Tracking 3D Deformable Objects in Real Time

Tiago Silva[1], Luís Magalhães[1], Manuel Ferreira[2], Salik Ram Khanal[2] and Jorge Silva[2]
[1]*Centro ALGORITMI, University of Minho, Guimarães, Portugal*
[2]*Neadvance, Braga, Portugal*

Keywords: Deep Learning, 3D Tracking, Deformable Objects, RGB-D Data, Object Segmentation.

Abstract: 3D object tracking is a topic that has been widely studied for several years. Although there are already several robust solutions for tracking rigid objects, when it comes to deformable objects the problem increases in complexity. In recent years, there has been an increase in the use of Machine / Deep Learning techniques to solve problems in computer vision, including 3D object tracking. On the other hand, several low-cost devices (like Kinect) have appeared that allow obtaining RGB-D images, which, in addition to colour information, contain depth information. In this paper is proposed a 3D tracking approach for deformable objects that use Machine / Deep Learning techniques and have RGB-D images as input. Furthermore, our approach implements a tracking algorithm, increasing the object segmentation performance towards real time. Our tests were performed on a dataset acquired by ourselves and have obtained satisfactory results for the segmentation of the deformable object.

1 INTRODUCTION

The problems regarding 3D Tracking on deformable objects are challenging and only recently some research work has been done on the subject. Regarding tracking objects topic, the research work is quite extensive, but relatively to deformable objects there are only a few because its difficulty increases considerably(Hu et al., 2019). Recently, low-cost devices (like Kinect) have emerged that allow obtaining RGB-D images, which, in addition to color information, contain depth information. The possibility of obtaining depth information can be an added value in carrying out tracking of deformable objects because, unlike rigid objects, deformable objects change shape according to their manipulation, making it difficult to carry out. To deal with this problem, machine learning techniques may be used, specifically deep learning, which are more robust in conjunction with the object's depth information, increasing its precision(Song & Xiao, 2013). Furthermore, the 3D Tracking of deformable objects will necessarily have to be performed in real time, raising problems such as error tolerance or computational cost, since the speed of the process becomes essential for its development.

The 3D Tracking increases the complexity of the tracking task because the level of information to be calculated is higher. Nevertheless, it has been studied recently by researchers for deformable objects because the depth information helps in the deformable object tracking.

By using the depth information it is possible to obtain the object information in three dimensions and with this, the Artificial Intelligence model manages to obtain a better perception of the object, with superior results, compared to a system with only the color information (RGB)(Lai et al., 2011).

Object segmentation using CNNs has been investigated in several studies and often segmentation and tracking are performed simultaneously. Wang et al. (2018) created a method named “SiamMask” that improves the offline training procedure of popular approaches to track objects. The ”SiamMask” algorithm is quite popular when we want to use the tracking and segmentation approach simultaneously. Furthermore, its greatest feature is its high velocity, where it obtained the highest velocity for objects segmentation in video, in the DAVIS-2016 (Perazzi et al., 2016) and DAVIS-2017 (Pont-Tuset et al., 2018) competition. Its accuracy is also quite competitive compared to other tracking algorithms.

Voigtlaender et al. (2019) created the algorithm "TrackR-CNN" which is able to track multiple objects, contrary to the previous algorithm” SiamMask”, which is exclusive for cases of tracking

Silva, T., Magalhães, L., Ferreira, M., Khanal, S. and Silva, J.
Tracking 3D Deformable Objects in Real Time.
DOI: 10.5220/0010806700003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 4: VISAPP*, pages 823-830
ISBN: 978-989-758-555-5; ISSN: 2184-4321

only one object. Although it was created only as a basis for the Multi Object Tracking and Segmentation (MOTS) challenge (Voigtlaender et al., 2019), its performance proved to be really effective for the tracking and segmentation process.

Ronneberger et al. (2015) created a Deep Learning architecture called "U-Net". U-Net has been studied a lot due to its success for image segmentation. The architecture receives a RGB image and returns the binary segmentation of the intended image. The U-NET model architecture consists of using a contracting path and an expansive path through CNN´s. The contracting path consists of a typical CNN´s architecture. In the expansive path, the reverse path is performed, where the output is the binary segmentation of the provided image. The U-NET architecture has been widely used recently due to its low computational cost and the fact that there is not needed to provide many examples for the model successfully identify the object. Thus, the U-NET architecture proved to be an excellent candidate for creating binary object segmentation. Furthermore, Ronneberger et al. (2015) won the ISBI cell tracking challenge 2015, one of the main competitions for evaluating segmentation and tracking models in medical images, thus showing the success of their architecture.

Many studies have been carried out to demonstrate the importance of CNNs in object segmentation. Almost all studies use only RGB images, but recently, approaches have emerged where CNNs also receive depth information to segment the intended object. Liu et al. (2019) presented two approaches to train RGB-D data with CNNs: The first approach is to create two models where RGB data and depth data are trained separately, and the second approach is to create only one trained model with RGB images and depth data. In the first approach, the same importance is given to RGB data and depth data, in the second approach, greater importance is given to RGB data (Liu et al., 2019).

This paper presents a new approach for 3D tracking of deformable objects that uses a deep learning model and RGB-D data as input. The approach was designed and planned aiming real time performance. The approach was tested and evaluated using a dataset built specifically for this purpose.

The rest of the paper is structured as follows: In the second section, we describe our method. In the third section presents our results and discusses the main conclusions of this study. The fourth section concludes the article.

2 PROPOSED APPROACH

This section presents a solution for performing automatic tracking of deformable objects, using Computer Vision algorithms and methods developed specifically for this problem. The development of this solution involves the detection of the deformable object and monitoring its movement over time. Since they were used supervised machine learning/deep learning algorithms, the tracking process is done in two stages: Offline Process (data processing and model training) and Online Process (tracking of the deformable object in real time).

2.1 Offline Process

The offline process (Figure 1) contains the steps necessary to train the models to be used in the tracking algorithm. In an initial step it is necessary to process the data received. It is important to analyse the data provided and divide it into two groups: RGB Data and Depth Data. The two types of data are treated separately and the techniques used in the depth data do not necessarily have to be used in RGB data and vice versa.

In the end, there will be only one model that receives two images (RGB and Depth) and returns an image with the binary segmentation of the intended object. With this architecture it is possible to use the information of Depth data and RGB data simultaneously. Liu et al. (2019) two-model approach was chosen because the depth information is important for the tracking algorithm to understand the shape of the object and, with this architecture, the depth information will have the same importance as the RGB information.

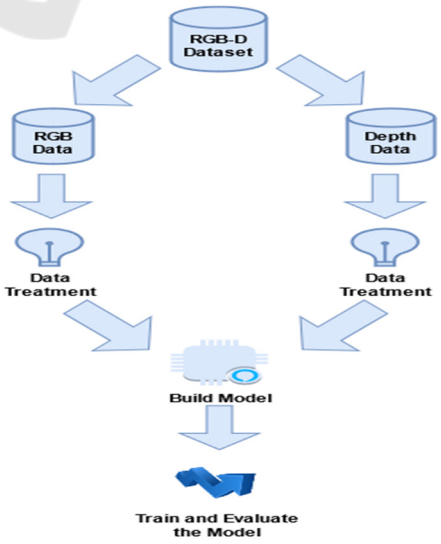

Figure 1: Offline Process.

For the construction of the model, the U-NET (Ronneberger et al., 2015) architecture was used. It was necessary to perform some optimization of the hyperparameters such as the input size. The model receives two groups of data as input: RGB data and Depth data. For each of the groups, a U-Net architecture is built that outputs a layer with a 448 x 448 x 3 format size. After that, it is necessary to perform a concatenate operation that transforms the two outputs into just one layer with a 448 x 448 x 6 format size (with this operation it is possible to use the information from both groups to predict the deformable object). In the end, a Conv 1 x1 is applied that transforms the output into a 448 x 448 x 1 image, as shown in Figure 2.

2.2 Online Process

The Online Process (Figure 3) will contain the necessary steps to track the deformable object. In an initial phase it is necessary to apply some pre-processing techniques to the frame received. RGB and depth data are processed separately and are applied the same pre-processing techniques that were used in the Offline Process. It is important that the frame format is identical to the images trained by the model, that is, the same input size and use the same data pre-processing techniques. In this way, the results will increase considerably. After pre-processing the RGB-D frame, it is necessary to use the tracking algorithm and locate the deformable object. This process is repeated for all frames.

The Tracking Algorithm aims to perform the target object segmentation in a fast and efficient way. In contrast to object detection that predicts on all frames and all frames are independent of each other. The developed Tracking Algorithm uses information from previous frames to perform the deformable object's segmentation more quickly. As the objective is to use the system in real time, the processing velocity is a fundamental point, which is the main reason for using a tracking approach in detriment of object detection. The Tracking Algorithm receives as input the current RGB-D frame and outputs the segmented object and the coordinates of the corresponding Bounding Box. This way, the system will be able to perform the segmentation of the deformable object and follow the object's movement in the next frames.

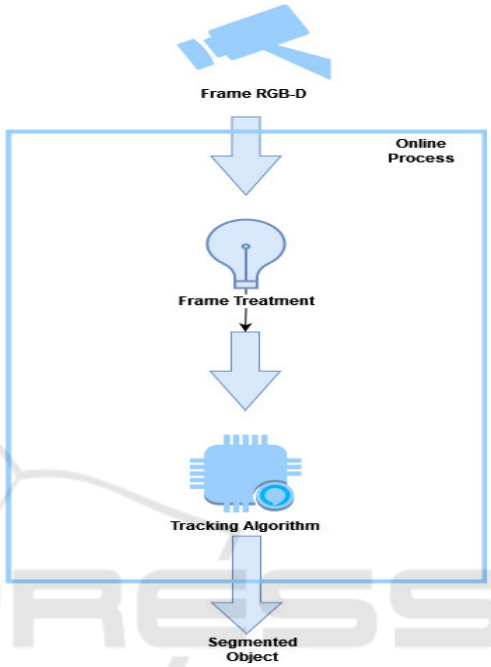

Figure 2: Online Process.

For the construction of the Algorithm, it was necessary to create two prediction models: **Model capable of segmenting the object in a full image** and **Model capable of segmenting the object only in the Bounding Box**. The model capable of segmenting the object in a full image will have the same functionality as a Segmentation model in object detection. The model does not receive any information regarding the previous frames and performs the prediction in the complete frame. In the

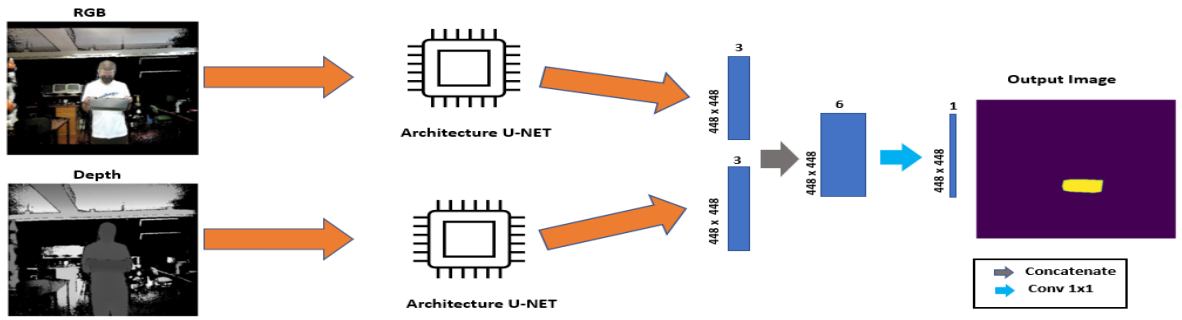

Figure 3: Model RGB-D.

end, the model will provide two pieces of information: The complete image with the segmented object and the Bounding Box corresponding to the object's location. This model will only be used in the first frame because it is not yet known where the object is in the image, and it is necessary to calculate its initial location. The coordinates of the Bounding Box, corresponding to the target object, are given to the next frame. The next frames always receive the coordinates of the Bounding Box of the previous frame and, thus, the model capable of segmenting the object only in the Bounding Box will always be used for those frames. In the end, this model will have to provide two pieces of information: Object segmentation coordinates having as reference the Bounding Box of the previous frame and update the coordinates of the Bounding Box for the current frame.

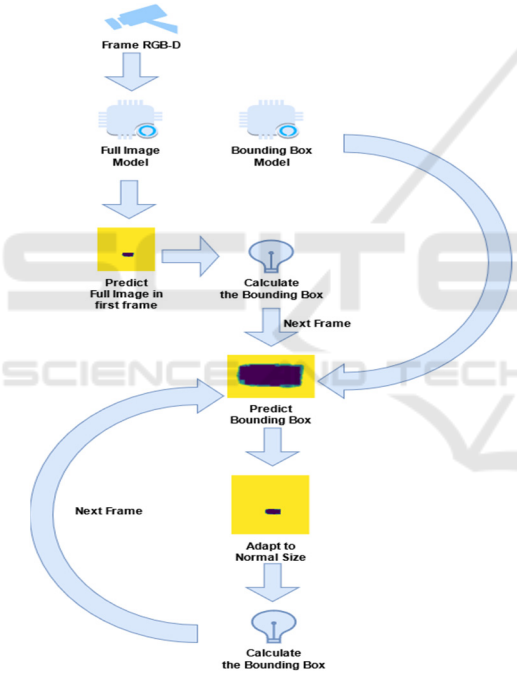

Figure 4: Tracking Algorithm.

Since the object's speed is not too high, the object's location in the previous frame is practically the same as the object's location in the current frame. For this reason, it is possible to use the coordinates of the Bounding Box in the previous frame to locate the object in the current frame. The Tracking Algorithm segments the object only in the Bounding Box, which, computationally, is essential to reduce the frame size supplied to the predict model and, in this way, increase the system response time.

In addition, it is also possible to perform the prediction, in the full image, for certain frames, balancing the tracking quality and processing velocity. This strategy can be used in situations in which the object moves very fast, preventing the object from getting lost. In Figure 4, the Tracking Algorithm schematic is presented.

3 RESULTS AND DISCUSSION

In all tests it was decided to use the U-NET architecture because it obtains better results and converges faster than the implementation of CNN's from scratch. All hyperparameters were similar in all tests to obtain the results with maximum precision. We used Adam as the optimizer (Kingma & Ba, 2017), and a Learning Rate of 0.0004 on all tests. In the Full Image Model, all images were resized to 448x448 pixels and in the Bounding Box Model all images were resized to 128x128 pixels. Furthermore, before the training process, all images were normalized. All results presented by the models are related to test datasets to verify the Model's performance against unfamiliar images. The algorithm was implemented using Python programming language and open-source libraries such as OpenCV, Tensorflow, Numpy, among others. To optimize the processing time, part of the implemented algorithms and methods were executed on a graphics card, using the parallel computing platform CUDA (Compute Unified Device Architecture) and an NVIDIA GeForce RTX 3060 6GB GDDR6 graphics card was used for the training phase.

3.1 Dataset

Data acquisition was obtained in the robotics laboratories at the University of Minho, using a Kinect 1.0 camera where, in addition to the RGB image, it is possible to obtain the depth matrix. The videos were recorded from 6 different points of view to increase the diversity of the dataset, thus transforming it into a more robust dataset. In addition, some videos were made with two different type of movements for detecting leather defects: Movements Type 1 Dataset and Movements Type 2 Dataset. In the Movements Type 1 Dataset, videos were recorded while more smooth movements are performed. In the Movements Type 2 Dataset, videos were recorded while the leather suffers an undulatory movement. In Table 1, it is shown the original size of the two Datasets.

Table 1: Size of Movements type 1 Dataset and Movements type 2 Dataset.

Dataset	RGB Data	Depth Data	Total
Movements Type 1	11549	11549	23098
Movements Type 2	6096	6096	12192

For the training and evaluation of the model, 1765 annotations were made, 1155 in the Movements Type 1 Dataset and 610 in the Movements Type 2 Dataset.

An image was chosen every 10 frames for both datasets to be annotated. To carry out the annotations, Labelme (Russell et al., 2008) was used, which is an open-source tool for graphical annotations of images.

Kinect, for each captured frame, provides three pieces of information: RGB image in 1920x1080 pixels format, RGB image in 512x424 pixels format, Depth matrix in 512x424 format. As it is intended to use the depth information and synchronize it with the RGB image, it was decided to use the RGB images that are in the same format and discard the images in 1920x1080 pixels format.

Both training datasets were divided into 70% for training, 15% for validation and 15% for testing. In addition, all images have been resized to 448 x 448 pixels and normalized to a 0-1 scale. The file extension of the RGB images is .jpg and the depth array is in the format .npy (numpy array).

3.2 Testing Models with Data Shuffle

This section presents the results of the Deep Learning Models in the Test Datasets, in which data shuffle was performed. In this way, the Models will be evaluated as usual. The purpose of performing these tests is to compare the results of using the model with the depth images to the ones with only RGB images, in different situations, to verify if the depth information is important for the leather segmentation or not.

For the Full Image Model training, the number of epochs varied between 40 and 60 depending on the situation, the training was carried out with the GPU and the time needed was about half an hour for each test. All results can be seen in Table 2.

Comparing the results of the models in the Movements Type 1 Dataset, the depth information does not improve the results, and the model with only the RGB information, presents better results. It is important to emphasize that both models segment the deformable object with a high IoU (Intersect over Union). Regarding the Movements Type2 Dataset, the depth data improved the Deep Learning Model although the difference between the results is not significant, where the Model with only RGB information has a Dice Loss of 0.1312 and the Model with RGB-D information has a Dice Loss of 0.1192.

Table 2: Test Results with Full Image Model.

Dataset	Dice Loss	IoU	Recall	Precision
RGB Movement Type 1 Dataset	**0.0481**	**0.9117**	**0.9221**	**0.9849**
RGB-D Movement Type 1 Dataset	0.0727	0.8793	0.9011	0.9826
RGB Movement Type 2 Dataset	0.1312	0.7810	0.8174	**0.9712**
RGB-D Movement Type 2 Dataset	**0.1192**	**0.7933**	**0.8435**	0.9583
RGB Both Datasets	0.1101	0.8128	**0.9474**	0.8961
RGB-D Both Datasets	**0.0717**	**0.8761**	0.9289	**0.9760**

When both Datasets are joined, the images with both movements are evaluated and the intention is to compare the results with a more robust dataset. With the increase of data, it is possible to verify that the depth information has a greater importance for the Deep Learning Model and the difference is significant. In Table 2, it is possible to see that the value of Dice Loss with only RGB data is 0.1101 and with RGB-D data is 0.0717.

It is possible to conclude that, when the data shuffle is performed, the Full Image Model presents better results with the depth information according to the robustness of the Dataset. If the Dataset has few images, the depth information does not have a big impact on the deformable object segmentation.

Table 3: Test Results with Bounding Box Model.

Dataset	Dice Loss	IoU	Recall	Precision
RGB Movement Type 1 Dataset	0.1966	0.7286	**0.8697**	0.8621
RGB-D Movement Type 1 Dataset	**0.1553**	**0.8022**	0.8531	**0.9763**
RGB Movement Type 2 Dataset	**0.3488**	**0.5378**	**0.6156**	0.8905
RGB-D Movement Type 2 Dataset	0.3576	0.5445	0.5891	**0.9398**
RGB Both Datasets	0.2732	0.6023	0.7128	0.7930
RGB-D Both Datasets	**0.1991**	**0.7008**	**0.7761**	**0.9043**

For the Bounding Box Model, the same tests were performed but this time, using the necessary Model to perform the Tracking Algorithm. The results can be seen in Table 3.

Regarding Datasets Movements Type 1 and Type 2, both models presented equivalent results, but in both situations the Models with only RGB information presented better results. In the Dataset Movements Type 2, the difference in results is practically null and it is not possible to verify their difference in the predict images.

Regarding the results when both Datasets are merged, the depth information did not show any impact on the tracking of the deformable object, being that the value of Dice Loss when only RGB information is used is 0.0769 and the value of Dice Loss when it is RGB-D information used is 0.1656. The fact that the depth information does not show any impact on the Bounding Box Model in many situations show that, when images are small, the depth information is not important for Leather Tracking.

3.3 Testing Models with a New Point of View

In this section, are presented the results of the models when it is necessary to predict a new point of view that had never been presented in the training phase. The purpose of performing these tests is to verify the generalization level that the proposed Models can achieve.

For the Full Image Model training, the number of epochs varied between 40 and 45 depending on the situation, the training was carried out with the GPU and the time needed was about half an hour for each test. All results can be seen in Table 4.

In the first test, only the Movement Type 1 Dataset was used and the results with RGB-D information are better compared to the model with only RGB images. The Model with RGB-D information has a Dice Loss of 0.1553 while the Model with only RGB information has a loss of 0.1966.

In general, the leather segmentation is more complicated in the Movement Type 2 Dataset because it presents more articulated movements of the leather, making leather segmentation results difficult. Regarding RGB and RGB-D data comparison, the best results are obtained by the model with only RGB information. The model with RGB information has a dice loss of 0.3488 and the model with RGB-D information has a dice loss of 0.3576.

Table 4: Test Results with Full Image Model.

Dataset	Dice Loss	IoU	Recall	Precision
RGB Movement Type 1 Dataset	**0.0412**	**0.9216**	**0.8015**	0.9877
RGB-D Movement Type 1 Dataset	0.0546	0.9007	0.7656	**0.9913**
RGB Movement Type 2 Dataset	**0.1303**	**0.7859**	**0.5465**	0.9837
RGB-D Movement Type 2 Dataset	0.1308	0.7837	0.5438	**0.9892**
RGB Both Datasets	**0.0769**	**0.8645**	**0.7203**	**0.9898**
RGB-D Both Datasets	0.1656	0.7535	0.6411	0.9895

In the end, it was decided to create a model with information from both datasets (Movement Type 1 Dataset and Movement Type 2 Dataset). The Model are evaluated with the Movement Type 2 Dataset, presented in Table 4. Since the purpose of creating this model was to improve the Movement Type 2 Dataset results, as it is the most complex Dataset for leather segmentation. As the model trained in the Movement Type 2 Dataset and the model in Both Dataset were tested with the same images, it is possible to compare the results between them.

It is easy to verify that the model with both datasets presents better results indicating that if the model trains with different types of leather and different movements, it helps in its segmentation. Another important factor to highlight is that the model with RGB-D information has a much higher performance compared to the model with only RGB information. The reason for the Movement Type 2 Dataset to have superior results with only RGB information may be related to the fact that the Movement Type 2 Dataset has less images, 610, and only 70% of them were used for training. When more images are used for training (as in the case of the Both Datasets Model) the results were superior with the depth information indicating that it is important for

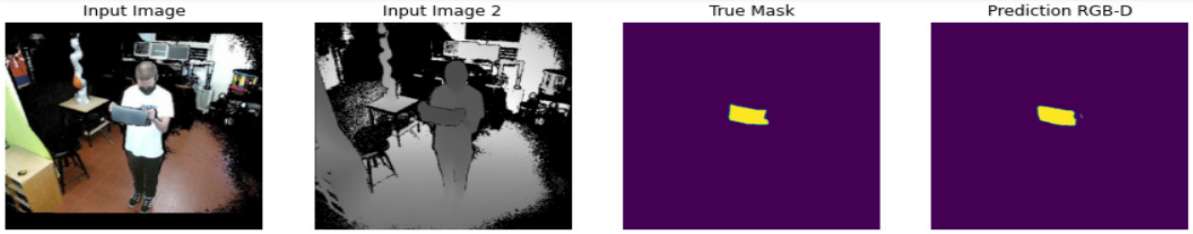

Figure 5: Example of a predict with Full Image Model.

the leather segmentation, but more images are needed for the training process. In Figure 5 you can see an example of a forecast with the Full Image Model.

In general, depth information is important for leather segmentation when using the Full Image Model. After the tests were carried out on the Full Image Model, the same tests were carried out but, this time, referring to the Bounding Box Model to verify or not the importance of depth information in small images, such as in the Bounding Box case.

For the Bounding Box Model training, the number of epochs varied between 40 and 60 depending on the situation, the training was carried out with the GPU and the time needed was about half an hour for each test. All results can be seen in Table 5.

In the first test with the Bounding Box Model, it is possible to verify that, in the Movement Type 1 Dataset, the Model presents better results when RGB-D information is used instead of RGB information, but the results are quite similar. The dice loss with RGB-D information is 0.1599 and only with RGB information is 0.1727. In relation to the Movement Type 2 Dataset, the depth information has no relevance for its prediction and, as in the Movement Type 1 Dataset, the results are quite similar between them. It is interesting to verify that, contrary to the Full Image Model which presented much better results in the Movement Type 1 Dataset in relation to the Movement Type 2 Dataset, in the Model Bounding Box the results are better in the Movement Type 2 Dataset. This fact is since because as this is the Bounding box model, the images it predicts are approximations of Leather and, due to this fact, there is no difference in results between datasets because both are similar.

Finally, the model that trains with both datasets also shows better results when it is only trained with RGB information. It is possible to see that the model with only RGB information has a Dice Loss of 0.1272 and with RGB-D information it has a Dice Loss of 0.1628. In Figure 6 it is presented an example of a prediction using the Full Image Model. It is possible to conclude that, when we have the complete image and a larger amount of data, the depth information is useful for leather segmentation, as is the case with the Full Image Model. When we have few images or the images are very small, the depth information is not useful, as in the case of Bounding Box Model.

Table 5: Test Results with Bounding Box Model.

Dataset	Dice Loss	IoU	Recall	Precision
RGB Movement Type 1 Dataset	0.1727	0.7681	0.6702	**0.9859**
RGB-D Movement Type 1 Dataset	**0.1599**	**0.7845**	**0.6864**	0.9855
RGB Movement Type 2 Dataset	**0.1226**	**0.7876**	**0.6350**	0.9640
RGB-D Movement Type 2 Dataset	0.1411	0.7623	0.6167	**0.9715**
RGB Both Datasets	**0.1272**	**0.7813**	**0.6151**	0.9694
RGB-D Both Datasets	0.1628	0.7288	0.5499	**0.9794**

Comparing the results of section 3.2 and this section, it is possible to verify that the results of the section 3.2 are better due to the fact that the models were trained with all points of view, but the purpose of this section is to demonstrate that the Model, even if it is never trained with a specific point of view, can obtain good results.

3.4 Tracking Algorithm Speed

The purpose of this section is to compare the performance of the proposed approach when the Tracking Algorithm is or not used and when the GPU is or not used.

Table 6: Tracking Algorithm performance.

Input Data	Without Tracking Algorithm	With Tracking Algorithm	GPU
RGB	4 FPS	10 FPS	No
RGB-D	3 FPS	8/9 FPS	No
RGB	11/12 FPS	13 FPS	Yes
RGB-D	10 FPS	13 FPS	Yes

As can be seen in Table 6, when the CPU is used, the Tracking Algorithm has a big impact on the FPS value, without the Tracking Algorithm the FPS values are between 3/4 and with the Tracking Algorithm the FPS values are between 9/10 and the use of the

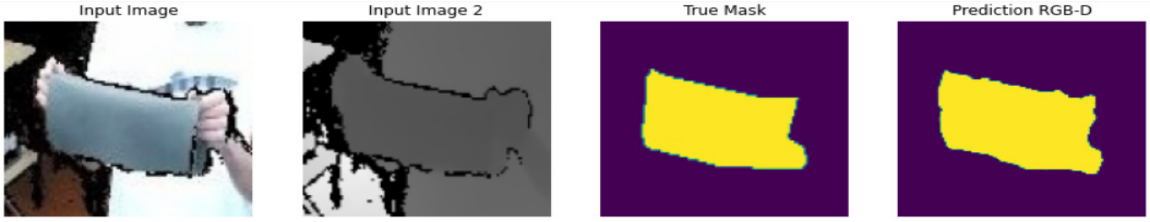

Figure 6: Example of a predict with Bounding Box Model.

Tracking Algorithm is beneficial either with RGB-D information or just with RGB information. When using the GPU, the impact of the Tracking Algorithm is less, but in both situations, it is beneficial to use the Tracking Algorithm.

It is important to emphasize that, using the Tracking Algorithm and the GPU, the depth information has no impact on the leather segmentation velocity. The FPS results were calculated with the models performing the prediction in all frames and, when it is necessary to increase the tracking velocity, the prediction does not need to be performed in all frames, increasing the FPS.

4 CONCLUSIONS

Using a U-NET architecture for the Deep Learning model helped us to get better results compared to creating an architecture from scratch. In addition to not having to train for many epochs, the results are also satisfactory.(Ronneberger et al., 2015)

In general, the depth information is important for the Deep Learning Model when it is intended to segment a deformable object, but it is necessary to pay attention to the dataset size. With a larger dataset the depth information has more impact, but if the dataset is small or the images have low resolution, the depth information is not so useful for the Deep Learning Model.

The Tracking Algorithm proved to be useful to increase the system's processing velocity, in this way, it is possible to increase the number of FPS and manage to track the Leather in the same way. However, it is necessary to pay attention to situations in which the Bounding Box Model loses the object. To prevent this situation, it is possible to create an architecture that, in some situations, uses the Full Image Model to guarantee the correct location of the Leather.

For future work it is important to increase the dataset size and include different types of Leather. This way, the depth information will have a greater impact on the Models. In addition, it is possible to add more data processing steps, but it is necessary to pay attention to the impact of each technique on the system as it will have to work in real time.

ACKNOWLEDGEMENTS

This work is supported by: European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project nº 42778; Funding Reference: POCI-01-0247-FEDER-042778].

REFERENCES

Hu, Z., Han, T., Sun, P., Pan, J., & Manocha, D. (2019). 3-D Deformable Object Manipulation Using Deep Neural Networks. IEEE Robotics and Automation Letters, 4(4), 4255–4261. https://doi.org/10.1109/LRA.2019.2930476

Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. ArXiv:1412.6980 [Cs]. http://arxiv.org/abs/1412.6980

Lai, K., Bo, L., Ren, X., & Fox, D. (2011). A large-scale hierarchical multi-view RGB-D object dataset. 2011 IEEE International Conference on Robotics and Automation, 1817–1824. https://doi.org/10.1109/ICRA.2011.5980382

Liu, Z., Shi, S., Duan, Q., Zhang, W., & Zhao, P. (2019). Salient object detection for RGB-D image by single stream recurrent convolution neural network. Neurocomputing, 363, 46–57. https://doi.org/10.1016/j.neucom.2019.07.012

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., & Sorkine-Hornung, A. (2016). A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 724–732. https://doi.org/10.1109/CVPR.2016.85

Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., & Van Gool, L. (2018). The 2017 DAVIS Challenge on Video Object Segmentation. ArXiv:1704.00675 [Cs]. http://arxiv.org/abs/1704.00675

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. ArXiv:1505.04597 [Cs]. http://arxiv.org/abs/1505.04597

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: A Database and Web-Based Tool for Image Annotation. International Journal of Computer Vision, 77(1–3), 157–173. https://doi.org/10.1007/s11263-007-0090-8

Song, S., & Xiao, J. (2013). Tracking Revisited Using RGBD Camera: Unified Benchmark and Baselines. 2013 IEEE International Conference on Computer Vision, 233–240. https://doi.org/10.1109/ICCV.2013.36

Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B. B. G., Geiger, A., & Leibe, B. (2019). MOTS: Multi-Object Tracking and Segmentation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7934–7943. https://doi.org/10.1109/CVPR.2019.00813