Learning human motion features and trajectory predictions in large changing environments

Robin Eberhard, Stefan Dörr

***Abstract*—This study presents a new approach to detecting humans in large environments. Making use of the latest advances in neural networks, fast scalable applications can be modeled without having knowledge about the inner workings. We present a leg detection tool, that operates only on laser scan data, classifying each point as human or non-human. The results are further used to learn to identify people, in order to predict their trajectories, which will be used for movement in crowded scenes.**

***Keywords*— .**

I. Introduction

Introducing autonomous robots into customer service environments, adds a new set of problems, that deal with interactions of people and robots. These could include for example following and approaching customers, and navigating through a crowded environment.

To solve these problems, the first step is the detection of people, which is here done using a safety laser detector on leg-height[1]. In the past, this has already been approached with model-based solutions, that require prior knowledge about shapes and behaviour of legs [1] [2]. We present a new approach, where a neural network learns characteristics on its own and places unique identifiers on persons, making it easy to track them over a long period of time.

Having the information and history of positions of people, we can learn trajectories, behaviour and intentions when the robot is serving customers. We provide a way to model the trajectories using Long short-term memory (LSTM) cells in combination with a Mixture density layer, which outputs a set of normal distributions, similar to [3] [4].

In this paper, we present the models that are used to train the program, as well as some benchmarking and comparisons with similar projects.

II. State of the art

LIDAR scanners pose an attractive way to scan the environment, due to the simple interpretation of the data and the low computational power required. The idea to extract information about people has been introduced in [5] where laser range measurements (LRM) are grouped into blobs and objects. Moving objects could then be considered as people while any other object was not of interest. In order to detect still-standing people, different approaches [6], [7] used a combination of LRM and imaging sensors.

Arras et al. proposed a machine learning approach [1], where pre-defined geometrical features were trained using supervised learning. It was further adapted in [2] to expand the detection in retirement homes, allowing the classification of people with different walking aids. This approach is very versatile, as it can easily be adapted by simply adding further features for a given class.

We improve on this idea, by using neural networks as the machine learning technique. This way, no knowledge about shapes or behaviour is required, which makes this approach easy to implement and also requires less computational power. For the human eye, one big factor in the detection of legs on a laser scan is the movement of people. By providing the network with multiple frames at once, we can therefore increase the accuracy, as the system will then take motion into account.

III. Model architecture

Learning positions of people and identifying them requires a neural network to solve several tasks. First, the laser scans are classified by using a set of convolutional and pooling layers. The points labeled as belonging to legs are sorted out and typically clustered based on euclidian distance, in order to identify persons. This approach does not work well, when people are close together or even when they are moving, as the program would then either cluster too many legs together or see two legs as separate people respectively.

This can be countered by supplying the clustering algorithm with additional information. During the convolution step, the network learns features which are unique to legs in different situations. Those can be extracted and passed as additional parameters to the clustering algorithm, which not only allows higher accuracy, but also providing a pair of legs with a unique identifier.

Having identifiers and positions, they can then be fed into the third and last step of the setup, which aims to predict trajectories of individual people. The network takes a set of positions, that are run through an LSTM cell, which implicitly works as a memory for the program and holds the information necessary to build future trajectories.

A. Classification of the laser scan

In order to classify a laser scan, a snapshot of the current scan is taken in equal timesteps. Each snapshot contains one dimensional arrays of length N_{points} with distances r_i to the closest intersections from the laser beams.

From the one-dimensional array as input, the network would only find characteristics in the objective shapes of the laser scans. To also account for movement, N_{frames} snapshots are composed together. The desired outputs are probabilities for each point in the most current snapshot to be in either class leg or non-leg.

[1]Sick S300 safety laser scanner

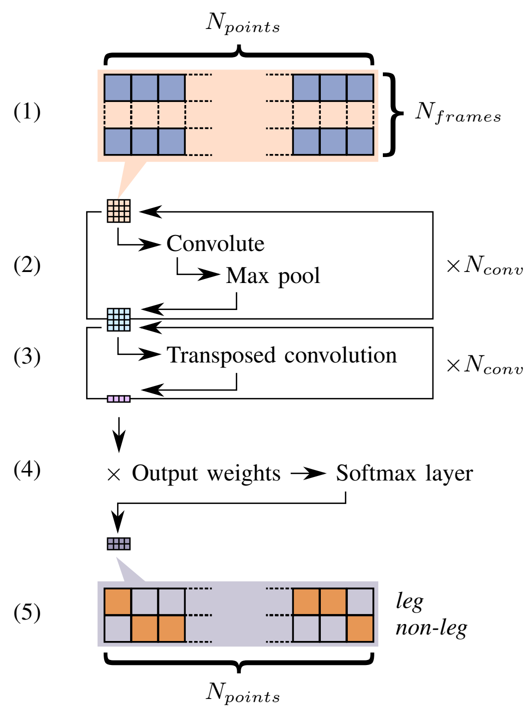

Fig. 1. **Laser scan classification.** (1) The input is a 2D array of $N_{points} \times N_{frames}$ containing distances from a laser scan. It is fed into (2), where features are extracted using convolutions and downscaled using a pooling layer. After N_{conv} iterations of (2), the now one-dimensional output is upscaled in (3) into the shape N_{points}. (4) multiplies the output and applies the softmax function, resulting in probabilities of each point belonging to class leg or non-leg in (5).

Therefore, the mathematical description of the model can be described by a function F that maps an input of size $[N_{frames}, N_{points}]$ to an output of size $[N_{points}, 2]$.

The output is modelled from the input as outlined in Figure 1 and described in more detail in the following. When convolutional layers are used for classification, a network will train the weights such that they represent characteristics for a class. In this way, the characteristics are first of geometrical nature and become more abstract the more layers there are.

The input is therefore fed into a convolutional layer and then downsampled using a max pooling layer. This is repeated N_{conv} times, after which the output is a one dimensional array of length smaller than N_{points}.

To gain a classification for each point in the most current laser scan, the output is upscaled using transpose convolutions, again extracting features. Multiplication by output weights and applying the softmax function results in the desired shape $[N_{points}, 2]$. Each entry consists of the two probabilities of a point being in class leg P_{leg} or non-leg P_{nonleg} with $P_{leg} + P_{nonleg} = 1$.

B. Clustering of leg points

In order to gain information about the current position and identity of people, the classified points are clustered together. This usually poses two major problems:

- The clustering has a high margin of error when legs of a single person are too far away or when legs of different people are together
- tracking of a person has to be handled externally.

With the following approach, we find a high accuracy and are also able to track people on certain features.

As the inputs are convoluted, the network finds characteristics, such as shapes and movement. In order to find the most important characteristics, a training was implemented to find weights v, such that parameters of people in different situations are easily separable. The ith convoluted layer has its weights w_i multiplied by v_i and the resulting cluster parameter p_i is then found by the sum of all weights:

$$p_i = v_i * \sum_j w_{i,j} \tag{1}$$

This introduces $2 * N_{convs}$ additional parameters to the clustering algorithm, highly increasing accuracy when used together with the euclidian distance. As the parameters have been chosen such that people in different situations are easily separable and given the unlikeliness of two people being in the same position in the same situation, we can additionally use the parameters to place a similar identifier on a single person over several frames. This allows to track them in order to predict their movements.

C. Trajectory prediction

With the information of position of individual people, we can learn to predict trajectories over some time. A trajectory of a single person can be modeled with a normal distribution, or if the path is unclear, a number of distributions N_{dist} are necessary.

This leads to the approach of using a mixture density network [3]. The network learns from parameters π, μ, σ and ρ according to [4], so that a distribution is given as:

$$\mathcal{N}(x|\mu, \sigma, \rho) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[\frac{-Z}{2(1-\rho^2)}\right] \tag{2}$$

with

$$Z = \frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} - \frac{2\rho(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2}. \tag{3}$$

Therefore we can define the ensemble of distributions as

$$Pr(x|y) = \sum_{j=1}^{N_{dist}} \pi_j \mathcal{N}(x|\mu_j, \sigma_j, \rho_j) \tag{4}$$

for every input point x from the laser scan.

As the future movement depends on previous positions, the parameters μ, σ, ρ and π are derived from a recurrent network cell. A long-short term memory (LSTM) was used, which allows for information to be kept over a long time. This way, the network was trained on different patterns, so that it can recognize similar ones later in the evaluation. The LSTM cell will take the input positions and directly outputs the parameters for the mixture of normal distributions.

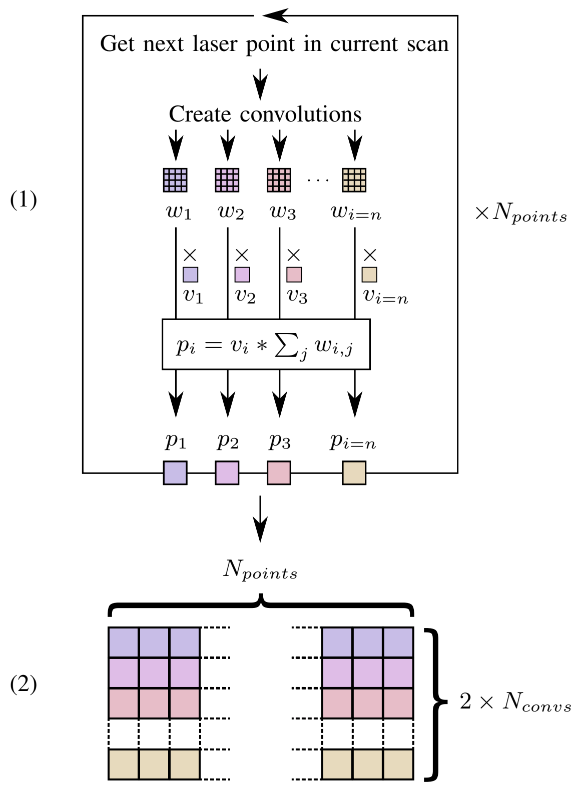

Fig. 2. **Convolution-based clustering.** (1) For every point in the current laser scan, N_{conv} convolutions are created. A clustering parameter p_i is derived from the weights w_i of the convolution and the trained weights v_i, which assumes that some convolutions are more important than others. (2) The resulting clustering matrix holds $n = 2 * N_{conv}$ parameters for each point in the current laser scan.

IV. Experimental results

In order to evaluate the leg detector, the robot was set up in a corridor, recording legs of people passing through over a course of three hours. To validate the program, a region was defined, containing only walls and immovable objects. Therefore any laser scans, that were not inside the region could be identified as legs.

We find, that the detector works best in close range, while becoming weaker as the range increases. This is demonstrated in Figure IV, where a short, 100 second long exposure was analyzed. The leg detection was compared to the ground truth and we find almost perfect results in close range to the robot. The detection gets increasingly worse until a maximum range of $5.0m$.

The decrease originates from two major causes. For larger ranges, legs are defined by fewer laser points, which makes it more difficult to extract features, as there is not enough data available. Secondly, the steep decrease, which is seen close to $5.0m$ is due to the absence of training data in that range, the limits of clustering too few points and the range of the laser scanner itself.

Finally, we set the scanning range to $2.2m$, where the accuracy is still high and enough data points are available. In a customer environment, it is important, that there are as few false positives as possible, to not occupy a robot in a false

TABLE I. My caption

	Detected Label		
True Label	Person	No Person	Total
Leg	**100** ($xyz\%$)	**100** ($xyz\%$)	**200**
No Leg	**100** ($xyz\%$)	**100** ($xyz\%$)	**200**

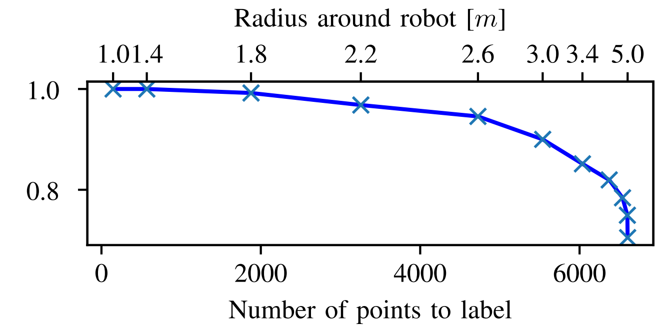

Fig. 3. **Distance dependency of the leg detector.** The diagram shows accuracies of the leg detection as a function of the number of points in a range around the laser scanner. A range around the robot was defined and the containing number of points were counted.

state. From the three hour test run, we find the truth table IV which meets these expectations and also reflects the accuracy of Figure IV.

V. Conclusion

Acknowledgment

The authors would like to thank...

References

[1] K. O. Arras, Óscar Martínez Mozos, and W. Burgard, "Using boosted features for detection of people in 2d range scans," in *IN PROC. OF THE IEEE INTL. CONF. ON ROBOTICS AND AUTOMATION*, 2007.

[2] C. Weinrich, T. Wengefeld, C. Schroeter, and H.-M. Gross, "People detection and distinction of their walking aids in 2d laser range data based on generic distance-invariant features," in *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*. IEEE, 2014, pp. 767–773.

[3] C. M. Bishop, "Mixture density networks," 1994.

[4] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[5] A. Fod, A. Howard, and M. A. J. Mataric, "A laser-based people tracker," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, 2002, pp. 3024–3029.

[6] M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lomker, G. A. Fink, and G. Sagerer, "Person tracking with a mobile robot based on multi-modal anchoring," in *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*. IEEE, 2002, pp. 423–429.

[7] E. Aguirre, M. Garcia-Silvente, and J. Plata, "Leg detection and tracking for a mobile robot and based on a laser device, supervised learning and particle filtering," in *ROBOT2013: First Iberian Robotics Conference*. Springer, 2014, pp. 433–440.