SAMPLE-TUBE POSE ESTIMATION BASED ON TWO-STAGE APPROACH FOR FETCHING ON MARS

Israel Raul Tiñini Alvarez[1], Ignacio Amat Pérez[1], Tim Wiese[2], Laura Bielenberg[2], and Renaud Detry[1]

[1]*KU Leuven, Leuven, Belgium*
[2]*ESA/ESTEC, Noordwijk, The Netherlands*
{*israelraul.tininialvarez, ignacio.amatperez*}*@student.kuleuven.be*
tim.wiese@ext.esa.int
laura.bielenberg@esa.int
renaud.detry@kuleuven.be

ABSTRACT

As part of a future mission to Mars currently studied by NASA and ESA, the Mars Sample Return campaign aims to bring back the sample tubes collected by the Perseverance rover. In this paper, we propose a two-stage approach for estimating the pose of the sample tubes deposited on the Martian surface. In the first stage, keypoints are obtained using data-driven techniques; then, PnP combined with RANSAC is used to obtain the translation and rotation of the tubes. In this work, the results on the first stage are reported. Specifically, three representations are implemented and evaluated to localize 2D keypoints on the tubes for a later matching with their corresponding 3D coordinates to obtain their pose. The solution was trained and evaluated using a dataset collected at the NASA's Jet Propulsion Laboratory.

Key words: MSR, object pose estimation, keypoint estimation.

1. INTRODUCTION

As identified by the Planetary Science Decadal Survey in 2011 [1], the Mars Sample Return (MSR) is a high priority long-term goal for NASA. This MSR campaign is based on a 3 mission concept [2]. The first mission, currently ongoing by NASA's Perseverance rover, consists of collecting rock samples and storing them in sealed tubes, which will be left on the surface of Mars for a future mission to return them to Earth. The second mission, a Sample Retrieval Lander (SRL), would collect the sample tubes and load them into an Orbiting Sample (OS) payload in a Mars Ascent Vehicle (MAV). The MAV would release the OS into Martian orbit. For the last mission, a Sample Return Orbiter (SRO) would retrieve the OS in Martian orbit and come back to Earth.

This work focuses on the second mission, in which the SRL would collect the tube samples. In detail, the SRL would be comprised of a Sample Fetch Rover (SFR) provided by ESA, a Sample Transfer Arm (STA), the previously mentioned OS and MAV. From these, the SFR, see Figure 1, is meant to collect the samples autonomously in a short period [3]. Therefore, it must be capable of efficiently and accurately estimating the tubes' pose on the Martian surface. On top of that, tube detection must be robust to dust, shadows, poor lighting conditions, diverse terrain, and possible occlusions, among other challenges [4]. Moreover, the nature of the problem implies a limitation in the resources such as memory, CPU, and GPU. Finally, tube detection is required to work on the rover's stereo navigation cameras, but also on a monocular wrist camera mounted on a robotic arm to pick up the tubes. Therefore, the vision system must be capable of inferring tube position and orientation by using monocular images only.

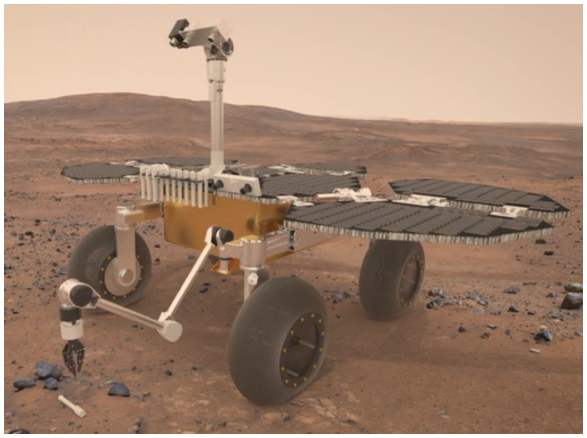

Figure 1. Illustration of ESA's SFR rover collecting a sample tube

Because the estimation of the pose has been widely studied, the proposed framework is based on state-of-the-art techniques for object pose estimation (OPE). OPE is a problem from the image processing and computer vision field that aims to infer the relative position and orientation of an object using as input either grayscale, color, or/and depth information. For achieving this objective, throughout the years both traditional and data-driven techniques have been proposed. In fact, in the last years, OPE has

grown due to its multiple applications such as augmented reality, self-driving cars, and robotics [5].

Although the classical image processing algorithms for this task are mature and transparent, their reliance on fixed matching procedures and handcrafted features limit their performance. On the other hand, deep-learning-based techniques offer higher accuracy and more versatility. Nonetheless, their drawbacks regard the consumption of high computing resources and the difficulty to interpret their decisions. Therefore, mixing the two techniques has been found to be the recommendable approach for accurate, lightweight, and predictable systems.

As previously mentioned, memory and computational power are limited; thus, lightweight solutions are needed. Because of the tubes' symmetry, it is not necessary to get the full 6D pose. However, this represents an additional challenge when defining keypoints to be easily distinguishable. On the other side, we count with prior knowledge about the dimensions of the tubes and the number of classes (just sample tubes). All these restrictions and problem-specific peculiarities were considered in the proposed solution.

In this work, we propose to estimate the full pose of the sample tubes using monocular RGB images based on a two-stage approach designed to fit the limited memory and computational resources. For achieving this, first, the positions of the tubes are obtained using a modified off-the-shelf object detector. Next, a set of keypoints of the obtained region of interest in the previous step, are predicted using an image regressor. Finally, the pose is obtained using a traditional method. Our contributions are as follows:

- We propose a new representation for encoding the position of keypoints that aims to improve the keypoint detection.
- A two-stage approach for sample tube pose estimation is defined by combining deep-learning techniques for keypoint estimation and traditional methods for pose computation.
- An experimental evaluation of the proposed representations and framework is provided, which aims to find the most suitable architecture to meet the requirements and constraints.

The remainder of the paper is organized as follows: Section 2 reviews previous works. Section 3 describes the proposed method. Experiments and results are presented in Section 4. Next, in Section 5 we present our conclusions. Finally, in Section 6 the next steps and future work are defined.

2. RELATED WORKS

Regarding the problem of object pose estimation, from a technical point of view, the proposed state-of-the-art solutions can be divided into **1) Traditional approach**: Geometry-driven approach in which the pose estimation is computed with classic methods based on handcrafted features. **2) End-to-end approach**: Data-driven approach in which the detection and pose estimation steps are done together in a single forward pass through a neural network. **3) Two-stage approach**: First regress 2D keypoints using deep learning techniques, then infer 6D pose parameters by mapping the relationship of 2D to 3D coordinates. From there, the proposed framework belongs to the two-stage approach since we obtain first a set of custom keypoints which are used to predict the pose by matching them to their corresponding 3D points, see Section 3.

2.1. Traditional techniques

Traditional solutions can be classified into **1) Point-pair based features**: A model description is created based on global oriented point pair features and matches them locally with a Hough voting scheme such as in [6; 7]. **2) Template-based**: Match the input image and the template to obtain the 6D pose of the matched template as the pose estimation result. An example of this technique is [8] where gradient orientations and/or surface normals are used to obtain a new binary representation and capture the appearance of the object in a set of templates covering different views. **3) 3D local features**: 3D features are extracted from the RGB-D images and classified to estimate the 6D pose, such as in [9].

Although traditional methods based on the previously mentioned techniques have had considerable success in the computer vision field, the reliance on fixed matching procedures and handcrafted features limit their performance. Moreover, they have difficulty handling textureless objects as well as processing low-resolution images [5]. Therefore, alternative solutions were explored that better suit our problem, because the image of the tube has low resolution and is faintly textureless.

2.2. Data-driven techniques

Several end-to-end techniques have been proposed to directly predict the 6D object pose from monocular RGB images. BB8 [10], is a method that first identifies the 2D objects using a segmentation model in a two-level coarse-to-fine manner to find the centers of the objects and then applies a Deep Network based on [11] to estimate 3D points which are then matched via PnP. SilhoNet [12], a framework that first creates feature maps concatenated with features from a set of rendered object viewpoints to obtain a 3D vector of the center of the object and combine it with an L2-normalized quaternion to get the 6D pose.

In the last years, more sophisticated solutions have been proposed. For example, EfficientPose [13] is an extension of the 2D detector EfficientNet [14], with two extra subnetworks on top of it to predict the rotation and translation of one or more instances. RePose [15], a fast iterative refinement method based on an encoder-decoder architecture. Here, the encoder uses pre-trained weights and the decoder is trained on the Levenberg-Marquardt optimization [16] with the output features from a deep texture rendered of the template 3D model and the output

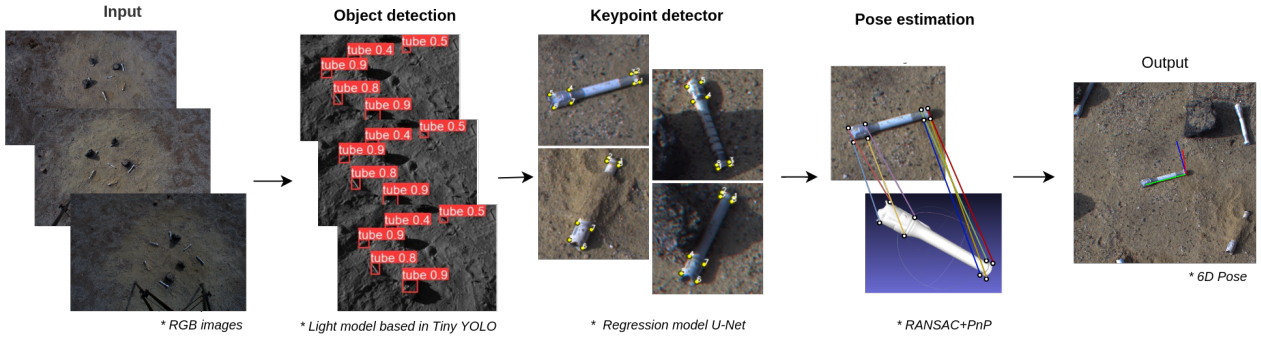

Figure 2. Proposed framework

of a U-Net from the RGB image to obtain the final pose after a defined number of iterations.

The classical computer vision algorithms for pose estimation are transparent and easy to verify. However, deep-learning approaches offer better performance and versatility since they do not rely on handcrafted features. Nevertheless, the main drawback of these is their black-box natures, making them difficult to validate. Moreover, they also require thousands of examples to train and tend to consume more computing resources. Therefore, mixing these two technologies is recommendable for accurate and high-performance systems [17].

2.3. Two-stage techniques

In Pavlakos et al. [18] and Park et al. [19] the authors propose a pipeline that includes object detection, keypoint localization, and pose optimization. [18] use a Stacked hourglass CNN to predict a set of semantic keypoints represented as heatmaps. Then, the pose is estimated by maximizing the geometric consistency between a parametrized deformable model and the 2D keypoints. On the other hand, Pix2Pose[19] proposes an autoencoder architecture designed to estimate the 3D coordinates per pixel.

Tekin et al.[5] propose to predict 2D projections of the corners of the 3D bounding box around our objects. For obtaining the 2D projections, they use a modified version of YOLOv2 [20]. Zakharov et al. in DPOD [21] estimate dense 2D-3D correspondences between an RGB input image and the object 3D models using auto-encoders [22]. Given the 2D-3D correspondences, [5; 21; 19] use PnP combined with a RANSAC scheme algorithm to obtain the pose of the objects.

Many recent works have shown that a two-stage approach, which first detects keypoints and then solves a PnP problem for pose estimation, achieves remarkable performance [17; 23]. Therefore, our solution is based on this approach.

3. PROPOSED APPROACH

In this work, we focus on solving the problem of sample-tube pose estimation using a two-stage approach. Indeed,

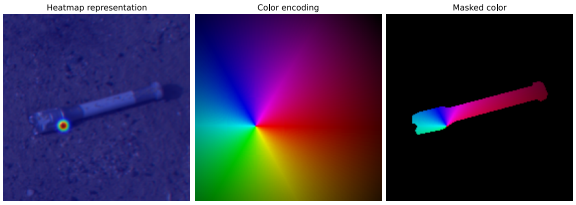

Figure 3. Keypoint representation encoding for the 7th keypoint

to the best of our knowledge, we found that deep learning combined with traditional methods produces better results as stated in [17]. The proposed framework, see Fig. 2, combines data-driven and traditional techniques. We first introduce the object detector, which obtains the position of the tubes. Having the localization of the tubes in the image, a cropped ROI is then passed to the next stage. In the keypoint detector, a set of predefined keypoints are identified through a color encoding scheme. Finally, the pose is estimated using a PnP algorithm in combination with a RANSAC voting scheme. The last stage is not part of this work but is aimed to be presented in a future paper.

3.1. Sample tube detection

Inspired by [24], as a first stage, we aim to reduce the searching space, processing time, and computational power by localizing the object(s) of interest visible in the input image. During the last decade, various CNN [25] algorithms have been proposed to tackle the task of object detection, such as the family of region proposal methods R-CNN [26], Fast R-CNN [26] or Faster R-CNN [27], Mask R-CNN [28] along with the family of YOLO algorithms [29; 20; 30] or other methods such as SSD [31]. Motivated by [32; 33; 34; 35], we decided to choose an object detector based on YOLOv3 as it has been proven to have an overall good accuracy, and it is faster compared with the other methods. However, the accuracy of YOLOv3 is not the best, but to detect the sample tube coarsely, we can tolerate these small detection errors as the idea of using this detector first is to reduce the size of the image to just the ROI of the detected object and pass it to the next stage.

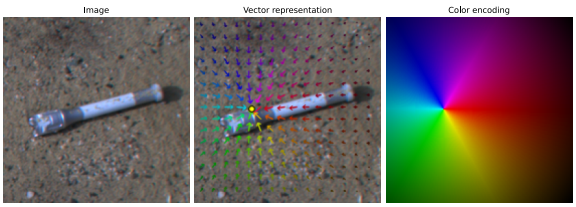

Figure 4. Color encoding for a single keypoint

3.2. Keypoint definition and representation

Following the two-stage approach, given the cropped region of a sample tube, the objective is to regress the position of a set of keypoints to estimate the pose of the object. Thus, we defined a set of keypoints as shown in **keypoint detector** in Fig. 2. In total, eight keypoint were defined in [23]. In a nutshell, a keypoint is a point easily distinguishable, so we defined the most prominent corners in an anti-clockwise direction as our keypoints.

Having the keypoint defined, next, a suitable representation is needed. There are mainly two options to represent the localization of the keypoints: sparse and dense representation. The former represents the keypoints as a simple 3D bounding box [5; 10] or custom coordinates [18]. On the other hand, by dense representation, we refer to encode information about the keypoint localization in each pixel [23; 21; 19].

To obtain the best performance, three representations for encoding the localization of the keypoints are evaluated: using heatmaps, color encoding, and masked color encoding. Fig. 3 shows the three representations for a single keypoint. The heatmap is obtained by placing a 2D Gaussian distribution with the peak at the location of the keypoint. In the color encoding, each pixel in the representation can be interpreted as a vector pointing to a certain keypoint. Finally, the masked color encoding is obtained by combining a binary segmentation mask of the tube with its color-coding.

Because numerous correspondences are beneficial for obtaining high-quality 6D poses [21], we expect that a dense representation will outperform the sparse one. Our color representation is inspired by the vector-field representation from [23] since it is reported to be robust against occlusions. Therefore, because we expect to face challenges such as shadows, occlusions due to dust covering, and poor lighting conditions; by using this representation, we aim to have a strong representation for encoding the localization of the keypoints.

For obtaining the color encoding. 1) We compute the horizontal and vertical distances of each pixel w.r.t the location of a keypoint. 2) Using these distances, the magnitude and orientation of vectors pointing to the keypoint per each pixel are obtained. 3) The orientations are encoded in a color format using the Hue value in an HSV color space. At the same time, the magnitude of the vectors is encoded as the Value. 4) The HSV images are transformed to RGB for visualization purposes. A representation of this process is shown in Fig. 4

3.3. Keypoint estimation

Having the representations of the keypoint defined, we propose to regress the keypoint representations and segmentation masks of the tubes using an architecture based on a U-Net. We adopted this architecture since it only needs a few annotated images and has a very reasonable training time [36]. Moreover, it has been proved to perform well in segmentation tasks where fine predictions are required. Unlike other previous solutions [21] which use an encoder-decoder network as a feature extractor, the U-Net skip connections concatenate the activations from the encoder to the decoder, which duplicates the number of channels in the decoder. This action helped us to relieve the bottleneck problem and reduce the loss and improve the quality of the results. The architecture of the model is shown in Fig. 6.

3.4. Estimation of the pose

Once the localization of the keypoints is obtained, the next stage is to compute the pose of the tubes. For achieving this, we decided to use the traditional Perspective-n-Point (PnP) algorithm in combination with a RANSAC voting scheme. This combination is popular in the state-of-the-art since it is computationally fast and obtains acceptable results. The PnP matches the 2D localization of the keypoints with their 3D correspondences to obtain the rotation and translation vectors, i.e. the 6D pose of the object. Instead of using a physical object of the sample tube, a 3D rendering [37] was used instead for obtaining the measures of the 3D coordinates.

4. EXPERIMENTS AND RESULTS

In this section, we introduce the dataset used in the experiments. Next, the parameter and training schemes are detailed. Finally, we compare the performance of the proposed representations and obtain the predicted keypoints.

4.1. Dataset

In this work, we used a dataset collected at the NASA's Jet Propulsion Laboratory (JPL) [4] to evaluate our model. This dataset composed of outdoor images was acquired at JPL's Mars Yard to simulate the environmental conditions that the SFR will face on Mars. These diverse conditions include different terrain formations, different lighting conditions, and object occlusions provoked by rocks or dust covering (with different amounts of coverage). The images were acquired using four FLIR BlackFly S cameras, forming two stereo pairs with baselines of 20cm and 40cm. The cameras were placed at two different heights of 1m and 2m trying to replicate the Perseverance rover's onboard cameras (HazCam and NavCam). The dataset comprises RGB images of 5472×3648px, ground truth segmentation masks with their associated bounding boxes, and rotation and translation information for a small portion of samples. The dataset in total contains 824 images with 4852 annotated instances of tubes, out of which 256 of those have an associated 6D pose.

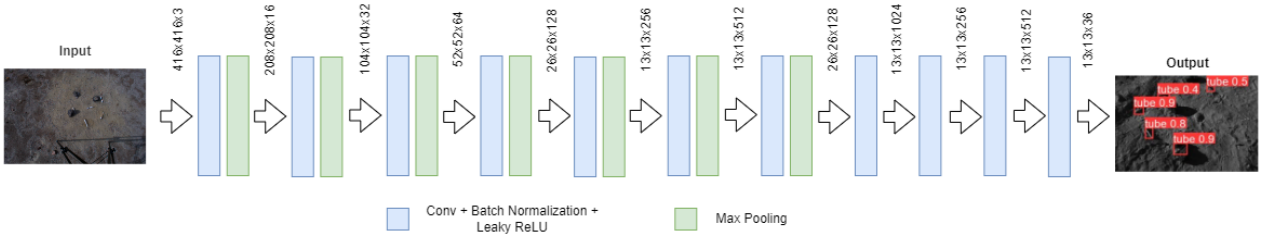

Figure 5. Sample tube detection architecture.

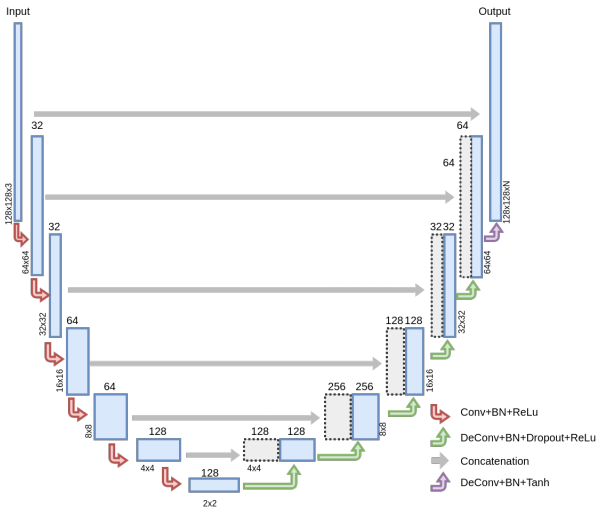

Figure 6. Keypoint detector architecture.

As previously mentioned, the dataset used for the experiments comprises a big subset of instances annotated with a segmentation mask along with a bounding box, and a small subset with an additional annotation of translation and rotation. For training the object detection models, the first set was enough. However, because of the reduced size of the samples annotated with their pose, an additional manual annotation stage was required where instead of directly labeling the pose of the tubes, the localization of the defined keypoints and the dimensions of a 3D model of the tube were enough to obtain an estimation of the pose using the pose estimation algorithms mentioned before. After this process, 3081 samples were collected.

4.2. Model parameters

The architecture proposed for the object detector is based on YOLO [29], an architecture mainly built with modules of a Convolutional layer, a Batch Normalization layer, a Leaky ReLU activation function, and a Max Pooling layer. Thus, if we denote `CkBM` to denote each complete module where Convolution-BatchNorm-LeakyReLU-MaxPool are applied, `CkB` to denote each block where Convolution-BatchNorm-LeakyReLU are applied and `Ck` to denote each block where a Convolution layer and a Linear activation function are applied, with k representing the number of filters for the Convolutional layer, then the resulting architecture has the following structure: `C16BM-C32BM-C64BM--C128BM-C256BM-C512BM-C1024B-C256B--C512B-C36`. Additionally, kernels of size 3x3 were used in every Convolutional layer except in `C256B` and `C36` where a kernel of size 1 was used. Kernels of size 2X2 with a stride of 2 were used in all the Max Pooling layers except on the last one where a stride of 1 was used. Lastly, the Leaky ReLUs activation functions were implemented with a slope of 0.1.

The architecture proposed for the keypoint detector is based on a U-Net [36]. The U-Net can be divided into two main parts: an encoder (E) and a decoder (D). In the former, the convolutions downsample the feature maps by a factor of 2. In contrast, the decoder upsample them by the same factor. In our model, the encoder and decoder use modules of the form: `(De)Convolution - Batch Normalization - ReLu`. Thus, to denote these modules, `Ck` will represent a module composed of Conv-BatchNorm-ReLU with k filters. Whereas `CDk` states for a module of DeConv-BatchNorm-ReLU with k layers and a dropout of 0.5. Moreover, filters of 4x4 with a stride of 2 were used in all (de)convolutions layers. Lastly, we implemented Leaky ReLUs in the encoder layers with a slope of 0.2, whilst in the decoder network, simple ReLUs were used. We trained a network with the next structure: `E: C16-C32-C64-C128 D: CD64-CD32-CD16`.

4.3. Training details

The object detector network was trained using pre-trained weights from the COCO dataset [38], and thus transfer learning was applied to retrain the network with the images from the JPL dataset. As for the optimization function, Adam optimizator was employed, with a learning rate of 0.003. A loss function based on Mean Squared Error (MSE) and Cross Entropy was employed to obtain optimal results, as for this task we are only dealing with one class to be identified, Binary Cross Entropy turned out to be more suitable. Notice that, all images were resized to 416x416 to be fed to the network.

All networks for the keypoint detector were trained from scratch, so the weights were initialized from a Gaussian distribution with a mean of 0 and a standard deviation of 0.02. Because in all the representations, we needed to obtain continuous values, bitwise MSE was used as a loss function with Adam as an optimizer, with a learning rate of 0.0002 and momentum parameters of $\beta 1 = 0.5$, $\beta 2 = 0.999$. However, since the number of samples is limited, data augmentation was necessary. For this purpose, the random jitter technique was applied before the training

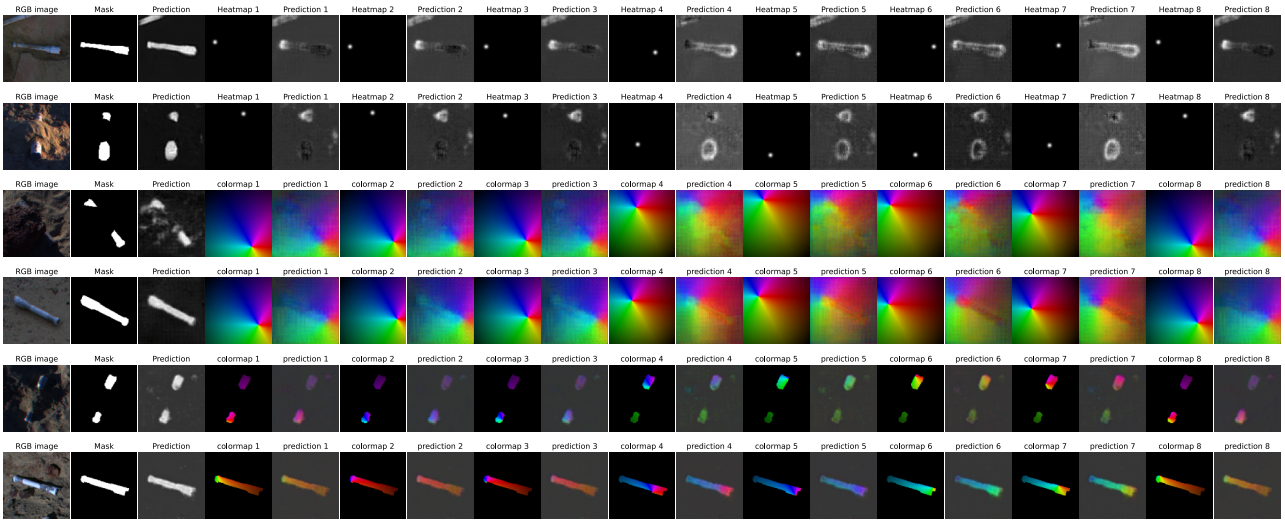

Figure 7. Prediction on a sample of the test set for the proposed representation. Each two rows correspond to the heatmap encoding, color coded, and color masked encoding approach, respectively

stage. All the representations were resized from 128×128 to 153×153 and then randomly cropped back to their original size. We noticed that a stable performance was achieved after 50 epochs of training.

The different configurations of our models were implemented in Python using Keras, TensorFlow, and Pytorch framework. The training and test stages were performed using the Google Colab tool. The virtual machine used in the experiments had a 2.3 GHz dual-core processor, 25 GB of RAM, an NVIDIA Tesla K80 graphic card with 12 GB of memory, and 2496 CUDA cores.

4.4. Results and analysis

Regarding the performance of our object detector, we noticed that after 82 epochs there was no more improvement of the model. Therefore, early stopping during training was needed to avoid overfitting in the training data. We then calculated the Precision-Recall curve for the object detector, with an average precision (AP) of 0.807. This value indicates that the model has high precision and a high recall, which means that there is a low false-positive rate meaning that there will be a low rate of detecting tubes where they are not as well as a low false-negative rate meaning that there will be a low rate of missing a tube. Although the average precision is good and our approach performs pretty well in general, there are some cases where the detection is performed poorly or completely missed when for example the tube is mainly occluded and only a small part of it is visible. As the aim of this object detector is to be able to crop the ROI of the bounding box of the tube, we consider that these results are still good for this purpose as in the next step of the process the keypoint predictor will refine this work.

To demonstrate the performance of our keypoint detector, first, we present the qualitative results for the three proposed representations in Fig. 7. In the first column, the input image is plotted, in the next columns, the ground truth and predictions are shown next to each other for the

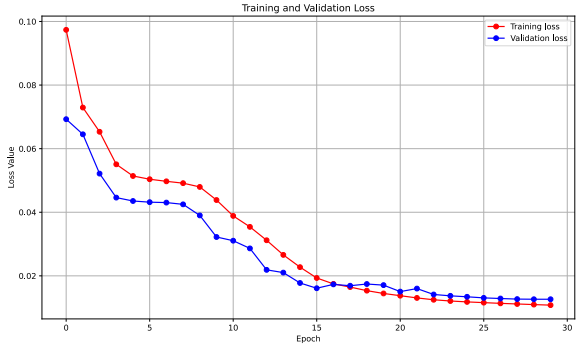

Figure 8. Training and validation loss

segmentation mask and the eight keypoints. In total, three models were trained, and the predictions for two test samples for one model are shown in the figure.

The first two rows of Fig. 7 represent the prediction of the U-Net for the heatmap encoding. As it can be seen, the model do not learn effectively the task since multiple areas are highlighted. Nevertheless, it is important to note that the output has some relation w.r.t. expected prediction since they are located in a similar region.

On the other hand, the model effectively maps the dense representations encoded in color when using the color encoding. This is less evident when using the masked color representation. This can be corroborated by comparing the label with the prediction color tone. Moreover, notice that models can cope with challenging conditions. The regressor is capable of identifying the pose of the encoding of the tubes, even under occlusions and shadows.

From these results, we noticed that the best performance was achieved when using color-coding. Therefore, we continued the experiments using this color-coding representation. Fig. 8 shows the loss during the training for the training and test set. Notice that after 25 epochs, the learning stops improving. Nevertheless, due to the reduced model, and data augmentation the model does not

Figure 9. Predictions for the 8 keypoints.

overfit.

Having the predictions for the keypoints, we obtained the position of the keypoints using the color information. In Fig 9, we present some obtained results, where the keypoints are marked in a cyan and enumerated. Notice that, the model effectively encodes the information about keypoints $4, 5, 6, 7$, and places them close to the correct location; however, it fails to distinguish between keypoints $1, 2, 3, 8$, and places them in a similar position.

From the predicted locations, we noticed that the sparsity of the keypoints helps the model to generalize and correctly predict the location. Indeed, points corresponding to the head of the tube were correctly localized in most of the samples with some small displacement, which can easily be corrected by modifying their locations using the predicted binary mask i.e. by moving keypoints to the close boundary point. Nevertheless, the 6D pose estimation was not correctly obtained due to the misplaced predicted points in the tail of the tube. Therefore, a new annotation scheme is needed considering the findings in this work.

5. CONCLUSIONS

In this paper, we propose to use a two-stage approach for estimating the pose of sample tube using monocular color images. The solution is aimed to overcome variations due to environmental challenges and fit the requirements. Since the sample tube pose estimation is a complex task where the amount of information and resources are limited, we propose to use a two-stage approach combining data-driven and traditional models. First, an object detector obtains the ROI of the tubes. Then, a set of keypoints are regressed using a color-coding scheme. Three encoding schemes are proposed, from which, through experiments, we have seen that color encoding obtained better results. Because of the wrong prediction of some keypoints the pose of the tubes was not correctly obtained. However, this could be solved by defining a new set of locations for the keypoints based on the outcomes of this work.

6. FUTURE WORK

In the future, we will complete the proposed framework by implementing the pose estimator. Moreover, we will collect a dataset with a setting similar to the navigation and wrist camera that will be mounted on the SFR. Moreover, the new dataset is aimed to be as close as possible to the expected scenario. Furthermore, we are going to improve the performance of the object detector and the keypoint estimator.

ACKNOWLEDGMENTS

The authors would like to thank NASA's JPL for supplying the dataset used for experiments and evaluations. This project is a collaboration between ESA's Automation and Robotics Section and KU Leuven.

REFERENCES

[1] N. R. Council, *Vision and Voyages for Planetary Science in the Decade 2013-2022*. Washington, DC: The National Academies Press, 2011.

[2] B. Muirhead, A. Nicholas, J. Umland, O. Sutherland, and S. Vijendran, "Mars sample return mission concept status," *Acta Astronautica*, vol. 176, 06 2020.

[3] J. Papon, R. Detry, P. Vieira, S. Brooks, T. Srinivasan, A. Peterson, and E. Kulczycki, "Martian Fetch: Finding and retrieving sample-tubes on the surface of mars," 2017.

[4] S. Daftry, B. Ridge, W. Seto, T. H. Pham, P. Ilhardt, G. Maggiolino, M. Van Der Merwe, A. Brinkman, J. Mayo, E. Kulczyski, and R. Detry, "Machine Vision based Sample-Tube Localization for Mars Sample Return," vol. 2021-March, 2021.

[5] B. Tekin, S. N. Sinha, and P. Fua, "Real-Time Seamless Single Shot 6D Object Pose Prediction," 2018.

[6] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," pp. 998–1005, 07 2010.

[7] T. Birdal and S. Ilic, "Point pair features based object detection and pose estimation revisited," in *2015 International Conference on 3D Vision*, pp. 527–535, 2015.

[8] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, V. Lepetit, and b. Lepetit, "Gradient response maps for real-time detection of texture-less objects," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 1, 05 2012.

[9] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *ECCV*, 2014.

[10] V. Lepetit, "BB8 : A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3828–3836, 2017.

[11] A. Crivellaro, M. Rad, Y. Verdie, K. Yi, P. Fua, and V. Lepetit, "A novel representation of parts for accurate 3d object detection and tracking in monocular images," pp. 4391–4399, 12 2015.

[12] G. Billings and M. Johnson-Roberson, "SilhoNet: An RGB Method for 6D Object Pose Estimation," vol. 4, 2019.

[13] Y. Bukschat and M. Vetter, "EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach," 2020.

[14] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *ArXiv*, vol. abs/1905.11946, 2019.

[15] S. Iwase, X. Liu, R. Khirodkar, R. Yokota, and K. M. Kitani, "RePOSE: Fast 6D Object Pose Refinement via Deep Texture Rendering," 2021.

[16] J. J. Moré, "The levenberg-marquardt algorithm: Implementation and theory," in *Numerical Analysis* (G. A. Watson, ed.), (Berlin, Heidelberg), pp. 105–116, Springer Berlin Heidelberg, 1978.

[17] J. Chen, L. Zhang, Y. Liu, and C. Xu, "Survey on 6D Pose Estimation of Rigid Object," vol. 2020-July, 2020.

[18] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," 2017.

[19] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6D pose estimation," vol. 2019-Octob, 2019.

[20] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, 2017.

[21] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," vol. 2019-Octob, 2019.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," 1986.

[23] S. Peng, X. Zhou, Y. Liu, H. Lin, Q. Huang, and H. Bao, "PVNet: Pixel-wise Voting Network for 6DoF Object Pose Estimation," 2020.

[24] Z. Li, G. Wang, and X. Ji, "CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, pp. 7677–7686, 2019.

[25] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.

[26] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[28] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[29] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[30] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 04 2018.

[31] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015.

[32] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car detection using unmanned aerial vehicles : Comparison between faster r-cnn and yolov 3 conference paper," 2019.

[33] M. Li, Z. Zhang, L. Lei, X. Wang, and X. Guo, "Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster r-cnn, yolo v3 and ssd," *Sensors*, vol. 20, no. 17, 2020.

[34] K. Zhao and X. Ren, "Small aircraft detection in remote sensing images based on YOLOv3," *IOP Conference Series: Materials Science and Engineering*, vol. 533, p. 012056, may 2019.

[35] M. G. Dorrer and A. E. Tolmacheva, "Comparison of the YOLOv3 and mask r-CNN architectures' efficiency in the smart refrigerator's computer vision," *Journal of Physics: Conference Series*, vol. 1679, p. 042022, nov 2020.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[37] "Mars 2020 sample tube 3d print files." https://nasa3d.arc.nasa.gov/detail/Mars-2020-Sample-Tube-3D-print-files. Accessed: 2022-04-30.

[38] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.