the GT data and thus of the systems employed to acquire it. In this case, the overall accuracy is strictly connected to the use of a smartphone AHRS which, although being limited to the low-cost sensors mounted on it, is still able to provide reliable and accurate measurements. The use of high-end and more expensive devices would in fact ensure a higher grade of GT accuracy with consequent improvements in the DOES performances.

7. Conclusions

This paper presents a novel Deep Learning-based approach to the attitude estimation problem, which has been developed and intensively tested on a new dataset (the ROPIS dataset) specifically built for the scope and released in the context of this work. Deep Orientation (of roll and pitch) Estimation at Sea (DOES) is able to predict the attitude of the device in terms of roll and pitch angles by analysing the frames recorded by the camera pointing towards the sea horizon. DOES has been tested using several known architectures (e.g., ResNet152, ResNet18, VGG19) and with different configurations and hyper-parameters, obtaining excellent results. Unlike other visual-based methods, DOES is able to produce the output without the explicit knowledge of the camera intrinsic and extrinsic parameters or the distortions introduced by the camera lens. There is in fact no necessity to make any assumption on the use of specific models to parametrize the camera, since the model training only depends on the dataset given as input; the latter generally provides different sampling characteristics, thus making the network able to learn and then estimate the attitude regardless of the camera specifics.

The ROPIS dataset has been created for this particular task and is here presented in its first release; the lack of public datasets suitable for DL applications made it necessary to search for a valid alternative for the experiments conduction. For this reason, the FrameWO Android application has been developed using the Open Source B4A platform and will be made publicly available online. This app allows to simultaneously acquire the frames to be fed to the model as input, and the attitude estimations measured through the internal sensors of the smartphone, which will be used as Ground Truth in the training/testing phases.

ROPIS dataset is intended to be further improved by the introduction of more subsets of data collected in different scenarios (i.e., during the dusk/dawn, rainy days, etc) and environments (e.g., different cities coastlines, onboard of a vessels), using different acquisition devices. This will improve the DOES ability to generalize over heterogeneous data, making it even more invariant to the camera configurations, the acquisition condition and cluttering factors, thus providing better results in any kind of situation in which the vehicle will be navigating. In this regard, the authors wish to encourage the users to download and test the FrameWO application with the aim of enhancing the ROPIS and its usage among the scientific community, to give a concrete contribution to this task.

The objective of this project is to develop a supportive technology to be integrated to the existing low-cost methodologies employed for the attitude estimation task. In fact, it has to be noticed that this approach has been specifically designed using affordable devices and applications and, as such, its results are not intended (at least in its preliminary version) to reach the accuracy provided by high-precision modern sensors. Further experiments will be made to test other light-weight DL architectures, which could be deployed on low-resources embedded hardware with the aim of providing better accuracy results in real-time applications on autonomous vehicles. These enhancements will make DOES a robust system to be integrated in visual and visual-inertial odometry methodologies.

Code availability section

DOES - Deep Orientation (of roll and pitch) Estimation at Sea

Contact: fabiana.dicia@gmail.com, +39 328-0935198

Hardware requirements: Nvidia GPU with CUDA 10+ support

Program language: Python 3

Software required: Python environment, CUDA 10+ library

Program size: 78.5KB (code), 39.3GB (dataset)

The source codes are available for downloading at the link: https://github.com/fabidicia/does

The dataset is available for downloading at the link: ROPIS Dataset

Acknowledgments

The Authors would like to give a special thanks to Mr. Alberto Greco, which has been of fundamental importance in the development stage of the app employed to acquire the ROPIS dataset.

References

Adler, S., Schmitt, S., Wolter, K., Kyas, M., 2015. A survey of experimental evaluation in indoor localization research, in: 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE. pp. 1–10.

Alatise, M.B., Hancke, G.P., 2017. Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter. Sensors 17, 2164.

Aligia, D.A., Roccia, B.A., De Angelo, C.H., Magallán, G.A., González, G.N., 2021. An orientation estimation strategy for low cost imu using a nonlinear luenberger observer. Measurement 173, 108664.

Allotta, B., Caiti, A., Costanzi, R., Fanelli, F., Fenucci, D., Meli, E., Ridolfi, A., 2016. A new auv navigation system exploiting unscented kalman filter. Ocean Engineering 113, 121–132.

Almalioglu, Y., Saputra, M.R.U., de Gusmao, P.P., Markham, A., Trigoni, N., 2019. Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks, in: 2019 International conference on robotics and automation (ICRA), IEEE. pp. 5474–5480.

Android, . Sensormanager.java. https://developer.android.com/guide/topics/sensors/sensors_overview. Accessed: 2021-11-06.

AnywhereSoftware, . Simple, powerful and modern development tools. https://www.b4x.com/. Accessed: 2021-10-04.

Avci, A., Bosch, S., Marin-Perianu, M., Marin-Perianu, R., Havinga, P., 2010. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey, in: 23th International conference on architecture of computing systems 2010, VDE. pp. 1–10.

Baerveldt, A.J., Klang, R., 1997. A low-cost and low-weight attitude estimation system for an autonomous helicopter, in: Proceedings of IEEE International Conference on Intelligent Engineering Systems, IEEE. pp. 391–395.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features, in: European conference on computer vision, Springer. pp. 404–417.

Bernal-Polo, P., Barberá, H., 2017. Orientation estimation by means of extended kalman filter, quaternions, and charts. Journal of Physical Agents 8. doi:10.14198/JoPha.2017.8.1.03.

Bosh, . Bmi260: Imu combining accelerometer and gyroscope. https://www.bosch-sensortec.com/products/motion-sensors/imus/bmi260/. Accessed: 2021-10-01.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R., 2016. The euroc micro aerial vehicle datasets. The International Journal of Robotics Research URL: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract, doi:10.1177/0278364915620033, arXiv:http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.full.pdf+html.

Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D., 2021. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Transactions on Robotics .

Carrio, A., Bavle, H., Campoy, P., 2018. Attitude estimation using horizon detection in thermal images. International Journal of Micro Air Vehicles 10, 352–361.

Chien, H.J., Chuang, C.C., Chen, C.Y., Klette, R., 2016. When to use what feature? sift, surf, orb, or a-kaze features for monocular visual odometry, in: 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ), IEEE. pp. 1–6.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3213–3223.

De Marina, H.G., Pereda, F.J., Giron-Sierra, J.M., Espinosa, F., 2011. Uav attitude estimation using unscented kalman filter and triad. IEEE Transactions on Industrial Electronics 59, 4465–4474.

Del Pizzo, S., Gaglione, S., Angrisano, A., Salvi, G., Troisi, S., 2018. Reliable vessel attitude estimation by wide angle camera. Measurement 127, 314–324.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee. pp. 248–255.

Di Ciaccio, F., Gaglione, S., Troisi, S., 2019. A preliminary study on attitude measurement systems based on low cost sensors, in: International Workshop on R3 in Geomatics: Research, Results and Review, Springer. pp. 103–115.

Ettinger, S.M., Nechyba, M.C., Ifju, P.G., Waszak, M., 2003. Vision-guided flight stability and control for micro air vehicles. Advanced Robotics 17, 617–640.

Feng, T., Gu, D., 2019. Sganvo: Unsupervised deep visual odometry and depth estimation with stacked generative adversarial networks. IEEE Robotics and Automation Letters 4, 4431–4437.

Ferrera, M., Moras, J., Trouvé-Peloux, P., Creuze, V., 2019. Real-time monocular visual odometry for turbid and dynamic underwater environments. Sensors 19, 687.

Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D., 2016. On-manifold preintegration for real-time visual–inertial odometry. IEEE Transactions on Robotics 33, 1–21.

Fourati, H., Manamanni, N., Afilal, L., Handrich, Y., 2010. A nonlinear filtering approach for the attitude and dynamic body acceleration estimation based on inertial and magnetic sensors: Bio-logging application. IEEE Sensors Journal 11, 233–244.

Ganbold, U., Akashi, T., 2020. The real-time reliable detection of the horizon line on high-resolution maritime images for unmanned surface-vehicle, in: 2020 International Conference on Cyberworlds (CW), IEEE. pp. 204–210.

Gebre-Egziabher, D., Elkaim, G.H., Powell, J., Parkinson, B.W., 2000. A gyro-free quaternion-based attitude determination system suitable for implementation using low cost sensors, in: IEEE 2000. Position Location and Navigation Symposium (Cat. No. 00CH37062), IEEE. pp. 185–192.

Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) .

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Advances in neural information processing systems 27.

Han, L., Lin, Y., Du, G., Lian, S., 2019. Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. arXiv preprint arXiv:1906.11435 .

Harle, R., 2013. A survey of indoor inertial positioning systems for pedestrians. IEEE Communications Surveys & Tutorials 15, 1281–1293.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

He, Y., Zhu, C., Wang, J., Savvides, M., Zhang, X., 2019. Bounding box regression with uncertainty for accurate object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2888–2897.

Hong, E., Lim, J., 2018. Visual-inertial odometry with robust initialization and online scale estimation. Sensors 18, 4287.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 .

Huang, G., 2019. Visual-inertial navigation: A concise review, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE. pp. 9572–9582.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708.

Huawei Device Co., L., 2021. Huawei p9. URL: `https://consumer.huawei.com/uk/support/phones/p9/`.

Jeong, C.Y., Yang, H.S., Moon, K., 2018. Fast horizon detection in maritime images using region-of-interest. International Journal of Distributed Sensor Networks 14, 1550147718790753.

Kim, A., Golnaraghi, M., 2004. A quaternion-based orientation estimation algorithm using an inertial measurement unit, in: PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No. 04CH37556), IEEE. pp. 268–272.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Kok, M., Hol, J.D., Schön, T.B., 2017. Using inertial sensors for position and orientation estimation. Found. Trends Signal Process. 11, 1–153. URL: `https://doi.org/10.1561/2000000094`, doi:`10.1561/2000000094`.

Li, C., Wang, S., Zhuang, Y., Yan, F., 2019. Deep sensor fusion between 2d laser scanner and imu for mobile robot localization. IEEE Sensors Journal .

Li, R., Wang, S., Long, Z., Gu, D., 2018. Undeepvo: Monocular visual odometry through unsupervised deep learning, in: 2018 IEEE international conference on robotics and automation (ICRA), IEEE. pp. 7286–7291.

Li, W., Wang, J., 2013. Effective adaptive kalman filter for mems-imu/magnetometers integrated attitude and heading reference systems. The Journal of Navigation 66, 99–113.

Ligorio, G., Sabatini, A.M., 2013. Extended kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: Comparative analysis and performance evaluation. Sensors 13, 1919–1941.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

Lowe, D.G., 1999. Object recognition from local scale-invariant features, in: Proceedings of the seventh IEEE international conference on computer vision, Ieee. pp. 1150–1157.

Luinge, H.J., Veltink, P.H., 2005. Measuring orientation of human body segments using miniature gyroscopes and accelerometers. Medical and Biological Engineering and computing 43, 273–282.

Memsic, . Monolithic, high performance, low cost 3-axis magnetic sensor. http://www.memsic.com/uploadfiles/2020/08/20200827165137254.pdf. Accessed: 2021-10-01.

Michel, T., Geneves, P., Fourati, H., Layaïda, N., 2017. On attitude estimation with smartphones, in: 2017 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE. pp. 267–275.

Mittal, S., 2019. A survey on optimized implementation of deep learning models on the nvidia jetson platform. Journal of Systems Architecture 97, 428–442.

OnePlus, . Oneplus nord - specs. https://www.oneplus.com/uk/nord-specs. Accessed: 2021-10-01.

OpenSourceProject, A., . Sensors overview. https://github.com/aosp-mirror/platform_frameworks_base/blob/master/core/java/android/hardware/SensorManager.java. Accessed: 2021-11-06.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics 9, 62–66.

Phuong, N.H.Q., Kang, H.J., Suh, Y.S., Ro, Y.S., 2009. A dcm based orientation estimation algorithm with an inertial measurement unit and a magnetic compass. Journal of Universal Computer Science 15, 859–876.

Poddar, S., Kottath, R., Karar, V., 2018. Evolution of visual odometry techniques. arXiv preprint arXiv:1804.11142 .

Podpora, M., Korbas, G.P., Kawala-Janik, A., 2014. Yuv vs rgb-choosing a color space for human-machine interaction., in: FedCSIS (Position Papers), pp. 29–34.

Praczyk, T., 2018. A quick algorithm for horizon line detection in marine images. Journal of Marine Science and Technology 23, 164–177.

Quan, M., Piao, S., Tan, M., Huang, S.S., 2019. Accurate monocular visual-inertial slam using a map-assisted ekf approach. IEEE Access 7, 34289–34300.

Rahman, S., Li, A.Q., Rekleitis, I., 2019. Svin2: an underwater slam system using sonar, visual, inertial, and depth sensor, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 1861–1868.

Rambach, J.R., Tewari, A., Pagani, A., Stricker, D., 2016. Learning to fuse: A deep learning approach to visual-inertial camera pose estimation, in: 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE. pp. 71–76.

Renaudin, V., Combettes, C., 2014. Magnetic, acceleration fields and gyroscope quaternion (magyq)-based attitude estimation with smartphone sensors for indoor pedestrian navigation. Sensors 14, 22864–22890.

Ruiz, N., Chong, E., Rehg, J.M., 2018. Fine-grained head pose estimation without keypoints, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 2074–2083.

Russo, P., Di Ciaccio, F., Troisi, S., 2021. Danae++: A smart approach for denoising underwater attitude estimation. Sensors 21, 1526.

Russo, P., Tommasi, T., Caputo, B., 2019. Towards multi-source adaptive semantic segmentation, in: International Conference on Image Analysis and Processing, Springer. pp. 292–301.

Schnee, J., Stegmaier, J., Lipowsky, T., Li, P., 2020. Auto-correction of 3d-orientation of imus on electric bicycles. Sensors 20, 589.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .

Sony, . Sony releases stacked cmos image sensor for smartphones. https://www.sony.com/en/SonyInfo/News/Press/201807/18-060E/. Accessed: 2021-10-01.

Sun, Y., Fu, L., 2018. Coarse-fine-stitched: A robust maritime horizon line detection method for unmanned surface vehicle applications. Sensors 18, 2825.

Valenti, R.G., Dryanovski, I., Xiao, J., 2015. Keeping a good attitude: A quaternion-based orientation filter for imus and margs. Sensors 15, 19302–19330.

Veit, A., Wilber, M.J., Belongie, S., 2016. Residual networks behave like ensembles of relatively shallow networks. Advances in neural information processing systems 29, 550–558.

Vertzberger, E., Klein, I., 2021. Attitude adaptive estimation with smartphone classification for pedestrian navigation. IEEE Sensors Journal 21, 9341–9348.

Vitali, R.V., McGinnis, R.S., Perkins, N.C., 2020. Robust error-state kalman filter for estimating imu orientation. IEEE Sensors Journal 21, 3561–3569.

Wang, B., Su, Y., Wan, L., 2016. A sea-sky line detection method for unmanned surface vehicles based on gradient saliency. Sensors 16, 543.

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C., 2018. Esrgan: Enhanced super-resolution generative adversarial networks, in: Proceedings of the European conference on computer vision (ECCV) workshops, pp. 0–0.

Wen, K., Yu, K., Li, Y., Zhang, S., Zhang, W., 2020. A new quaternion kalman filter based foot-mounted imu and uwb tightly-coupled method for indoor pedestrian navigation. IEEE Transactions on Vehicular Technology 69, 4340–4352.

Yongshou, D., Bowen, L., Ligang, L., Jiucai, J., Weifeng, S., Feng, S., 2018. Sea-sky-line detection based on local otsu segmentation and hough transform. Opto-Electronic Engineering 45, 180039–180039.

Zhang, J., Ila, V., Kneip, L., 2018. Robust visual odometry in underwater environment, in: 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO), IEEE. pp. 1–9.

Zhao, C., Sun, Q., Zhang, C., Tang, Y., Qian, F., 2020. Monocular depth estimation based on deep learning: An overview. Science China Technological Sciences , 1–16.

List of Figures

Figure 1: Illustration of an image from the ROll and PItch at Sea (ROPIS) dataset.

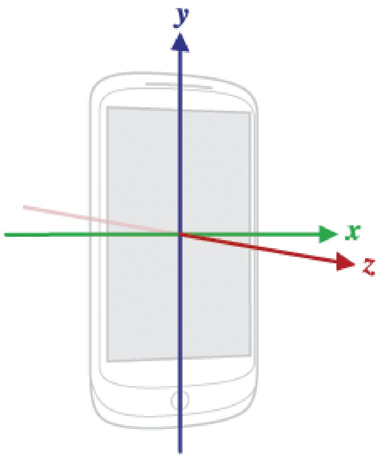


Figure 2: Device coordinate system used by the Android Sensor API (Android).

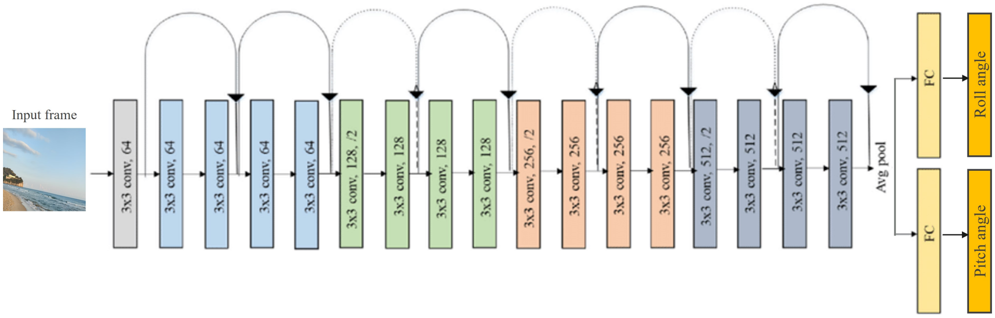
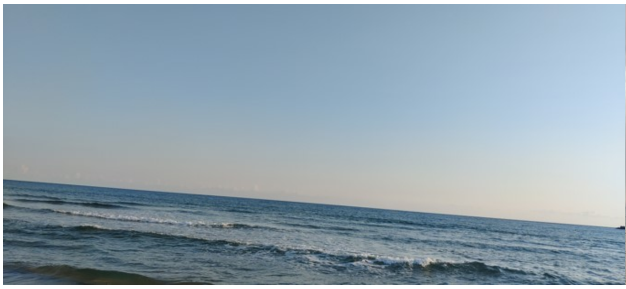


Figure 3: DOES architecture with default ResNet18 backbone network.

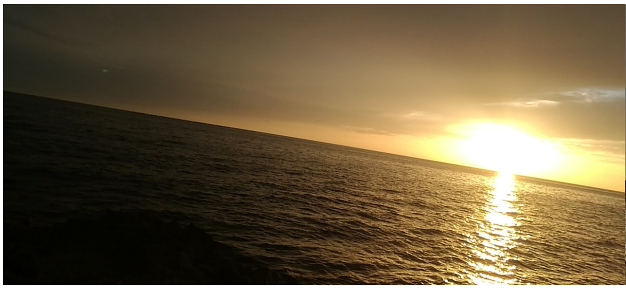


(a) Gaeta_Serapo_sunny subset, Frame 10_2021-04-20_185520.jpg

(b) Gaeta_Serapo_cloudy subset, Frame 282_2021-04-26_184257.jpg

(c) Gaeta_Harbour_cloudy subset, Frame 39_2021-04-26_181529.jpg

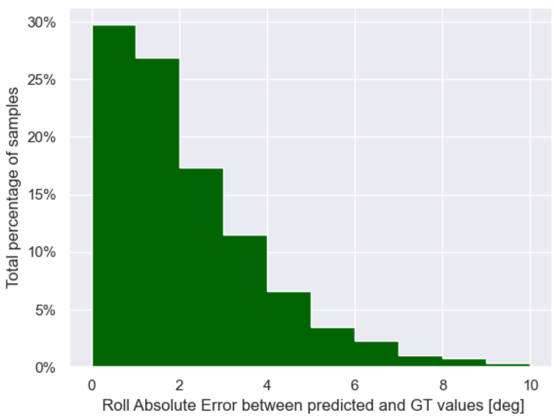
(d) Gaeta_City_cloudy subset, Frame 537_2021-04-26_175240.jpg



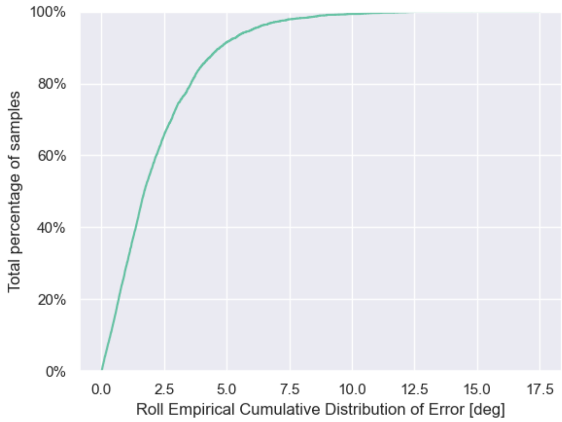
(e) Racale_sunset subset, Frame 80_2021-04-26_181710.jpg

(f) Gaeta_S.Agostino_sunny subset, Frame 8_2021-05-03_173835.jpg

Figure 4: ROPIS dataset samples. Figures 4a to 4e belong to the training set, Figure 4f to the test set.

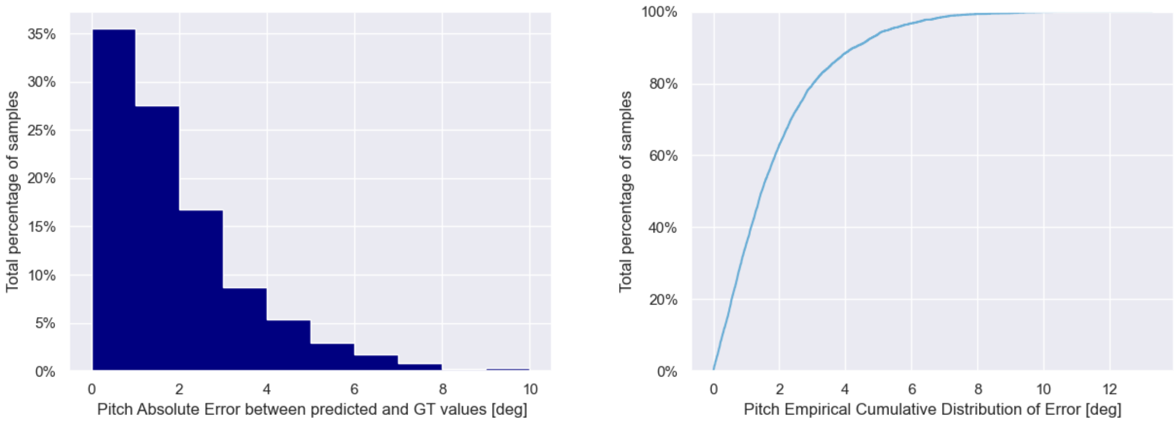


(a) Roll Absolute Error



(b) Roll Empirical Cumulative Distribution

Figure 5: Graphical distribution of the errors for the estimation of the Roll angle.



(a) Pitch Absolute Error

(b) Pitch Empirical Cumulative Distribution

Figure 6: Graphical distribution of the errors for the estimation of the pitch angle.

Figure 7: A frame from the low-light condition separated set.