ITERATIVE DATASET FILTERING FOR WEAKLY SUPERVISED SEGMENTATION OF DEPTH IMAGES

Thibault Blanc-Beyne[1,2] *

[1]Ebhys
ZA La Cigalière III, 84250 Le Thor

Axel Carlier[2] *and Vincent Charvillat*[2]

[2]Université de Toulouse - IRIT
2 rue Charles Camichel, 31079 Toulouse

ABSTRACT

In this paper, we propose an approach for segmentation of challenging depth images. We first use a semi-automatic segmentation algorithm that only takes a user-defined rectangular area as an input. The quality of the segmentation is very heterogeneous at this stage, and unsufficient to efficiently train a neural network. We thus introduce a learning process that takes this imperfect nature of data into account, by iteratively filtering the dataset to only keep the best segmented images. We show this method improves the neural network's performance by a significant amount.

Index Terms— Depth image segmentation, Weakly supervised learning

1. INTRODUCTION

Tremendous progress has been made in the past six years in image processing, driven by the advances of deep learning and the rise of GPU computational power. Deeper and deeper convolutional neural network can now be trained thanks to residual blocks based architectures, allowing to address problems of virtually any complexity. However, all these advances require substantial amounts of training data, which is costly and may in some cases be impossible to obtain. A shift is thus starting to appear in the research community, towards learning paradigms requiring less data, such as unsupervised, weakly-supervised, or few-shots learning.

In the project we are working on, we are interested in a problem along these lines. We address the problem of human activity monitoring in an industrial workplace, to prevent physical injuries such as musculoskeletal disorders. We are bound by several constraints due to the particular context of our work. We use depth sensors (without RGB information) to protect worker's privacy and favor our system's acceptability. The sensor position is constrained by the industrial environment; sensors are placed on the ceiling, looking down towards the observed human. The environment is challenging, with uncontrollable lighting conditions, reflexing surfaces and clothes, and moving objects everywhere. All these hard conditions create very noisy depth images in which the human operator is difficult to detect and track. We can not rely on existing approaches, which often assume the human to be facing the camera at a reasonable distance, to even segment the human body in our images (cf. Figure 2).

In this paper, we focus on the problem of segmenting human body in noisy depth images, in order to later perform higher-level tasks (e.g. action recognition [1] or pose estimation [2]). Some sensors, like the Microsoft Kinect, may provide a segmented depth frame, but it is not the case for the majority of current sensors. Furthermore, most datasets, such as the Microsoft Research Action3D dataset [3], provide an already segmented image of the user. Indeed, most current work on depth images require the user to be segmented from the scene [1, 2, 3], while the remainder of the image may be seen as noise and significantly pollute the results of most algorithms, often making those unusable due to their lack of robustness. Segmenting the user from the rest of the frame is therefore a crucial task when dealing with depth images.

The particular context of our project forces us to assume we do not have enough labelled data to build a training dataset, because of the variability of our images: the environment may vary and the morphology and clothes of the human operators is unpredictable as well. Instead, we introduce a process (see Figure 1) to build and refine an automatically-labelled dataset, thanks to basic image processing techniques and weak human supervision. In what follows, we first review the literature on the topic in Section 2, then detail our algorithm for semi-automatic human segmentation (Section 3). In Section 4, we explain how we progressively cleanse our dataset from too imprecise segmentation masks during training. Finally, we present our experiments in Section 5 and conclude.

2. RELATED WORK

While RGBD cameras have been introduced for many years now (the first version of the Kinect sensor was commercialized by Microsoft in 2010), only a few recent work have focused on inferring information out of depth images only. Most previous work use the human segmentation and pose estimation embedded in the Kinect [2], which implicitly assumes the human is standing in front of the sensor, at approximately the same height. In our work, we choose to only

*This work was supported by CIFRE ANRT 2017/0311.

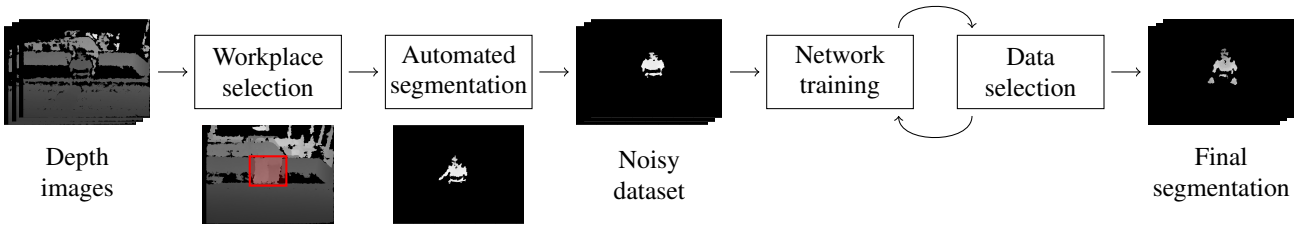

Fig. 1. The pipeline of our proposed segmentation process. The only part that needs user interaction is the selection of the zone before performing the automated segmentation. The rest of the process is automated.

Fig. 2. Example of images from our dataset (top row) and their associated ground truth segmentation (bottom row).

use the sensor's depth information, to respect human operators privacy. This approach has been previously adopted in the context of hospitals, for fall detection [4] or action recognition [5]. Some authors have performed face recognition on depth images [6] depicting close-ups on human faces. This constraint is very far from our setup, for which face recognition would be too challenging due to the distance and orientation of the camera.

Image segmentation has been traditionally a very challenging task in computer vision, due to the complexity of having to classify each pixel in an image. While deep learning approaches became popular for image classification in 2012 [7], groundbreaking work in image segmentation arrived later with fully convolutional networks [8] and U-net [9]. State-of-the-art architectures nowadays include residual blocks, introduced in ResNet [10] which won the ImageNet Large Scale Visal Recognition Challenge in 2015. In our work, we adapted the SegNet architecture proposed in [11] and the BiSeNet network recently introduced in [12] which obtains very good results on natural image segmentation and has the advantage to run very fast. We propose an automatic segmentation approach using neural networks as they have already proven their ability in resolving such hard tasks in the RGB domain [12, 13]. Neural networks have become popular partly because they are known to generalize well, and we think this could help improve results in our problem. However, there are not as many works in this area using depth images, as the most common sensor, that is the Microsoft Kinect, already provide a suitable segmentation of the user in common use-cases (i.e when the sensor lays in front of the user).

Algorithm 1: Automatic algorithm to segment data

input : the depth image *image*,
the static background *background*,
the selected zone *zone*,
the tracking point *point*,
the minimum area of the operator *area*
output: the segmented depth image
def *segment(image, background, zone, point, area)*:
  fill blank pixels in *image*
  substract *background* in *image*
  remove noisy pixels in *image* using erosion
  construct a binary mask of *image*
  detect the contours of objects in the mask
  remove objects whose area $<$ *area*
  if *there is a least one object inside zone* **then**
    take the one whose center is closest to *point*
  else
    take the object whose center is closest to *point*
  end
  set *point* as the center of selected object
  set *area* as half of the area of the selected object
  segment *image* by using object as a binary mask
  return segmented *image*

3. WEAKLY-SUPERVISED SEGMENTATION

One of the challenges of the problem we are trying to solve is the critical need of a lot of labelled data. It is well known that labelling data is often a challenging task, and always a human-time consuming task. That is the reason why we decided to rely on a first automated but noisy segmentation to train our network.

As we show in Section 4, the training of our network is thus adapted to the raw nature of this dataset, as we can not expect it to be a real ground truth.

We perform a first segmentation using an ad-hoc process, adapted to our particular setup. The purpose of this segmentation is to construct a cheap, large dataset of noisy segmentations to feed our segmentation neural network. Consequently, this segmentation does not need to be perfect, but it needs to be precise enough to be reliable for the next processing. The pseudo-code of our algorithm is given in Algorithm 1.

This method requires pre-processing: the user is asked to select a quadrilateral zone where the operator should evolve and we establish the center of it as the initialization of a tracking point. It is also needed to build a static background. We achieve this by taking the mean image of a set of depth frames of the empty scene.

The first step is to remove the static background of the scene. We begin by filling the blank pixels in the depth image. We then remove the background on the images to segment by comparing the depth values between each pixel of the static background and the depth image to segment and applying erosion operations to remove isolated pixels due to the noisy structure of a depth image. This allows us to remove any static object present in the scene.

The second step is to remove the moving objects that are not the operator. We then detect the contours of the remaining objects using the Canny edge detector [14] and keep only elements that are big enough to represent a human. Afterwards, we find objects that are located inside the zone selected by the user, and finally choose the object whose center is the closest to the tracking point. If there is no object located inside the zone, we take the object located outside whose center is also closest to the tracking point.

When we are segmenting the following frames of a depth video, we update both the tracking point position and the minimal size of the object for the next frame segmentation.

This allows us to extract a first segmentation of the operator from the depth frames. As it is easily understandable, this segmentation is not robust: a trivial example is that, an object that is held by the operator will almost always be extracted as a part of the operator, as depicted in Figure 3.

Our dataset contains 59 384 images and their associated segmentation. Examples of this dataset are given in Figure 3.

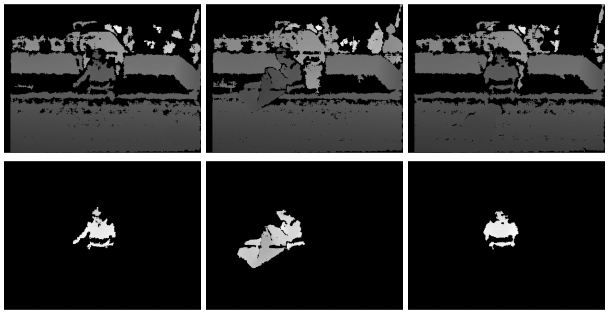

Fig. 3. Examples of images from our dataset (top row) and their associated automatic segmentation obtained thanks to our algorithm (bottom row). While some segmentations are very good (left), presence of large objects may pollute the results (middle). Furthermore, hands may be missing (right).

4. ITERATIVE TRAINING WITH DATA FILTERING

We use a neural network to improve the quality of the segmentation. We feed this network with the dataset presented in Section 3. To reduce the number of weights in the network, we resize the images from 640×480 to 64×64.

4.1. Neural network architecture

Our neural network is an encoder-decoder inspired from SegNet introduced by [11]. As we are dealing with images both as input and output, it is a deep fully convolutional neural network. The encoder part contains four pairs of convolutional layers separated by a max-pooling layer. We apply a batch normalization layer and a ReLU activation after each convolutional layer. The decoder part follows the same patterns, with max-pooling layers replaced by up-sampling layers. The last convolution layer is followed by a sigmoid activation.

We also compare its results against an adaptation of the recently introduced BiSeNet from [12], which has shown good performances in semantic segmentation of RGB images. We adapted both BiSeNet network and its Xception component [15] to work with our depth images, by simply modifying the input shape of both networks.

To train both networks, we use the default Adam optimizer and a binary cross-entropy loss.

4.2. Updating the dataset

As our training dataset can not be considered as a ground truth, we introduce a method to actively select data of acceptable quality, and update our dataset accordingly.

We observe that the quality of the segmentation obtained through the automatic algorithm presented in Section 3 is highly variable depending on the images. Some segmentations are almost perfect, while others are totally wrong. The challenge is to correctly separate the wheat from the chaff using only the output of the network's training. We choose to use the Jaccard similarity coefficient (or intersection over union) to evaluate the similarity between the segmentation predicted by the network and the segmentation given in the dataset for all images contained in the dataset.

We decide to update the dataset and remove the worst data during the learning phase of the network. Our idea comes from the fact that the network is quickly able to provide an intuition of the correct segmentation of the image, and that further learning, that might be seen as over-fitting, refine this intuition into the expected segmentation, in our case by adding what we see as noise, such as objects held by the operator, to fit the training labels.

Method	Min.	Median	Mean	Max.
Noisy Seg.	0.000	0.814	0.762	**1.000**
BiSeNet	0.037	0.686	0.649	0.874
SegNet	0.015	0.738	0.691	0.915
BiSeNet Up.	0.126	0.821	0.774	0.988
SegNet Up.	**0.259**	**0.842**	**0.797**	0.990

Table 1. Jaccard of our different segmentations against the ground truth after 50000 minibatches (around 27 epochs of the complete dataset).

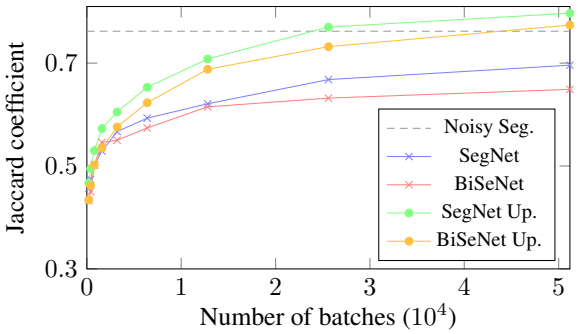

Fig. 4. Evolution of the Jaccard during the training. One epoch on the initial dataset is around 1856 batches. Learning with the complete dataset stays under the noisy segmentation performance, while updating the dataset outperforms it.

We make use of this intuition of the segmentation by the neural network to remove the worst data of the dataset during the first steps of the learning. As shown in Section 5, this helps to improve the results of the network. We do not replace the removed data by the output of the neural network, because we think that these outputs are too coarse to be used as learning labels.

5. EXPERIMENTS

In order to quantify the impact of our training policy, we manually annotated 520 images using the software provided by [16], and consider these images to be the ground truth of our segmentation. We carefully chose this set of images to display a wide range of poses taken by the operator. We use this data to measure the performance of our algorithms (see Figure 2 for examples). The comparison between ground truth and the segmentations are given in Table 1 and Figure 4. Results of the segmentation are shown in Figure 5.

As we can see, our noisy segmentation performs generally well, with a mean Jaccard value of 0.762. The main issue is that even if it is perfect in some case (Jaccard of 1.0), it may completely fail in other cases (Jaccard of 0.0). Besides and as expected, training state-of-the-art neural networks such as BiSeNet or SegNet using this dataset does not improve the results: both median and mean Jaccard value decrease. The

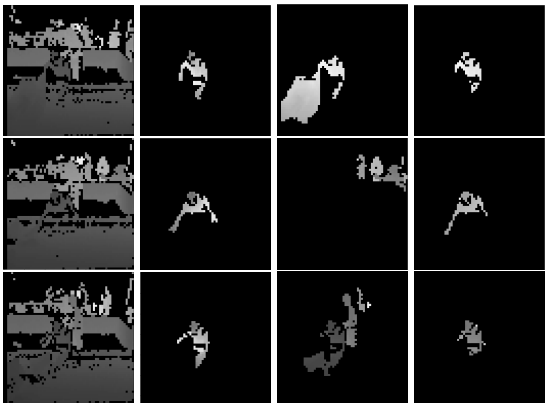

Fig. 5. Results obtained by our network. Left column: input of the neural network. Second column: ground truth. Third column: output of the automated segmentation. Last column: output of the neural network. The neural network removes objects from the hands of the operator (first row) and gives a reliable segmentation in cases where the automated segmentation fails (second and third rows).

neural networks training is affected by the too large proportion of bad segmentations in our dataset.

Table 1 shows that updating the training set not only improves the results compared to the usual learning, but also improves the results given by the noisy segmentation, which is our goal. We use a threshold Jaccard coefficient of 0.75 to select the data to remain in our dataset, and update the dataset every 200^n minibatches.

We conducted a lot of experiments to correctly tune the parameters of our updating scheme. The main information given by these experiments are that the parameters highly depend on the network and the quality of the noisy dataset so there is no miracle parameter. One key point is to frequently update the dataset during the early stages of the learning, and then to reduce the update frequency or even stop them. It is also crucial to reinitialize the weights of the network at each dataset update.

6. CONCLUSION

We propose a method to provide a good quality human segmentation from depth images using a neural network trained on an noisy dataset. We obtain this dataset by performing an ad-hoc weakly supervised automated segmentation algorithm. We use it to train our neural network. During the learning, we update the training data several times to remove the worst data, using the early outputs of the neural network.

Our original learning scheme allows the network to learn better and we achieve a mean result of 79.7% on our dataset, while the usual training procedure only reaches 69.1%.

In future work, we plan to explore this learning scheme to further improve the results, for instance by combining the noisy segmentations with the outputs of the neural network.

7. REFERENCES

[1] C. Chen, K. Liu, and N. Kehtarnavaz, "Real-time human action recognition based on depth motion maps," *Journal of real-time image processing*, vol. 12, pp. 155–163, 2016.

[2] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 1297–1304.

[3] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2010, pp. 9–14.

[4] T. Banerjee, M; Rantz, M. Li, M. Popescu, E. Stone, M. Skubic, and S. Scott, "Monitoring hospital rooms for safety using depth images," *AI for Gerontechnology, Arlington, Virginia, US*, 2012.

[5] E. Chou, M. Tan, C. Zou, M. Guo, A. Haque, A. Milstein, and L. Fei-Fei, "Privacy-preserving action recognition for smart hospitals using low-resolution depth images," *arXiv preprint arXiv:1811.09950*, 2018.

[6] Z. Cheng, T. Shi, W. Cui, Y. Dong, and X. Fang, "3d face recognition based on kinect depth data," in *2017 4th International Conference on Systems and Informatics (ICSAI)*. IEEE, 2017, pp. 555–559.

[7] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 3431–3440.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *2017 IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, pp. 2481–2495, 2017.

[12] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *European Conference on Computer Vision*. Springer, 2018, pp. 334–349.

[13] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A.L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, pp. 834–848, 2018.

[14] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, 1986.

[15] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 1800–1807.

[16] K. McGuinness and N.E. Oconnor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, 2010.