Efficient Learning of Pre-attentive Steering in a Driving School Framework*

Reinis Rudzits Nicolas Pugeault†

December 24, 2014

Abstract

Autonomous driving is an extremely challenging problem and existing driverless cars use non-visual sensing to palliate the limitations of machine vision approaches. This paper presents a *driving school* framework for learning incrementally a fast and robust steering behaviour from visual gist only. The framework is based on an autonomous steering program interfacing in real time with a racing simulator (TORCS): hence the teacher is a racing program having perfect insight into its position on the road, whereas the student learns to steer from visual gist only. Experiments show that (i) such a framework allows the visual driver to drive around the track successfully after a few iterations, demonstrating that visual gist is sufficient input to drive the car successfully; and (ii) the number of training rounds required to drive around a track reduces when the student has experienced other tracks, showing that the learnt model generalises well to unseen tracks.

1 Introduction

Imagine you are driving on a countryside road. You steer the car keeping it on your side of the road, slow down in curves, shift gears accordingly. When another car is in front, you adapt your driving to match its speed, and if a wide truck comes in front, you correct your lane position to avoid collision. Now consider that you are crossing a town. In addition of these, you need to handle intersections, other cars, cyclists and pedestrians, traffic lights and signs. All of these tasks are performed routinely and effortlessly by experienced drivers, even to the point that driver inattention is named as a major cause for road accidents. In contrast, and despite significant progress over the last decades, artificial systems struggle to approach human performance in any of these tasks, let alone all of them. Early approaches in autonomous driving can be traced back to the 70's (*e.g.*, VITS [12], ALVINN [7], see [5] for a review) up to state-of-the-art systems such as the Stanley robot [11], the Google car[1] or Oxford's RoboCar UK[2]—for example the UK is planning to test them in selected cities as early as 2015. Yet today, all working systems must rely on extensive arrays of sensors (LIDAR, GPS, aerial images, road maps) to palliate the insufficiencies of machine vision. Despite the difficulties it entails, vision based driving remains the most promising way to spread driverless cars outside of city centers and motorways to rural areas and countryside roads: first because most of the existing infrastructure and signalisation in place is visual; second, vision offers the advantage of a fast, passive sensor that allows for early detection of hazards. Unfortunately, vision-based autonomous driving is still far from our reach, and significant progresses in machine vision are required.

*This article appeared in *KI - Künstliche Intelligenz*, 2014, `http://dx.doi.org/10.1007/s13218-014-0340-1`. The final publication is available at `link.springer.com`.

†Corresponding author. Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH, Guildford, UK. `n.pugeault@surrey.ac.uk`

[1] `https://plus.google.com/+GoogleSelfDrivingCars/posts`

[2] `http://mrg.robots.ox.ac.uk/robotcar/`

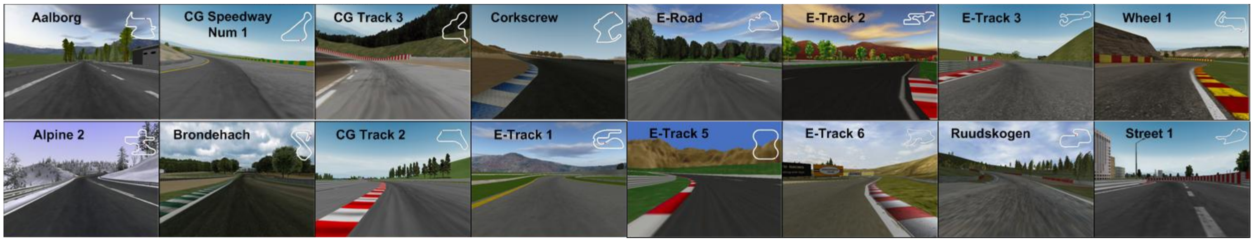


Figure 1: The tracks from the TORCS simulator that were used in this work.

One aspect that sets vision-based driving aside from other computer vision problems, is the vast amount of visual information that is continuously perceived and the high variability of the environment: driving differs in cities versus countryside or motorways, lane markings can be faded or nonexistent, and it is impractical to predict a priori all sort of road users that a car may encounter (cars, motorcyclists and trucks, but also pedestrians, cyclists in London, sheep in Scotland, wild poneys in Dartmoor, etc.). In addition, any driving system is required to analyse visual information swiftly in order to act with minimal latency—a delayed response when driving at 60mph can have lethal consequences. The mainstream of computer vision approaches, for example running detectors over scanning windows in images, require high processing power and is ill suited to this low-latency requirement.

In contrast to these approaches, the recent availability of large image collections on the internet has motivated computer vision scientists to look for faster approaches. Drawing inspiration from the pre-attentive capabilities of the human visual system, Oliva & Torralba proposed a holistic image descriptor based on the low frequency components of images' Fourier decomposition, to encode the *gist* of a visual scene [6]. They demonstrated that such a coarse holistic encoding was sufficient for classifying images between categories such as open or closed, indoor or outdoor... Because such gist descriptors discard all high frequency components, they are a plausible approximation of human pre-attentive perception based on peripheral vision. In addition to this, gist descriptors are computationally attractive because they only process low-resolution versions of the images, which makes them well suited for low-latency applications such as driving. The first application of visual gist in robotic domain was by Siagian & Itti [10], where visual gist was used for robot localisation (but not steering). Pugeault & Bowden [8] demonstrated that detectors based on visual gist could predict driving context such as 'crossroads', 'urban' versus 'non-urban' and 'single-lane' versus 'motorway'. Additionally, it was shown that the driver's actions such as pressing the brake pedal or turning the steering wheel could also be predicted to a large extent. A later article [9] showed that a human driver's steering angle could be predicted accurately using a Random Forest regression approach on the extracted gist descriptors. The results were demonstrated on a robotic platform that could steer itself at moderate speed on a narrow track and on driving data recorded in a real car on a countryside road. These studies demonstrated that (i) the majority of human driving actions can be explained from visual gist only, and (ii) such models have the capacity to adapt to large variations in the environment: lane markers were learnt as cues when visible, whereas other cues were learnt otherwise [9]. Both of these works used batch learning approaches however, where all training examples were provided at training time. This is a limitation because of the extensive variability in driving environment and situations, and missing training examples can cause the system to steer off the road. To make matters worse, informative situations, such as taking a narrow curve or a hairpin bend are uncommon which means that hundreds of hours of driving may be required before all informative situations are featured in the training set. This is impractical both from the perspective of data storage and training algorithms.

In the following, we overcome this limitation by devising a driving school in a racing simulation program. In this approach, the 'student' is the pre-attentive driving program, and the 'teacher' is another program that has perfect knowledge of the track and how to steer around it—and does not rely on vision for this.

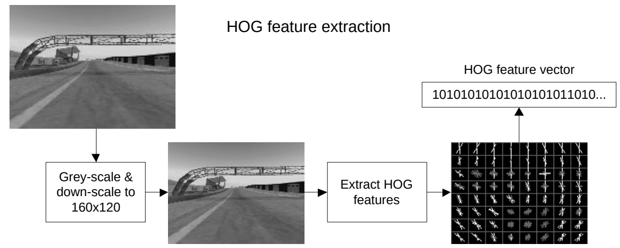


Figure 2: Visual gist extraction (HOG) from the car's cockpit.

2 Introducing the virtual driving school

In order to implement a driving school for an artificial vision-based program, two main challenges need to be addressed: safety and training. First, in a real-life driving school, the risk of giving the wheel to an untrained driver is alleviated by the assumption that the learner has enough common sense and caution. There is no such comfort with an algorithm: assessing algorithm performance on a racing track at a speed of 70 mph under varying environmental conditions would pose significant safety issues—notwithstanding material damage. Second, a driving school requires an experienced driver, the teacher, to take over and correct the student when he does wrong. Although we could imagine to use a human teacher, it would be easier and faster for the system to learn from *another* program that is known to be a good and safe driver and using additional sensors beside vision and world knowledge. For these reasons, we devised our driving school experiment inside a virtual environment using an artificial driver as tutor.

2.1 Simulated environment

In this work, we build our driving school on an Open Source racing simulator, called TORCS [13]. The video game engine provides with a convenient perception/action framework as well as performance information such as the car's position on the road. Additionally, given that we are only using visual gist as input, the 3D graphics are virtually undistinguishable from real life videos. Using an Open Source program allows to easily modify the simulator's code in order to (i) transmit visual data and controls to the driving program, (ii) override the player's steering control with our program, and (iii) to check for failure cases by controlling for the position of the car on the road, and override the student's controls with the teacher's when this happens.

TORCS provides us with realistic car and road physics and a variety of racing tracks : Figure 1 shows the selection of 16 tracks that we used in this experiment, featuring large variations in weather, landscape, layout, difficulty and markings. Finally, TORCS also provides automated racing programs (or 'bots'), that steer the car around the track based on perfect knowledge of the track shape and position on the road. In this project we use a bot developped by Bernhard Wymann, called 'damned', as the teacher for our system [13]. Note that this work only considers the problem of high speed steering for road following; therefore in all experiments described in this paper the system was racing on an empty track with no other cars or obstacles (as would an actual driver when training). Also, anyone could drive around a racing track at low speed, as it allows the driver plenty of time to correct errors and poor steering choices. In contrast, high speed driving require quick and accurate steering to remain on the track, challenging even expert drivers. Hence, we set the car's speed at 70mph in all experiments, to ensure that successful driving around the track is an indicator of skillful steering, while ensuring that an average human player could also succeed.

2.2 Visual gist

As explained before, we want our system to learn a model of pre-attentive driving, and therefore we ensure that it only perceives the visual gist of the view from the car's cockpit, as in [8, 9]. In [9], two descriptors were considered for regressing steering: the GIST descriptor from [6] and the related Histogram of Oriented Gradients (HOG) descriptor [3]. HOG is a popular feature descriptor in machine vision for its efficiency, originally designed for human detection in images. It consists in calculating the image gradients at all pixel localisation and pooling gradients' orientations into histograms over a coarse grid. In this work we make use of HOGs descriptors calculated on 8×8 grids over 160×120 images, which is an efficient approximation of visual gist [9] allowing us to process the simulator's visual input at 20 frames per second—see Figure 2. Formally, we define our visual gist extration from images $I \in \mathcal{I}$ as a process $\phi : \mathcal{I} \to \mathcal{X}$, where $\mathcal{X}$ is the space of our HOG features.

2.3 Steering angle regression

The learning is performed using random regression forests, as in [9]. Random forests were initially proposed by [1, 2], and consist in learning a randomised collection of decision trees from a dataset. Random forests can operate regression from large datasets with high dimensional visual gist observation vectors, they are efficient to train and can predict steering values in real-time, and therefore ideally suited for this study. Formally, a random forest regressor ψ is a mapping from the space of visual features $\mathcal{X}$ to a steering angle $\theta \in \mathbb{R}$: $\psi : \mathcal{X} \to \mathbb{R}$. A steering forest ψ_D will be defined from a training set D composed of N pairs (x_i, θ_i) where $x \in \mathcal{X}$ is a HOG feature extracted at time t_i and $\theta_i \in \mathbb{R}$ is the corresponding steering value decided at time t_i by the teacher program. In practice, random forest training time typically increases superlinearly with the number of training examples N, hence the aim of our approach is to add example pairs to D parsimoniously. We refer to [9] for details of the random forest learning approach.

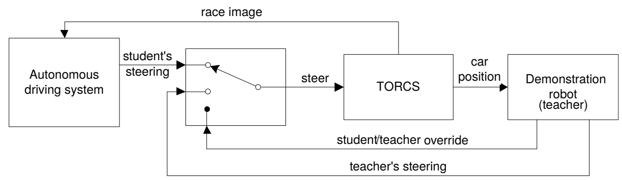


Figure 3: Diagram of the driving school framework built on top of the racing simulator.

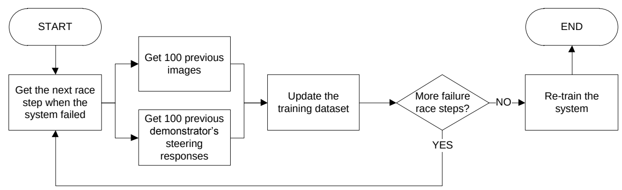
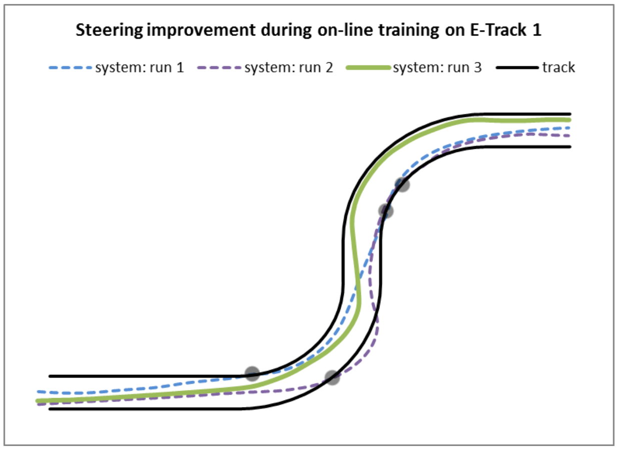


Figure 4: Flow diagram of the incremental learning approach.

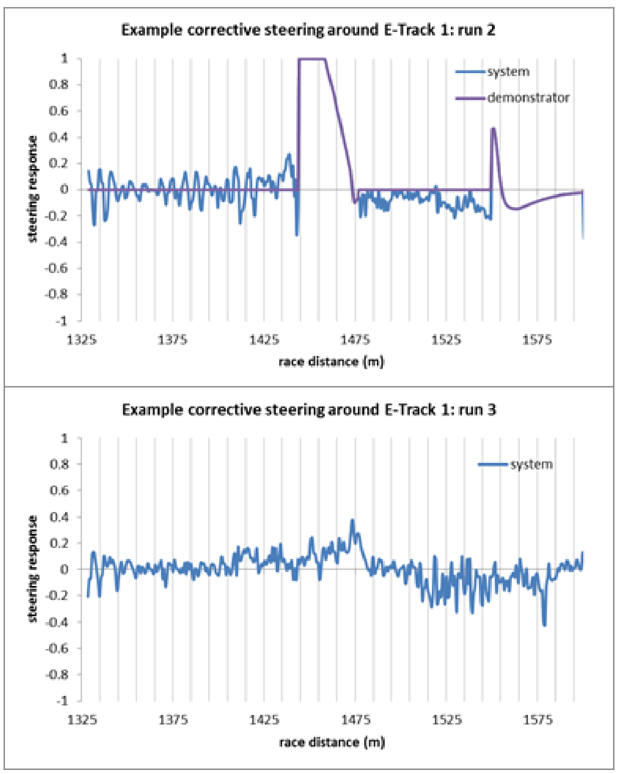
2.4 Incremental learning

This work focuses on incremental re-training by adding failure cases to the training set after each race. Incrementally adding failure cases to update a learnt model is a common approach in visual tracking—see, *e.g.*, [4]. The rationale is that the object's appearance is likely to change over time due to lighting and viewpoint, and that it is impossible to define a complete set of training examples *a priori*. Therefore a more successful approach is to update the model of the tracked object at each frame with new success and failure cases. The same rationale is used in this work: we start by learning a simple random forest regression for a minimal training sample (one race around the track demonstrated by the teacher), and update this training set after each race by adding the failure cases.

In our experimental set up, the car speed is fixed at 70mph, and the steering can be controlled either by the student, using the learnt random forest regressor, or by the teacher using TORC's bot program: the student controls the steering alone until the car gets off the road, then the control is passed onto the teacher to correct the car's position—this is illustrated in diagram 3, where the 'student/teacher



(a) trajectories of the student



(b) steering control

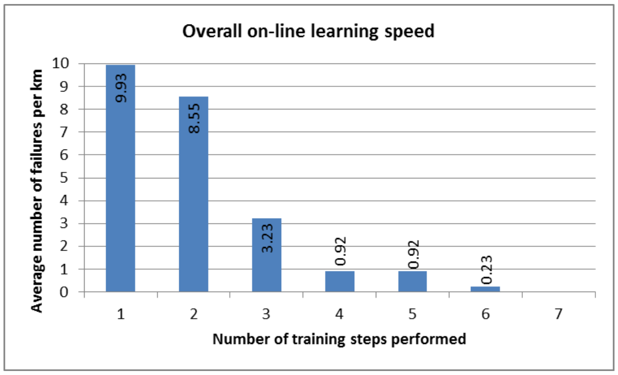
Figure 5: Student learning on a single curve: In (a) the trajectories followed by the system after one, two or three rounds of training, on the same curve. Each time the trajectory touches the road boundary (denoted as shaded circles on the figure), the teacher took over the steering and brought the car back on the road, hence runs 1 and 2 both failed twice, and run 3 was successful. Panel (b) shows the actual steering control for runs 2 and 3 on this curve: the blue curve is the student's control and the purple one is the teacher's correction when the student failed.

override' signal switches the car's control from student to teacher when the car goes off track. In addition to correcting the car's position, the teacher then provides feedback to the student in terms of correct steering controls in the 5 second that preceeded the incident. This duration of 5 seconds is arbitrary and chosen because at a speed of 70mph, it will include most, if not all, steering information leading to the failure (it corresponds to a distance of $\simeq 155$ meters). The system's performance is unaffected by small variations of this parameter, although shorter episodes will require more retraining while significantly longer episodes will increase the training set with less relevant examples and slow down each re-training.

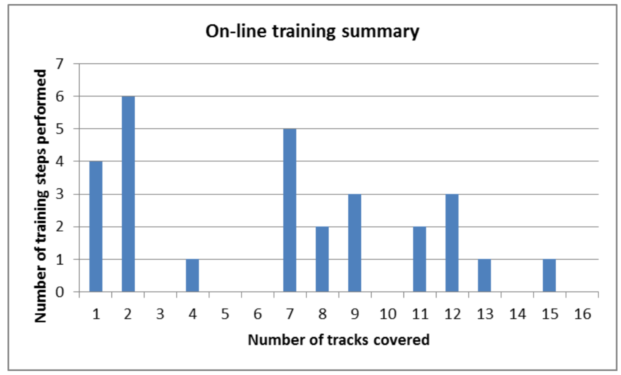
In practice, everytime the student drives the car off the track: (1) the 100 frames (*i.e.*, 5 seconds) prior to the failure are saved as a new training episode, along with the steering suggested by the teacher for these frames; (2) the teacher takes over and brings the car back on the track; and (3) the student resumes driving around the track—this is illustrated in diagram 4. When the student finally finishes racing around the track, all failure cases and associated teacher's steering are added to the training set of the student's regression model, and the random forest is re-trained. Both stages of experimenting and training are performed iteratively on the same track until the student manages to race 10 consecutive laps without mistake. Figure 5 illustrates the control signal generated by the student (b), along with the corresponding trajectories (a) for one curve.

3 Results

The experiment were set the following way: First, the teacher demonstrates racing one lap around track #1; then, the student program is given the controls and attempts to race around the track, acquiring new training data after each mistake. After the end of the lap, the student is retrained before attempting again. When the student succeeds in driving 10 consecutive laps around the track, it proceeds to the next track.



(a) average failures on known track



(b) training steps for adapting to new tracks

Figure 6: Incremental learning of the student driver: (a) average number of student failures (ie, going off track) against the number of laps and re-training performed; (b) number of training steps required when discovering a new track before driving without mistake. Note that the first column indicate the number of re-training required on the same track on which pre-training was performed.

3.1 Learning to race on a known track

Figure 6(a) shows the average number of times the student went off the track per km, after each training stage. This graph shows that the incremental retraining improves steering behaviour quickly and that the system can drive successfully around the track after 6 re-trainings. Figure 7 provides a more qualitative insight in the student's steering improvement due to re-training. In this figure, the bottom row shows the trajectories at the first trial, and after four re-training stages, and the top row shows the intermediate trajectories, after one two and three re-trainings. In these figures, the student's trajectory is denoted by the green line and the failure cases where the teacher had to intervene are denoted by the purple segments. The student originally fails in all curves (and even some straight portions), whereas a single failure occur after just two re-training steps. After four re-trainings, the student successfully drive around the track as shown by the lack of purple segments.

3.2 Generalisation to new tracks

One question is whether the behaviour learnt by the student generalises to different tracks, surfaces, environments, lane markings, etc. Figure 6(b) provides some evidence that this is the case. This graph plots the number of retraining stages required before the student could drive successfully around the track, against the number of tracks previously encountered by the student (the tracks were attempted in the same order as Figure 1, starting from top-left to bottom right). This shows that training generally becomes faster after the student has already mastered several tracks, although some more difficult tracks still require re-training. For example the student could drive around the third, fifth and sixth track without re-training, *i.e.* the training examples collected from the previous tracks were sufficient to inform driving on these ones. Overall, six tracks could be raced without additional training, and three more with only one additional training round. Conversely, some of the tracks seem to be intrinsically more difficult to learn, for example tracks 7 and 8 contain tighter curves than the previous ones, track 9 is the only track in a snowy weather, hence the need for re-training.

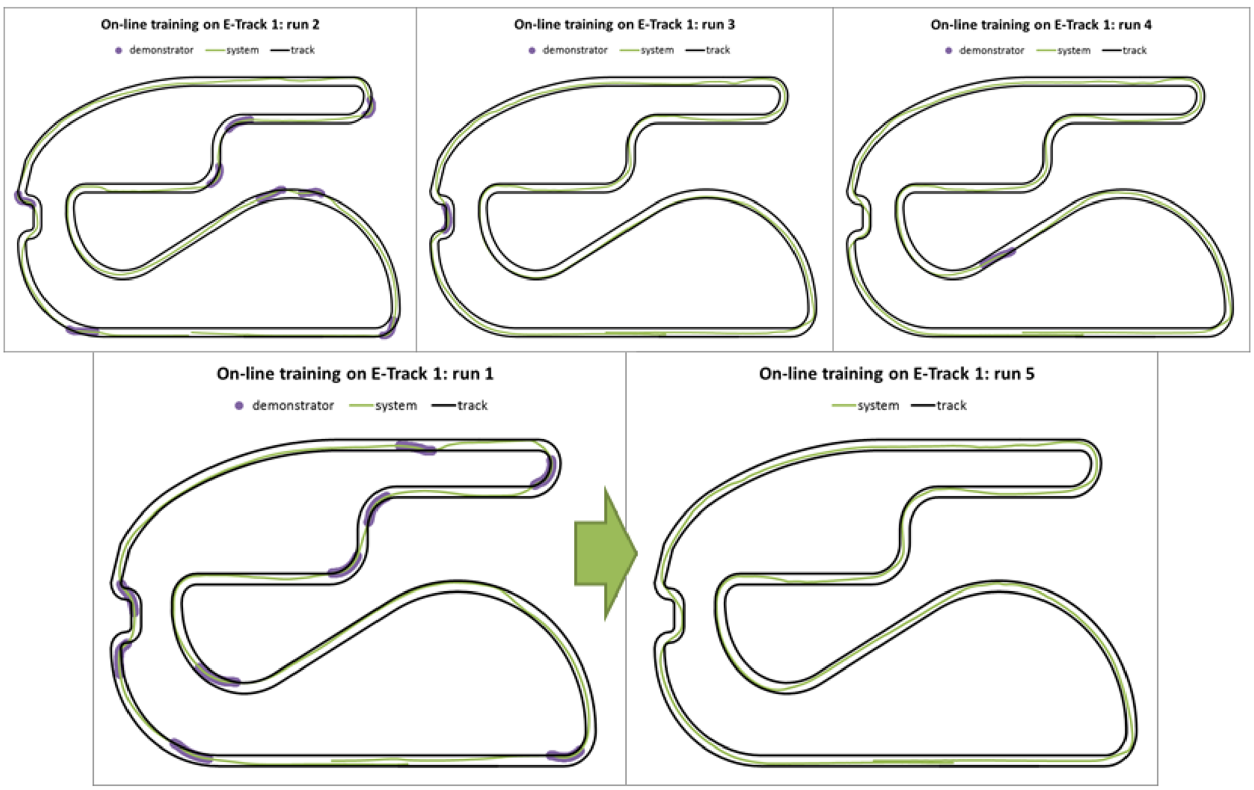


Figure 7: Illustration of car trajectories across successive trials and re-training stages. The green line is the trajectory, and the purple sections indicate the student's failures and the episodes selected for re-training.

4 Conclusions

In this article we have presented some results of a simulated driving school framework for artificial driving systems. The studied system used only visual gist as perceptual input and achieved a good approximation of an expert program's driving behaviour, despite the fact that the program in question benefit from perfect knowledge of the track and the car's position on the road. This is a confirmation of the evidence in [8, 9] that pre-attentive vision accounts for a large share of driving behaviour.

In addition, we presented an incremental learning approach that only sparingly increase the training set with failure episodes and is shown to reach good performance after only a few re-training stages. Moreover, the learnt behaviour is shown to generalise well to different tracks. The re-training stages could be improved in future work by implementing a variant of boosting instead of retraining the whole forest at each step.

Finally, this experiment showed that simulated environments can provide convenient test beds for learning, and especially for active learning approaches.

References

[1] Amit Y, Geman D (1997) Shape quantization and recognition with randomized trees. Neural Computation 9:1545–1588

[2] Breiman L (2001) Random forests. Machine Learning 45(1):5–32

[3] Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[4] Kalal Z, Mikolajczyk K, Matas J (2010) Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 6(1)

[5] Markelic I, Kulvicius T, Tamosiunaite M, Wörgötter F (2008) Anticipatory driving for a robot-car based on supervised learning. In: ABiALS, pp 267–282

[6] Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. International Journal of Computer Vision 42(3):145–175

[7] Pomerleau D (1989) Alvinn: An autonomous land vehicle in a neural network. In: Proceedings of NIPS

[8] Pugeault N, Bowden R (2010) Learning pre-attentive driving behaviour from holistic visual features. In: Proceedings of the European Conference on Computer Vision (ECCV'2010), Part VI, LNCS 6316, pp 154–167

[9] Pugeault N, Bowden R (2011) Driving me around the bend: Learning to drive from visual gist. In: 1st IEEE Workshop on Challenges and Opportunities in Robotic Perception, in conjunction with ICCV'2011

[10] Siagian C, Itti L (2009) Biologically inspired mobile robot vision localization. IEEE Transactions on Robotics 25(4):861–873

[11] Thrun S, Montemerlo M, Dahlkamp H, Stavens D, Aron A, Diebel J, Fong P, Gale J, Halpenny M, Hoffmann G, Lau K, Oakley C, Palatucci M, Pratt V, Stang P, Strohband S, Dupont C, Jendrossek LE, Koelen C, Markey C, Rummel C, van Niekerk J, Jensen E, Alessandrini P, Bradski G, Davies B, Ettinger S, Kaehler A, Nefian A, Mahoney P (2006) Stanley: The robot that won the DARPA Grand Challenge. Journal of Robotic Systems 23(9):661–692

[12] Turk M, Morgenthaler D, Gremban K, Marra M (1988) VITS—a vision system for autonomous land vehicle navigation. IEEE Transactions on Pattern Analysis and Machine Intelligence 10(3):342–361

[13] Wymann B, Espié E, Guionneau C, Dimitrakakis C, Coulom R, Sumner A (2013) TORCS: The open racing car simulator, v1.3.5

Reinis Rudzits graduated with first class BEng in Electronics Engineering at the University of Surrey (UK) in 2014. He is currently employed as a financial software developer at BNP Paribas, London.

Nicolas Pugeault is a Lecturer at the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey (UK). He holds a Ph.D. from the University of Göttingen, and has co-authored over 40 peer-reviewed publications. His research covers aspects of computer vision, machine learning and cognitive robotics.