Learning nuisances to track pedestrians in autonomous vehicles

George Adaimi

Alexandre Alahi

Visual Intelligence for Transportation **May 2018**

STRC

18th Swiss Transport Research Conference

Monte Verità / Ascona, May 16 – 18, 2018

Visual Intelligence for Transportation

Learning nuisances to track pedestrians in autonomous vehicles

George Adaimi
Visual Intelligence for Transportation
Ecole Polytechnique Federale De Lausanne
Ecublens VD
phone: +41 21 69 32349
fax:
george.adaimi@epfl.ch

Alexandre Alahi
Visual Intelligence for Transportation
Ecole Polytechnique Federale De Lausanne
Ecublens VD
phone: +41 21 69 32608
fax:
alexandre.alahi@epfl.ch

May 2018

Abstract

Abstract. Autonomous vehicles rely on an accurate perception module. One of the fundamental challenges is to efficiently track pedestrians surrounding a vehicle to anticipate risky situations. Over the past decades, researchers have formulated the tracking problem as a data association one where they proposed various representations aiming for invariance to nuisances such as viewpoint changes, body deformation, object occlusion, and illumination changes. However, these methods still suffer to address abrupt changes since they do not explicitly model the nature of the nuisances.

In this work, we propose to train a classifier that recognizes these nuisances, more specifically rotational body deformation of pedestrians. We aim to detect deformations as a method to find a good representation that will lead to better tracking of pedestrians as well as other tasks.

Keywords

Representation Learning, Tracking, Deformation, Nuisances

1 Introduction

Tracking, in self-driving cars, is an important feature used to deduce the speed and direction of a moving object, typically a pedestrian or a car. It helps in accurately forecasting a detected object's future trajectory. While there is an increasing trend towards self-driving cars, previous research in the field of tracking is still limited in its capacity. The most challenging part in such algorithms is to find a general representation for the object being tracked. Most of the previous work on tracking algorithms try to be invariant to the nuisances that affect its performance such as partial occlusion, lighting change, body deformation, and viewing angle change. Being able to identify and account for such problems during tracking will lead to a better representation, making the algorithm more robust to different real-life situations.

In order to improve such tracking, we first need to identify the type of deformation that occurs and deal with it accordingly. We start by first dealing with a specific type of deformation caused by the rotation of the person being tracked.

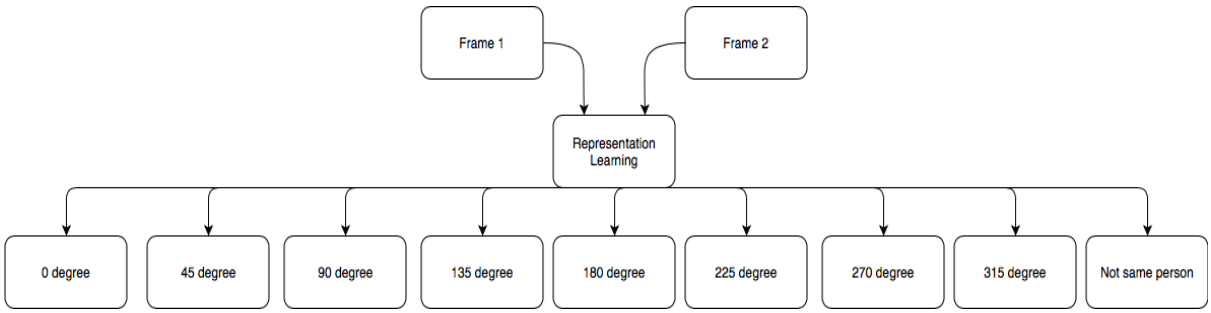
2 Related Work

There had been some work done in detecting the different changes that occur in real-world scenarios. Cheng *et al.* (2014) proposed a background model re-initialization (BMRI) method based on luminance change. This method proved useful in detecting a sudden luminance change by first finding whether the intensity value of a frame differs by a specific threshold from the previous frame. Then, a luminance histogram was used from the output of the first step to detect entropy change. Liu *et al.* (2017) implemented a rotation-invariant object detector. This was done by using a new feature extractor called Sector-ring HOG (SRHOG) and a classifier called Boosted Random Ferns(BRF). By calculating the gradient scale and orientation at every pixel and grouping into cells to obtain the SRHOG descriptor, features that are invariant to rotation were extracted and classified using BRF. BRF is used over other classifiers due its robustness to illumination.

To the best of our knowledge, not much work has been done for detecting rotation changes that can enhance a robot's tracking algorithm. In this paper, we propose an approach to this problem that is based on the GOTURN architecture by Held *et al.* (2016). For our purpose, we used the GOTURN architecture and trained it to detect rotation changes.

3 Dataset Collection

Figure 1: An overview of the implementation



As can be observed in Fig. 1, the final implementation should be able to take in two frames, detect the amount of rotation deformation and ID of the person between these two frames. To achieve this, a labeled dataset with images labeled by their rotation angle and ID are needed. Since no labeled dataset was found, we had to collect our own data and perform some preprocessing steps to create a complete labeled dataset.

Stage 1. Collecting images. Our task requires images of people rotating in front of the camera. Two datasets were found: IAS-Lab RGBD-ID Dataset (Munaro *et al.,* 2014a,c), BIWI RGBD-ID data set (Munaro *et al.,* 2014a,b). These datasets are RGB-D images of people moving and rotating in a room used for long-term people re-identification.

Both datasets are labeled according to people ID which provides us with one of the labels required by our implementation. The skeletal information of each person in an image is also provided which is used in the next stage to divide the data according to rotation angle.

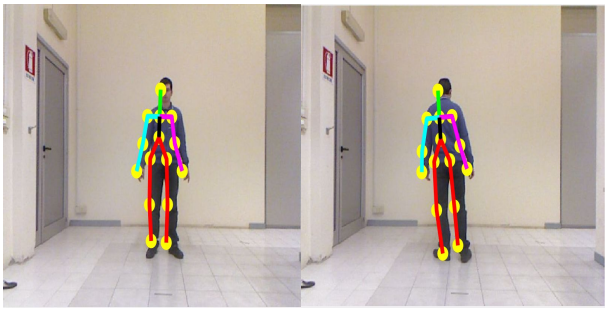
Stage 2. Filter images by rotation angle. Since the images extracted from the two mentioned datasets are only divided by ID and not by angle, we had to find an automated way to label them by rotation angle. To achieve this, information about the position of the left and right shoulder and the quaternion of the segment joining them to analyze their rotation.

This is first done by setting, for every label, a ground-truth quaternion which is compared to the quaternion of each person in an image. For every image, we calculate the angle difference between its quaternion and that of each label. To be considered part of this label, the difference should not exceed a specific threshold.

This method proved to be somewhat challenging as several problems were encountered. The

skeletal information provided does not take into account whether the person is facing the camera or not. This makes it difficult for the algorithm to separate supplementary angles such as 0^o and 180^o. This can be seen in Fig. 2.

Figure 2: Same alignment of right and left shoulder for two different rotations



To overcome this problem, we used a simple face detector implemented by OpenCV (Bradski, 2000) which detects the front and side of the face. Thus, when a match for a specific class is found, the image is then passed to the face detector to determine if the predicted class is correct and fixes it accordingly.

Stage 3. Pair and label data. Our use of the dataset requires the data to be paired together before inputing them to our model. There are many ways of pairing the data. The dataset is divided into two sub-datasets, positive and negative, each of which include a different combination of the pairs. The statistics of the different sub-datasets can be found in Table 1. An example of each combination is shown in Fig. 3 and Fig. 5.

Table 1: The statistics of the positive and negative datasets

Dataset	Number of Pairs
Positive Dataset	482,938
Same IDs + Different/Same angles	482,938
Negative Dataset	266,709
Different IDs + Same angles	33,577
Different IDs + Different Angles	233,132

Figure 3: Example from positive dataset

(a) Same ID + Same Angle

(b) Same ID + Same Angle

Figure 5: Example from negative dataset

(a) Different ID + Same Angle

(b) Different ID + Different Angle

4 Network Architecture and Implementation

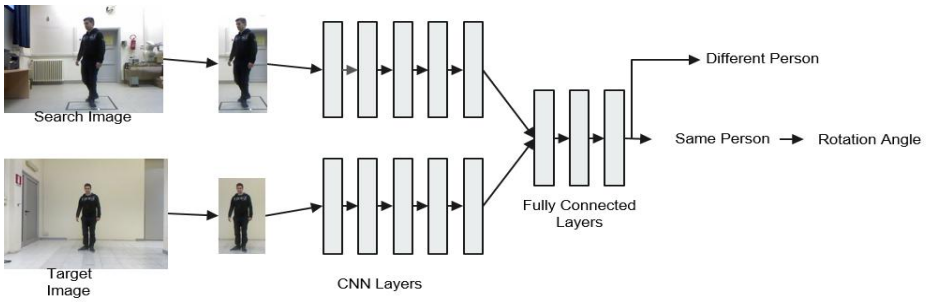
Our model requires an architecture that uses two images as inputs to compare them together and then outputs a specific label. Convolution neural networks have proven to be the best in analyzing images. Thus, for our initial prototype, we used the same architecture employed by the GOTURN algorithm in Held *et al.* (2016), which in turn is based on the CaffeNet architecture

(Jia *et al.,* 2014, Kuen *et al.,* 2016), but with a different loss function and final layer.

4.1 Architecture

As discussed before, the architecture shown in Fig. 7 is based on the architecture used in the GOTURN algorithm. Two images are inputed into the network that consists of 5 convolutional layers. The main role of these layers is to extract higher level representations of the image. These representations will provide the important features that are needed to compare the two images. The output of both convolutional layers are concatenated and inputted into a series of three fully connected layers. Their job is to learn the weights that best compare the two images to detect the rotation angle and ID.

Figure 7: Architecture of CNN used



4.2 Implementation

Our goal is to implement a model that is able to re-identify a person and regress the angle of rotation of a pedestrian being tracked by a machine.

The architecture is implemented using Tensorflow (Abadi *et al.,* 2015). Two images are read from the respective dataset and then passed to a pedestrian detector implemented by OpenCV (Bradski, 2000). The detector finds and crops the person in both images which are then inputted to the neural network to be trained. During training, the loss function for detecting rotation computes the softmax cross entropy between the real label and the predicted label. It follows the

equation below:

$$AverageLoss = \frac{1}{|B|}\sum_{X_i \in B} L_i \qquad where \begin{cases} X_i \in Sample\ Observation \\ L_i = -\ln\left(\sigma(X_i)\right) \\ B \subset Training\ Data \\ \sigma(X_j) = \frac{e^{X_j W}}{\sum_i e^{X_i W}} \qquad \text{(Softmax Equation)} \end{cases}$$

The model aims to minimize this loss during training.

5 Preliminary Results

For training and testing, we had two scenarios. In Scenario 1, we allowed the training and testing dataset to share similar IDs and backgrounds. However, in Scenario 2, we eliminated any possibility of any similar data between both datasets. The results of these two scenarios are shown in Table 2

Table 2: Preliminary Results showing that the model was able to learn the rotation but was over-fitting

Accuracy(%)	Scenario 1	Scenario 2
Detecting rotation angle and ID:	98.5219	30.7731
Detecting ID	98.7620	60.8278

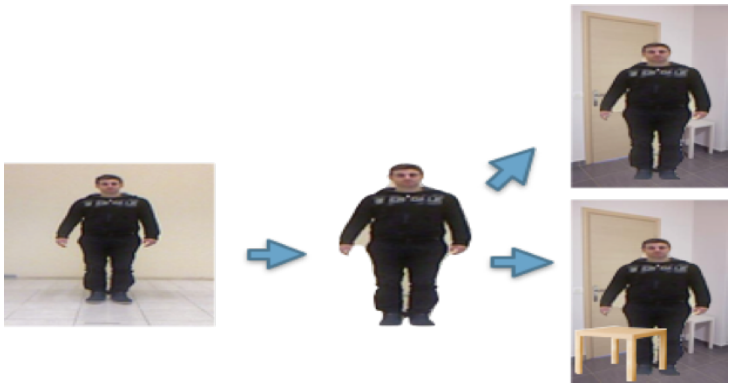
As a preliminary experimentation, we reached a significantly high accuracy of 98.5% in Scenario 1. This shows that the model is able to learn to detect angles and IDs. However, once the testing set was changed to contain completely different data than the training set (Scenario 2), the accuracy drastically dropped to 30.8%. This shows that the model is actually over-fitting on the noise found in the training dataset.

A solution for such a problem is to augment the dataset. Many techniques are available that

process the data to create more data which in turn helps the model generalize more:

- Adding noise
- Adding jitter
- Downscale + Upscale the image (Low-Pass filter)
- Image building (Fig. 8)

Figure 8: An example of image building



6 Future Work and Conclusion

In this work, we have addressed the problem of detecting rotational deformation encountered in real-life situations, which has not been addressed before. As a first step, we created our own dataset of images of pedestrians and trained a neural network to detect the presence of a rotation in a pair of images. We will then show whether the representation learned from this task can be used to improve the tracking process. The results will show us the effect of capturing rotation on the performance of tracking a person. An important aspect of the learning process that will effect the representation being learned is the loss function. When more nuisances will be added for classification, the loss function can be altered to include these new tasks. Various loss functions will be used to solve the detections of the different nuisances in the aim of better identifying the person. Some loss functions that can be used are hierarchal loss functions that account for meaningful organization of the classes.

In future work, we plan to test our implementation and study its performance in real-life scenario by integrating the algorithm into our lab's robot, Loomo. In addition, there are many other types of nuisances such as viewpoint changes, body deformation , object occlusion, and illumination changes that still need to be considered. Thus, our final goal is to extend our architecture to

recognize the different nuisances and improve the tracking process.

7 References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng (2015) TensorFlow: Large-scale machine learning on heterogeneous systems, `https://www.tensorflow.org/`. Software available from tensorflow.org.

Bradski, G. (2000) The OpenCV Library, *Dr. Dobb's Journal of Software Tools*.

Cheng, F.-C., B.-H. Chen and S.-C. Huang (2014) A background model re-initialization method based on sudden luminance change detection, **38**, 11 2014.

Held, D., S. Thrun and S. Savarese (2016) Learning to track at 100 FPS with deep regression networks, *CoRR*, **abs/1604.01802**.

Jia, Y., E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell (2014) Caffe: Convolutional architecture for fast feature embedding, paper presented at the *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, 675–678, New York, NY, USA, ISBN 978-1-4503-3063-3.

Kuen, J., K. Lim and C. Lee (2016) Self-taught learning of a deep invariant representation for visual tracking via temporal slowness principle, *CoRR*, **abs/1604.04144**.

Liu, B., H. Wu, W. Su, W. Zhang and J. Sun (2017) Rotation-invariant object detection using sector-ring hog and boosted random ferns, *The Visual Computer*, May 2017, ISSN 1432-2315.

Munaro, M., A. Basso, A. Fossati, L. V. Gool and E. Menegatti (2014a) 3D Reconstruction of Freely Moving Persons for Re-Identification with a Depth Sensor, paper presented at the *IEEE International Conference on Robotics and Automation (ICRA2014)*.

Munaro, M., A. Fossati, A. Basso, E. Menegatti and L. Van Gool (2014b) *One-Shot Person Re-identification with a Consumer Depth Camera*, 161–181, Springer London, London, ISBN 978-1-4471-6296-4.

Munaro, M., S. Ghidoni, D. T. Dizmen and E. Menegatti (2014c) A Feature-based Approach to People Re-Identification using Skeleton Keypoints, paper presented at the *IEEE International Conference on Robotics and Automation (ICRA2014)*.