Towards Intelligent Social Robots: From Naive Robots to Robot Sapiens
Amir Aly, Sascha Griffiths, Verena Nitsch, Katerina Pastra, Tadahiro Taniguchi

To cite this version:
Amir Aly, Sascha Griffiths, Verena Nitsch, Katerina Pastra, Tadahiro Taniguchi. Towards Intelligent Social Robots: From Naive Robots to Robot Sapiens. International Conference on Intelligent Robots and Systems (IROS), Oct 2018, Madrid, Spain. 2018. <hal-01953346>

HAL Id: hal-01953346
https://hal.archives-ouvertes.fr/hal-01953346
Submitted on 22 Dec 2018
Teaching robots to imitate a human with no on-teacher sensors. What are the key challenges?

Radoslav Skoviera¹, Karla Stepanova¹, Michael Tesar¹, Gabriela Sejnova¹, Jiri Sedlar¹, Michal Vavrecka¹, Robert Babuska¹,², and Josef Sivic¹,³

Abstract—In this paper, we consider the problem of learning object manipulation tasks from human demonstration using RGB or RGB-D cameras. We highlight the key challenges in capturing sufficiently good data with no tracking devices—starting from sensor selection and accurate 6DoF pose estimation to natural language processing. In particular, we focus on two showcases: gluing task with a glue gun and simple block-stacking with variable blocks. Furthermore, we discuss how a linguistic description of the task could help to improve the accuracy of task description. We also present the whole architecture of our transfer of the imitated task to the simulated and real robot environment.

I. INTRODUCTION

Imitation learning has a long history and many applications ranging from easier programming of industrial robots to household companions. Nonetheless, in practice, it still heavily relies on tracking devices and optical motion tracking [1] accompanied by markers on objects which help to identify 6D pose of the given object. If we want to develop social and cognitive robots which are able to learn from a direct interaction with humans and imitate or recognize their actions, we have to develop teaching methods which will not require any on-teacher sensors and will be able to learn in a natural environment (e.g., household). These robots will have to learn from direct observation (visual, linguistic, haptic) using only their own sensors and previously acquired knowledge. Our biggest concern is, whether current hardware and machine learning methods enable this type of imitation learning. Can recent rapid progress in computer vision and natural language processing make omitting cumbersome, task-specific sensing devices possible and enable robots to really understand the scene and reality they are observing?

Imitation learning first became an object of interest in the early 1980s as a possible way towards higher autonomy in industrial robots. The initial approach was manual operation of the robot, such as the teach-in method, guiding or a play-back method. The demonstrated task was represented as a series of transitions between states and actions, which were further converted into a set of graph-based symbolic rules and relationships [2]. Ever since, most of the progress happened in the field of teaching methods, varying from vision to kinesthetic teaching, where the robot is physically manipulated to perform the desired task. Remote teaching typically includes various on-teacher sensors. Argall et al. [3] tried to use only sensors on the robot to mimic human behavior. There are also some attempts to detect human activity from RGB-D sensors [4] as well as from RGB narrated videos [5]. Visual demonstration accompanied by language instructions was also used in [6] where they tried to learn grounded task structures for T-shirts folding task. Mühlig et al. [7] tried to teach a robot manipulation block-stacking task from a tutor sitting behind a table. To our best knowledge, there is no work done on learning from demonstration relying only on visual, linguistic and haptic information without any on-teacher sensors for complex manipulation tasks in a real-world environment.

In the following sections, we describe the challenges of capturing data for the purpose of imitation learning. We propose some solutions to the tasks based on the currently available methods and software packages. We also present our preliminary findings of the performance of the state of the art methods. The findings were gathered while implementing imitation learning architecture presented in Fig. 2. In Fig. 1, we present the setup of our experiment and also manipulation tasks on which we evaluated the compared approaches.

¹Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University, Prague, Czech Republic
²TU Delft
³Inria, WILLOW, Departement d’Informatique de l’Ecole Normale Superieure, PSL Research University, ENS/INRIA/CNRS UMR 8548, Paris, France
II. EXPERIMENTAL SETUP

The basic setup for the data acquisition consists of a table-desk with a calibration checkerboard, two Asus Xtion cameras with depth sensors, a high-resolution RGB camera and HTC Vive VR set. The data from all sensors are broadcasted via the Robot Operating System (ROS)\(^1\) framework. Both Xtion sensors produce 640x480 RGB-D images at 30Hz, the RGB camera produces 5MP images at 10Hz, and HTC Vive captures the position of the controller at 60Hz. The object images, depth maps, segmentation masks and 6DoF information are extracted from raw data after calibration and time synchronization.

III. CHALLENGES

A. Imaging Sensors

An important part of the imitation learning setup is an imaging sensor. The choice of the sensor depends partly on the algorithms used to extract information from the scene. Some algorithms may require just a simple color camera. However, if increased positional accuracy is required, as is usually the case in industrial tasks, depth information may be necessary as well. Depth can be obtained via several types of sensors.

**Stereo vision (SV):** Stereo vision is a well-known technique for obtaining 3D information of the sensed scene [8]. Stereo vision offers high image resolution and visual information can be obtained at the same time. The downside of stereo vision is that the field of view (FOV) is often narrow and the depth resolution is dependent on the length of the baseline, which is usually fixed. Also, a lot of computational power is required to obtain corresponding points from individual cameras, for which depth can be computed (i.e., the correspondence problem).

**LiDARs:** LiDARs use laser ranging technique to obtain distance information for points in the scene. They offer high accuracy and depth resolution, long range of detection, and wide FOV. However, they are also quite expensive. Additionally, they require temporal and spatial synchronization with the visual sensor to provide RGB-D image.

**Time of flight cameras (ToF):** ToF cameras use active illumination and the distance is measured from the reflected light. It can be computed either directly from the travel time of a pulse of light or indirectly from the phase shift of a modulated light [9]. Depth information can be calculated with much less computational power. Hence, ToF cameras can provide depth information with a higher rate than other 3D imaging technologies. They are therefore suited for application where fast moving objects might occur. However, the hardware for ToF cameras is quite expensive as high-speed electronic components are required. Moreover, the sensors usually have low image resolution.

**Structured light sensors (SLS):** The biggest boom of cheap consumer 3D imaging sensors came in the form of structured light sensors. These active sensors project a structured light on the scene [10]. The depth information is then calculated from the pattern distortion caused by capturing it from a shifted viewpoint and the scene structure. The advantage of SLS cameras is their low price and ease of use. The disadvantage is lower accuracy with a relatively small working range.

There are also general considerations related to image capture. A key consideration is a computational power and transport capacity of the hardware used with the sensors. High resolution images with high update frequencies will require broadband connection. This is particularly crucial if real-time processing is required and might limit sensor placement in a real environment.

Another consideration is the ambient light. Because direct sunlight is typically many times stronger than the illumination used in active sensors, many of them (mainly ToF and SLS) are not suitable for outdoor applications. A surface of the sensed objects also plays an important role. Highly reflective or too absorptive surfaces might distort the depth measurements but may also pose a problem in object classification from RGB images. A simple solution for an undesirable surface might be masking the objects with a opaque colored tape.

Most of the 3D imaging cameras have distance-dependent measurement accuracies. It is important to assess the working region of depth measuring, i.e., the region with sufficient

---

\(^1\)http://www.ros.org/
depth accuracy. For most cameras, the working region starts at several dozens of centimeters. For the SV cameras, the extent of their working region depends on their geometry (baseline). The commercially available SLS cameras usually work with high enough accuracy only up to about 1 meter, with a recommended working region between 50 and 80 cm [11] even though detection range can be several meters.

One should also keep in mind that interference from multiple active sensors can occur. Some sensors can use different light frequencies or modulate the light to prevent interference. However, the performance of cheaper consumer sensors will likely be degraded in the presence of other active sensors. Acquisition speed and frame rate should also be considered. Fast moving objects captured by sensors with slow acquisition speed will result in motion blur in the color image and measurement errors in the depth image. For the gluing task the high frame rate and good image and depth resolution is more important than in the block-stacking task.

B. Sensor calibration

Calibration is an important step when preparing a visual imitation learning setup. Basically, all of the calibration procedures are based on finding corresponding points in the respective co-ordination systems and computing the transformation between them. The basic algorithms are usually contained in most popular software packages, such as OpenCV\textsuperscript{2} or ROS. Although, for higher accuracy, more sophisticated calibration should be done as these packages do not implement state of the art methods.

First, intrinsic camera parameters, such as focal length and distortion model, must be calibrated. Higher grade cameras are often supplied pre-calibrated by the manufacturer. For less expensive cameras, it is usually possible to find approximate intrinsic parameters for the specific camera model. However, it is advised to perform the calibration manually. As a result of the manufacturing process, there are always (small) differences between individual cameras even of the same make.

Extrinsic camera calibration is used to localize the camera with respect to the scene. It is necessary to correctly position the objects detected from the camera image in the scene. Extrinsic calibration is crucial when using multiple cameras and other sensors, such as LiDARs. Extrinsic parameters of individual sensors are used to fuse information captured by each sensor. When fusing information from multiple sensors, temporal synchronization is also important. Especially, when objects in the scene are expected to move at higher speeds. There are several methods to achieve temporal synchronization for imaging sensors. Authors of [12] use the rolling shutter effect combined with a short burst of light while authors of [13] use trajectory of a moving object.

Hand-eye calibration is important as well when developing and testing methods for imitation learning. Hand-eye calibration is normally used in robotics to calculate mapping from camera to robot coordinate system. In the context of imitation learning, it might be mapping from camera to the coordinate system of a tracking device capturing the ground truth.

C. Data preprocessing

Before object detection and any other advanced processing can occur, the raw data captured by the sensors should be preprocessed. Raw image data coming from cameras should be at least rectified to remove lens distortion. Additional preprocessing may suppress effects such as motion blur. Pre-segmentation and background suppression may also be beneficial for the forthcoming processing steps.

As many methods for object classification or object pose estimation are susceptible to clutter, cropping the image as much as possible is advised. This can be done either by manually selecting the working area (e.g. crop out anything besides the working table). Alternatively, fast and robust methods for semantic segmentation or bounding box detection could be performed, such as the Mask R-CNN [14]. Afterwards, only areas believed to contain the objects of interest can be sent further down the processing pipeline. This can reduce the computational complexity and number of false positives in case of more complex scenes.

Depth images can have missing depth for some pixels or contain erroneous values. These are results of either camera construction (e.g. shift of multiple cameras for SV or camera and projector for SLS) or the environment (lighting conditions, surface properties). For algorithms that can handle occlusions well enough, this might not pose any significant difficulties. However, if the used algorithms are susceptible to these artifacts, there are several methods to deal with them. One option is to use simple interpolation (see Fig. III-C). Bicubic or bilinear interpolations produces smoother gradients on the surfaces of single objects. These methods, however, also introduce unwanted effects, such as smooth transition between object and a distant background. Nearest-neighbor seems to produce better results in this case. Alternatively, more advanced smoothing can be used. For example, limiting maximum gradient of filled image patches can be used to filter out smooth transitions between objects. There are also deep network based approaches for depth image reconstruction [15].

D. Training data collection

While gathering the training set for object detection and pose estimation, uniform sampling of the 6D pose space is important. Especially important is uniform and dense sampling in the rotational subspace; positional invariance is
However, in real-world and specially industrial environments as SIFT [17] or SURF [18] produce reasonable results. Methods based on matching local invariant features such as SIFT [17] or SURF [18] produce reasonable results. When objects are richly textured, reality, 6DoF object pose estimation has recently attracted significant attention [16]. When objects are richly textured, methods based on matching local invariant features such as SIFT [17] or SURF [18] produce reasonable results. However, in real-world and specially industrial environments objects often lack distinctive texture. To address this issue, [19] treat the 6DoF pose estimation from RGBD data as a (sliding window) template matching problem with different templates corresponding to different object orientations. An alternative to the template-based techniques are dense matching techniques [20] or sparse point clouds to compute point pair features (a relative position and orientation of two points) [21]. The point sparseness results in a much faster performance and also produces very good results in noise, clutter and partial occlusions.

In some cases, the depth information is distorted (in presence of specular materials or direct sunlight) or simply not available. Therefore it is useful to have solutions for 6DoF pose estimation from RGB images only. BB8 method [22] applies segmentation on RGB images to detect objects first in 2D and then predicts their 3D pose with Convolutional Neural Networks (CNN). This produces state-of-the-art results on the LINEMOD dataset and maintains good performance in cluttered images. An alternative is recent PoseCNN method [23].

For the purposes of our task we utilized template matching method of Hodan et al. [19] and BB8 method [22] for RGB images. Mask R-CNN was also helpful for 2D shape detection for the BB8 method; the results for 6D pose estimation by BB8 (without refinement) were rather poor, however (especially in presence of occlusion). Preliminary results can be seen in Fig. 5.

In our experiments, we encountered high sensitivity to the quality of the depth data and its synchronization with the image data. This is the main problem in the case of the quick movements of the objects, i.e., for the tool manipulation tasks, since even a small distortion in the 6DoF pose estimation of the whole object will cause a significant position error of the tip of the tool. Especially for Hodan method we observed a significant improvement in detections when we used segmentation from Mask R-CNN to cut region of interest from the image. Mask R-CNN was also helpful for 2D shape detection for the BB8 method; the results for 6D pose estimation by BB8 (without refinement) were rather poor, however (especially in presence of occlusion, as can be seen in Fig. 5). On top of these, as known, BB8 method has

### E. Object detection

Today, object detection and classification had progressed very far. Thanks to the advent of convolutional neural networks (CNN) detections can be fast and reliable even in complex scenes. The disadvantage is that they require large amounts of training data and are computationally expensive to train. For our task, we use the Mask R-CNN algorithm [14]. Results of the detection for the cube-stacking task can be seen in the Fig. 1, results for the tool manipulation task can be seen in the Fig. 5. Our observation is that occlusions are problematic for the method in case of more complex objects – in several frames, the object was not properly segmented. This might be resolved either by temporal stabilization, e.g., filtering of the position, or employing a tracking algorithm.

### F. 6Dof pose estimation

Motivated by applications in robotics and augmented reality, 6DoF object pose estimation has recently attracted significant attention [16]. When objects are richly textured, methods based on matching local invariant features such as SIFT [17] or SURF [18] produce reasonable results. However, in real-world and specially industrial environments the current state of the art methods.

![Image 4](image4.png)

Fig. 4. Sample image from the training of a used tool (gluegun). A) a gluegun in a random position on a green background (red borderline visualizes the mask boundary). B) binary mask segmented by color thresholding. The 6DOF pose information is stored in a text file for each frame.

Manual annotation of the training data is often time consuming. This can be partially solved by capturing the trained object on an easily separable background (see Fig. 4). To train the object on more complex backgrounds, thus gaining robustness to clutter, the images can be augmented with random backgrounds from a suitable image database. Capturing the pose ground truth can be done using traditional motion tracking techniques, such as optical motion tracking using markers. In our case, we attached the controller of a virtual reality gaming set to the tool for simple automatic 6DoF pose annotation.

![Image 5](image5.png)

Fig. 5. Detection of the tool. Left: MaskRCNN gluegun detection for the activity (testing) data (red shows the found segmentation of the tool, blue is the found bounding box). Note the problem when detecting the partly occluded object. Right: BB8 6DoF gluegun detection (green box denotes the 3D bounding box acquired from the HTC vive, purple box the 3D bounding box found by BB8 method.

In our experiments, we encountered high sensitivity to the quality of the depth data and its synchronization with the image data. This is the main problem in the case of the quick movements of the objects, i.e., for the tool manipulation tasks, since even a small distortion in the 6DoF pose estimation of the whole object will cause a significant position error of the tip of the tool. Especially for Hodan method we observed a significant improvement in detections when we used segmentation from Mask R-CNN to cut region of interest from the image. Mask R-CNN was also helpful for 2D shape detection for the BB8 method; the results for 6D pose estimation by BB8 (without refinement) were rather poor, however (especially in presence of occlusion, as can be seen in Fig. 5). On top of these, as known, BB8 method has
problems with symmetrical objects. In the stacking task, the detections were more reliable as the manipulated objects are simple geometric shapes.

Furthermore, it is worth noting that datasets on which are these 6DoF methods typically evaluated are very different from data which are important for imitation learning. Typical dataset includes only static objects, fixed distance of camera from the object, similar distance of camera from objects in training and testing dataset, ideal environment with ideal lightning conditions and well synchronized depth and RGB images.

G. Tracking human pose

For imitation of a human demonstrated activity, tracking human body is an obvious task – e.g., human joints in space. Estimation of human joints based on RGBD sensor is usually referenced as human pose estimation, skeleton tracking or skeletal pose. Current methods estimate human pose from depth images or monocular images or join depth and image data [4].

The state-of-the-art challenge is to estimate human pose while holding some a priori unknown objects or occlusions of body parts. In [24], the authors proposed a method for taking advantage of a context of human body parts and observed scene. Unfortunately, it still remains unsolved.

We observed ROS compatible open-source tools for estimation of human pose which is OpenNI and NiTE. Given an RGB-D image, we can process it in openniTracker package\(^1\) to obtain 15 transformations to the specified world for each human estimated joint. It is namely: head, neck, torso, left shoulder, left elbow, left hand, right shoulder, right elbow, right hand, left hip, left knee, left foot, right hip, right knee, right foot. These transformations are combined together as a transformation network which represents human pose in terms of joints for recording his/her actions.

Fig. 6. The failure of the human pose estimation algorithm to correctly estimate the pose of a sitting person that scratches their head while holding an object in the other hand. Note that the occlusion of legs behind the table causes a wrong estimate of the lower body parts. In the left arm the human has a tool which causes a wrong estimation of the wrist joint.

While holding an object in hands during manipulation tasks, we face huge variability of estimation, which leads to instability and unreliability of the estimation. Detection fails (see Fig. 6) mainly in occlusions and atypical positions (such as couching, sitting behind desk or putting hand into pocket). Many of the standard methods use pre-trained models trained on typical position of standing human in good light conditions. In any other position segmentation starts to fail and provide imprecise estimates of joint position and orientation. Another limitation is correct tracking of multiple humans, especially if overlap occurs or a person leaves the scene temporarily. However, there are recent attempts to detect human pose in wild. For example, Papandreou et al. [25] proposed a two stage method for multi-person detection and 2-D pose estimation in wild from RGB images.

OpenPose is an alternative algorithm [26] which uses only RGB information and estimates a maximum of 25 joints and 70 key point face points. Since this method uses only RGB information, only 2D information can be inferred and additional mapping to depth image is needed.

H. Language

Language commands can serve as a useful supplementary source of information for cases where the robot cannot extract enough data from vision – e.g., detecting time and pose of contact, exact position such as a corner, etc.. However, mapping between vision and language brings up a number of constraints which need to be considered.

Firstly, a method for language grounding needs to be adopted, so that the robot is able to transform the command into its contextual perception. Semantic parsing, i.e. mapping between a natural language (NL) sentence and its logical representation, can be obtained using a probabilistic Combinatorial Categorial Grammar (CCG) [27]. However, this approach relies on manually defined rules for such mapping - therefore, for our task it is more suitable to adopt a method where the relations are learned from data or human dialogues and can be applied for different environments. Such methods have been proposed e.g. in [28][29][30].

Secondly, it is important to obtain accurate temporal synchronization between performed actions and spoken commands. This must be held in mind by the person who is demonstrating the task and thus makes the framework less user-friendly. This issue can be minimized in final stages of the model development, e.g. by using the dependency relation matrix proposed in [31], enabling the model to extract correct command sequences from wrongly ordered inputs.

To implement language in our framework, we have been progressing in a bottom-up direction. The first goal is selecting a suitable automatic speech recognition (ASR) system. We have compared between the open-source version of Google Speech API and CMUSphinx Open Source Speech Recognition system. In CMUSphinx, we compared between a naive, untrained version and an adapted acoustic model with custom ARPA language model, specific to our gluegun task. Testing of all three systems (Google Speech API, default and adapted CMUSphinx) showed that for a predefined task with a limited vocabulary, it is most plausible to train a custom model with CMUSphinx (the measured WER for the trained model was below 1%). However, if we cannot specify the vocabulary beforehand, Google Speech API has a better general performance (16% WER compared to 76% WER for

\(^1\)http://wiki.ros.org/openni_tracker
untrained CMUSphinx). Therefore, our selected ASR system for the imitation learning task was Google Speech API.

IV. CONCLUSION

Enabling learning only by visual demonstration without any on-teacher sensors faces many challenges and opportunities. In this paper, we highlighted them and proposed the possible solutions. There are already new emerging algorithms which try to deal with many of these challenges (such as human pose detection in occlusions and atypical positions, etc.). However, we see the way to the cognitive robot, capable of understanding and imitating human activity without any external sensors and markers, still as a challenging task. Many of the current machine learning algorithms will have to be adopted and improved and novel approaches will have to be developed. In our future work, we want to use the presented findings and incorporate them to our architecture. We would like to quantitatively evaluate RGB and RGB-D methods (namely [16] and [23] methods) in our demonstration tasks.

ACKNOWLEDGMENT

This work was supported by the European Regional Development Fund under projects Robotics for Industry 4.0 (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000470), TACR Zeta project Imitation learning supported by language for industrial robots (no. TJ01000470) granted by Technological agency of Czech republic, and IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468).

REFERENCES

[1] G. Guerra-Filho, “Optical motion capture: Theory and implementation,” *Journal of Theoretical and Applied Informatics*, vol. 12, no. 2, pp. 61–89, 2005.
[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, “Robot programming by demonstration,” in *Springer handbook of robotics*. Springer, 2008, pp. 1371–1394.
[3] B. Argall, E. Sauser, and A. Billard, “Tactile guidance for policy adaptation,” *Foundations and Trends in Robotics*, vol. 1, pp. 79–133, 2010.
[4] C. Zimmermann, T. Welscheloh, C. Dornhege, W. Burgard, and T. Brox, “3d human pose estimation in rgbd images for robotic task learning,” *arXiv preprint arXiv:1803.02622*, 2018.
[5] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoisse-Julien, “Unsupervised learning from narrated instruction videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4575–4583.
[6] C. Liu, S. Yang, S. Saba-Sadiya, N. Shukla, Y. He, S.-C. Zhu, and J. Chai, “Jointly learning grounded task structures from language instruction and visual demonstration,” in *Proceedings of the 2016 IEEE Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1482–1492.
[7] M. Mühlig, M. Gienger, and J. J. Steil, “Interactive imitation learning of object movement skills,” *Autonomous Robots*, vol. 32, no. 2, pp. 97–114, 2012.
[8] S. Hussmann, T. Ringbeck, and B. Hagebeuker, “A performance review of 3d tof vision systems in comparison to stereo vision systems,” *in Stereo vision*. InTech, 2008.
[9] R. Horard, M. Hansard, G. Evangelidis, and C. Ménié, “An overview of depth cameras and range scanners based on time-of-flight technologies,” *Machine vision and applications*, vol. 27, no. 7, pp. 1005–1020, 2016.
[10] P. Zanutti, G. Marin, C. Dal Mutto, F. Domino, L. Minto, and G. M. Cortelazzo, “Operating principles of structured light depth cameras,” in *Time-of-Flight and Structured Light Depth Cameras*. Springer, 2016, pp. 43–79.

[11] M. Carfagni, R. Furferi, L. Governi, M. Servi, F. Uccheddu, and Y. Volpe, “On the performance of the intel s300 depth camera: metrological and critical characterization,” *IEEE Sensors Journal*, vol. 17, no. 14, pp. 4508–4519, 2017.
[12] M. Smid and J. Matas, “Rolling shutter camera synchronization with sub-millisecond accuracy,” in *Proc. 12th Int. Conf. Comput. Vis. Theory Appl.*, 2017, p. 8.
[13] C. Albi, Z. Kukelova, A. W. Fitzgibbon, J. Heller, M. Smid, and T. Pajdla, “On the two-view geometry of unsynchronized cameras,” in *CVPR*, 2017, pp. 5593–5602.
[14] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, “Mask R-CNN,” *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870.
[15] Z. Zhang and T. Funkhouser, “Deep depth completion of a single rgb-d image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 175–185.
[16] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, M. J., and C. Rother, “Bop: Benchmark for 6d object pose estimation,” in *Computer Vision (ECCV), 2018 IEEE European Conference on*, 2018.
[17] M. E. Munich, P. Pirjanian, E. Di Bernardo, L. Goncalves, N. Karlsson, and D. Lowe, “Sift-ing through features with vgs,” *IEEE robotics & automation magazine*, vol. 13, no. 3, pp. 72–77, 2006.
[18] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*. Springer, 2006, pp. 404–417.
[19] T. Hodan, X. Zabulis, M. Lourakis, Š. Obrdžálek, and J. Matas, “Detection and fine 3d pose estimation of texture-less objects in rgb-d images,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 4421–4428.
[20] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, “Learning 6d object pose estimation using 3d object coordinates,” in *European conference on computer vision*. Springer, 2014, pp. 536–551.
[21] B. Drost, M. Ulrich, N. Navab, and S. Ilic, “Model globally, match locally: Efficient and robust 3d object recognition,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 998–1005.
[22] M. Rad and V. Lepetit, “Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth,” in *International Conference on Computer Vision*, 2017.
[23] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes,” *arXiv preprint arXiv:1711.00199*, 2017.
[24] B. Yao and L. Fei-Fei, “Modeling mutual context of object and human pose in human-object interaction activities,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 17–24.
[25] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, “Towards accurate multi-person pose estimation in the wild,” in *CVPR*, vol. 3, no. 4, 2017, p. 6.
[26] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” in *CVPR*, 2017.
[27] P. Kwaikowski, L. Zettlemoyer, S. Goldwater, and M. Steedman, “Inducing probabilistic ccg grammars from logical form with higher-order unification,” in *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2010, pp. 1223–1233.
[28] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, “Learning to parse natural language commands to a robot control system,” in *Experimental Robotics*. Springer, 2013, pp. 403–415.
[29] J. Thomason, S. Zhang, R. J. Mooney, and P. Stone, “Learning to interpret natural language commands through human-robot dialog,” in *IJCAI*, 2015, pp. 1923–1929.
[30] M. Alomari, P. Duckworth, M. Hawasly, D. C. Hogg, and A. G. Cohn, “Natural language grounding and grammar induction for robotic manipulation commands,” in *Proceedings of the First Workshop on Language Grounding for Robotics*, 2017, pp. 35–43.
[31] Y. Cheng, J. Bao, Y. Jia, Z. Deng, Z. Sun, S. Bi, C. Li, and N. Xi, “Modeling robotic operations controlled by natural language,” *Control Theory and Technology*, vol. 15, no. 4, pp. 258–266, 2017.
What does my knowing your plans tell me?

Yulin Zhang  
Dept. of Comp. Sci. & Engr.  
Texas A&M University  
College Station, Texas, USA  
yulinzhang@tamu.edu

Dylan A. Shell  
Dept. of Comp. Sci. & Engr.  
Texas A&M University  
College Station, Texas, USA  
dshell@tamu.edu

Jason M. O’Kane  
Dept. of Comp. Sci. & Engr.  
University of South Carolina  
Columbia, South Carolina, USA  
jokane@cse.sc.edu

Abstract—For robots acting in the presence of observers, we examine the information that is divulged if the observer is party to the robot’s plan. Privacy constraints are specified as the stipulations on what can be inferred during plan execution. We imagine a case in which the robot’s plan is divulged beforehand, so that the observer can use this a priori information along with the disclosed executions. The divulged plan, which can be represented by a procrastinate graph, is shown to undermine privacy precisely to the extent that it can eliminate action-observation sequences that will never appear in the plan. Future work will consider how the divulged plan might be sought as the output of a planning procedure.

I. INTRODUCTION

Autonomous robots are beginning to be part of our everyday lives. Robots may need to collect information to function properly, but this information can be sensitive if leaked. In the future, robots will not only need to ensure physical safety for humans in shared workspaces, but also to guarantee their information security. But information leakage can occur in a variety of ways, including through logged data, robot’s status display, actions, or, as we examine, through provision of prior information about a robot’s plan.

Established algorithmic approaches for the design and implementation of planners may succeed at selecting actions to accomplish goals, but they fail to consider what information is divulged along the way. While several models for privacy exist, they have tended to be either abstract definitions applicable to data rather than an agent operating autonomously in the world (such as encryption [1], data synthesis [2], anonymization [3], or opacity [4] mechanisms) or are focussed on a particular robotic scenario (such as robot division of labor [5] or tracking [6, 7]).

Figure 1 illustrates a scenario where the information divulged is subtle and important. It considers an autonomous wheelchair that helps a patient who has difficulty navigating by himself. The user controls the wheelchair by giving voice commands: once the user states a destination, the wheelchair navigates there autonomously. While moving through the house, the wheelchair should avoid entering any occupied bedrooms, making use of information from motion sensors installed inside each bedroom. We are interested in stipulating the information divulged during the plan execution:

Positive disclosure of information: A therapist monitors the user, ensuring that he adheres to his daily regimen of activity, including getting some fresh air everyday (by visiting the front yard or back yard).

Negative disclosure of information: However, if there is a guest in one of the bedrooms, the user does not want to disclose the guest’s location.

Actions, observations, and other information (such as the robot’s planned motion) may need to be divulged to satisfy the first (positive) stipulation. The challenge is to satisfy both stipulations simultaneously. Suppose the robot executes the plan shown in the right of Fig. 1, and that this plan is public knowledge. If, as it moves about, the robot’s observations (or actions) are disclosed to an observer, then we know that the robot will attempt to see if M is occupied. Hence, on some executions, a third party, knowing there is a guest, would be able to infer that they’re in the master bedroom.

This paper examines in detail how divulging the plan, as above, provides information that permits one to draw inferences. In particular, we are interested in how this plan information might cause privacy violations. As we will see, the divulged plan need not be the same as the plan being executed, but they must agree in a certain way. In our future work, we hope to answer the question of how to find pairs of plans (one be to executed and one to divulged), where there is some gap between the two, so that information stipulations are always satisfied.

![Fig. 1: An autonomous wheelchair navigates in a home. A plan, on the right, generates actions that depend on perception of the pink star (denoting that the bedroom is occupied).](image)

II. PROBLEM DESCRIPTION

In this problem, there are three entities: a world, a robot, and an observer. As shown in Fig. 2, the robot interacts with the world by taking observations from the world as input, and
outputting an action to influence the world state. This interaction generates a stream of actions and observations, which may be perceived by the observer, though potentially only in partial or diminished form. We model the stream as passing through a function which, via conflation, turns the stream generated by the world–robot interaction into one perceived by the observer, the disclosed action-observation stream. As a consequence of real-world imperfections (possible omission, corruption, or degradation) or due to explicit design, the observer, thus, may receive less information. For this reason, the function is viewed as a sort of barrier, and we term it an information disclosure policy.

The observer is assumed to be unable to take actions to interact with the world directly—a model that is plausible if the observer is remote, say a person or service on the other side of a camera or other Internet of Things device. Given its perception of the interaction, the observer estimates the plausible action-observation streams, consistent with the disclosed action-observation stream. This estimate can be made ‘tighter’ by leveraging prior knowledge about the robot’s plan. The observer’s estimate is in terms of world states, so the notion of tightness is just a subset relation. In this paper, we will introduce stipulations on these estimated world states and our main contribution will be in examining how the divulged plan could affect the satisfaction of these stipulations.

A. Representation

To formalize such problem, we represent these elements with p-graph formalism and label map [8]. The world is formalized as a planning problem \((W, V_{goals})\), where \(W\) is a p-graph in state-determined form (see definition of state-determined in [8, Def. 3.7]) and \(V_{goals}\) is the set of goal states. The robot is modeled as a plan \((P, V_{term})\), where \(P\) is a p-graph and \(V_{term}\) specifies the set of plan states where the plan could terminate. The plan solves the planning problem when the plan can always safely terminate at the goal region in finite number of steps (see definition of solves in [8, Def. 6.3]). The information disclosure policy is represented by a label map \(\mathcal{h}\), which maps from the actions and observations from \(W\) and \(D\) to an image space \(X\). The observer is modeled as a tuple \((I, D)\), where \(I\) is a filter represented by a p-graph with edge labels from \(X\), \(D\) is the p-graph representing the divulged plan with actions and observations labeled in the domain of \(\mathcal{h}\). The plan in \(D\) might be less specific than the actual plan \(P\), representing ‘diluted’ knowledge of the plan; to capture this, we require that all possible action-observation sequences (called executions for short) in \(D\) should be a superset of those in \(P\), denoted as \(\mathcal{L}(D) \supseteq \mathcal{L}(P)\) (the set of executions is called the language, see [8, Def. 3.5], hence the symbol \(\mathcal{L}(\cdot)\)).

B. The observer’s estimation of world states

Given any set of filter states \(B\) from filter \(I\), the observer obtains an estimate of the executions that should’ve occurred to reach \(B\), through a combination of the following sources of information [9, Def. 13]:

1) The observer can ask: What are all the possible executions, each of which has its image, reaching exactly \(B\) in the filter? The set of executions reaching exactly \(B\) is represented as \(\mathcal{S}_B^I\). The preimages of \(\mathcal{S}_B^I\), which we denote as \(h^{-1}[\mathcal{S}_B^I]\), are the executions which are responsible for arriving at \(B\) in \(I\).

2) The observer can narrow down the estimated executions to the ones that only appear in the divulged plan \(D\). The set of all executions in \(D\) are represented by its language \(\mathcal{L}(D)\).

3) Finally, the estimated executions can be further refined by considering those that appear in the world, i.e., \(\mathcal{L}(W)\).

Hence, \(h^{-1}[\mathcal{S}_B^I] \cap \mathcal{L}(W) \cap \mathcal{L}(D)\) represents a tight estimation of the executions that may happen. This allows us to find the estimated world states, defined as \(\mathcal{W}_B^D\), by making a tensor product \(T\) of graph \(W\), \(D\) and \(h^{-1}(I)\), where \(h^{-1}(I)\) is obtained by replacing each action or observation \(\ell\) with its preimage \(h^{-1}(\ell)\) on the edges of the p-graph \(I\). For any vertex \((w, d, i)\) from the product graph \(T\), we have:

\[
\mathcal{W}_B^D = \mathcal{W}_B^D \cup \{w\}; \text{ if } i \in B.
\]
C. Information stipulations on the estimated world states

Information stipulations are written as propositional formulas on estimated world states $W_B^D$. Firstly, we will define a symbol $w$ for each world state $w$ in $W$. Then we can use connectives $\neg, \wedge, \vee$ to form composite expressions $\Phi$ that stipulate the estimated world states involving these symbols. The propositional formulas can be evaluated based on the following definition:

$$w = \text{True} \quad \text{if and only if} \quad w \in W_B^D.$$  

With all the elements defined above, we are able to check whether the stipulation $\Phi$ is satisfied on every estimate $W_B^D$, given the world graph $W$, information disclosure policy $h$, and the observer $(I, D)$.

III. The observer’s prior knowledge of the robot’s plan

The divulged plan $D$ is known by the observer prior to the robot’s monitoring of the disclosed action-observation stream. Depending on how much the observer knows, there are four possibilities, from most-to least-informed:

I) The observer knows the exact plan $P$ to be executed.
II) The plan to be executed can be hidden among a (non-empty) finite set of plans $\{P_1, P_2, \ldots, P_n\}$.
III) The observer may only know that the robot is executing some plan, that is, the robot is goal directed and aims to achieve some state in $V_{\text{goal}}$.
IV) The observer knows nothing about the robot’s execution other than that it is on $W$.

A p-graph exists whose language expresses knowledge for each of these cases:

Case I. When $D = P$, the interpretation is straightforward: the observer tracks the states of the plan given the stream of observations (as best as possible, as the operation is under $h$).

Case II. If instead a set of plans $\{P_1, P_2, \ldots, P_n\}$ is given, we must construct a single p-graph, $D$, so that $L(D) = L(P_1) \cup \cdots \cup L(P_n)$. This is achieved via the union of p-graphs $D = P_1 \uplus P_2 \uplus \cdots \uplus P_n$, cf. [8, Def. 3.6, pg. 18].

Case III. If the robot is known only to be executing some plan, we must consider the set of all plans, $P^\infty := \{P_1, P_2, P_3, \ldots\}$. As the notation hints, there can be an infinite number of such plans, so the approach of unioning plans won’t work. Fortunately, another structure, $P^*$, exists such that $L(D) = L(P^*) = L(P^\infty)$, which will be proved afterwards. Here $P^*$, a finite p-graph, is called the plan closure.

Case IV. When taking $D = W$ the executions are, again, intersected with $L(D)$ but as they already came from $L(W)$, this shows why the observer is the least informed in the hierarchy.

Next, we will show the construction of the plan closure $P^*$ and prove that $L(P^*) = L(P^\infty)$.

To start, we describe construction of $P^*$. The initial step is to convert $W$ to its state-determined form $W' = SDe(W)$ (this is an operation described in [8, Algorithm 2, pg. 30]). Then, to decide whether a vertex in $W'$ exists in some plan, we iteratively color each vertex green, red, or gray. Being colored green means that the vertex exists in some plan, red means that the vertex does not exist in any plan, and gray indicates that its status has yet to be decided. To start with, we initially color the goal vertices green, and non-goal leaf vertices (with no edges to other vertices) red. Using the iconography of [8], we show action vertices as squares and observation vertices as circles. Then gray vertices of each type change their color by iterating the following steps:

- $\square \rightarrow \square \exists$ some action $a$ reaching $\bullet$, which is not an initial state.
- $\square \rightarrow \square \forall$ action $a$ reaching $\bullet$.
- $\bullet \rightarrow \bullet \forall$ observation $o$ reaching $\square$, which is not an initial state.
- $\bullet \rightarrow \bullet \exists$ some observation $o$ reaching $\square$.

The iteration ends when no vertex changes its color. The subgraph that consists of only green vertices and their corresponding edges is $P^*$. And $P^*$ then contains only the vertices that exist in some plan leading to the goal states. For further detail of this algorithm for building $P^*$, we refer the reader to Algorithm 1.

Next, we prove that the $P^*$ constructed from this procedure has the same language as $P^\infty$. The proof shows that any green vertex is on some plan, by showing that we can construct a plan $\pi$, that will lead to a goal state within a finite number of steps form any such vertex.

Lemma 1. $L(P^*) = L(P^\infty)$.

Proof. $\supseteq$: For any $s = s_0s_1s_2\ldots s_k \in L(P^\infty)$, according to the definition of $P^\infty$, $s$ is in the execution of some plan $P^\ast$. Though $s_k$ may not be a goal, using $P^\ast$, $s$ can be extended: $\exists s' = s_0s_1\ldots s_kt_0t_1\ldots t_n \in L(P^\ast), k > 0, n \geq 0$ to reach an element of $V_{\text{goal}}$. Then $V_{\text{goal}}'$ comprises vertices associated with those in $W'$ marked green in $V_{\text{goal}}'$. And, tracing the execution $s'$ on $P'$ backwards on $W'$, we find every vertex green back to a start vertex. But this means they are in $P^\ast$, and hence $s' \in L(P^\ast)$, means $s \in L(P^\ast)$ as well.

$\subseteq$: For any execution $s = s_0s_1s_2\ldots s_k \in L(P^\ast)$, $s$ reaches $V_{\text{goal}}'$, or $s$ is a prefix of some execution reaching $V_{\text{goal}}'$ in $W'$. We show that there is a plan that can produce $s$. The execution $s$ does not include enough information to describe a plan because: (1) it may not reach $V_{\text{goal}}'$ itself, and (2) it gives an action after some observation that was revealed, but not every possible observation. To address this shortfall, we will capture some additional information during the construction of $P^\ast$, which we save in $\pi$. This provides an action that makes some progress, for states that can result from other observations. Now, using $s$ as a skeleton, construct plan where once a
transition outside of $s$ occurs, either owing to an unaccounted-for observation or having reached the end of $s$, the plan reverts to using the actions that $\pi$ prescribes. (See Fig. 3 for a visual example.) This is always possible because states arrived at in $W'$ under $s$ are green. This implies that all states in $W$ are also assured to reach a goal states. The resulting plan can produce $s$, so some plan produces $s$, hence $s \in \mathcal{L}(P^\infty)$.

Thus, one may use $D = P^*$, for Case III.

IV. EXPERIMENTAL RESULTS

We implemented the algorithms with Python, and execute them on a OSX laptop with a 2.4 GHz Intel Core i5 processor. To experiment, we constructed a p-graph representing the world in Fig. 1 with 12 states, and the plan with 8 states. All the experiments are finished within 1 second. The information disclosure policy maps all actions to the same image, but observations to different images. As we anticipated, the stipulations are violated when the exact plan is divulged. But we can satisfy the stipulations by disclosing less information, such as $D = W$.

V. SUMMARY AND FUTURE WORK

We examine the planning problem and the information divulged within the framework of procrustean graphs. In particular, the divulged plan can be treated uniformly in this way, despite representing four distinct cases. The model was evaluated, showing that divulged plan information can prove to be a critical element in protecting the privacy of an individual. In the future, we aim to automate the search for plans: given $P$ to be executed, find a $D$ to be divulged, where $\mathcal{L}(D) \supseteq \mathcal{L}(P)$, such that the privacy stipulations are always satisfied.

ACKNOWLEDGEMENTS

This work was supported by the NSF through awards IIS-1453652, IIS-1527436, and IIS-1526862. We thank the anonymous reviewers for their time and valuable comments.

REFERENCES

[1] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, Handbook of Applied Cryptography. CRC Press, Inc., 1996.
[2] D. B. Rubin, “Discussion of Statistical Disclosure Limitation,” Journal of Official Statistics, vol. 9, no. 2, pp. 461–468, 1993.
[3] C. Dwork, “Differential privacy: A survey of results,” in Proceedings of International Conference on Theory and Applications of Models of Computation. Springer, 2008, pp. 1–19.
[4] R. Jacob, J.-J. Lesage, and J.-M. Faure, “Overview of discrete event systems opacity: Models, validation, and quantification,” Annual Reviews in Control, vol. 41, pp. 135–146, 2016.
[5] A. Prorok and V. Kumar, “A macroscopic privacy model for heterogeneous robot swarms,” in Proceedings International Conference on Swarm Intelligence. Springer, 2016, pp. 15–27.
[6] J. M. O’Kane, “On the value of ignorance: Balancing tracking and privacy using a two-bit sensor,” in Proceedings of International Workshop on the Algorithmic Foundations of Robotics, 2008, pp. 235–249.
[7] Y. Zhang and D. A. Shell, “Complete characterization of a class of privacy-preserving tracking problems,” International Journal of Robotics Research—in WAFR’16 Special Issue, 2018.
[8] F. Z. Saberifar, S. Ghasemlou, D. A. Shell, and J. M. O’Kane, “Toward a language-theoretic foundation for planning and filtering,” International Journal of Robotics Research—in WAFR’16 Special Issue, 2018.
[9] Y. Zhang, D. A. Shell, and J. M. O’Kane, “Finding plans subject to stipulations on what information they divulge,” in Proceedings of International Workshop on the Algorithmic Foundations of Robotics, 2018.

Algorithm 1: $P^*$CONSTRUCTION($W, V_{goal}$)

\begin{algorithm}
\begin{algorithmic}
\State Initialize queues red, green, gray as empty
\State $W' \leftarrow \text{SDE}(W)$, and initialize $V'_{goal}$ as the associated vertices of $V_{goal}$
\State Initialize plan $\pi$ as empty
\For{$v \in V(W')$}
\If{$v \in V'_{goal}$}
\State green.append($v$)
\ElseIf{$v$ has no edges to other vertices}
\State red.append($v$)
\Else
\State gray.append($v$)
\EndIf
\EndFor
\While{$Q$ not empty}
\State $v \leftarrow Q$.pop
\State flag $\leftarrow$ True
\If{$v$ is a $\otimes$ then}
\If{one of its outgoing neighbors is $\square$}$\text{then}$
\State red.append($v$)
\ElseIf{all of its outgoing neighbors are $\square$}$\text{then}$
\State green.append($v$)
\Else
\State flag $\leftarrow$ False
\EndIf
\ElseIf{$v$ is a $\square$ then}
\If{one of its outgoing neighbors under label $\circ$ is $\bullet$}
\State green.append($v$) and $\pi[v] = a$
\ElseIf{all of its outgoing neighbors are $\circ$}$\text{then}$
\State red.append($v$)
\Else
\State flag $\leftarrow$ False
\EndIf
\EndIf
\EndFor
\State $Q$.extend($\text{InNeighbor}(\text{red} \cup \text{green}) \setminus (\text{red} \cup \text{green})$)
\EndWhile
\State $P^* \leftarrow \text{subgraph}(W', \text{green})$
\State return $P^*$ (and also $\pi$, if desired)
\end{algorithmic}
\end{algorithm}
Empathy Display Influence on Human-Robot Interactions: a Pilot Study

Laurianne Charrier, Alexandre Galdeano, Amélie Cordier, and Mathieu Lefort

Abstract—Social robots are designed to interact and communicate with humans. We have conducted a pilot study to explore how an artificial empathy module can affect Human-Robot Interactions. For that pilot study, we chose to evaluate the effects of a module we developed called “attention-based empathic module” and we set up an experiment within two conditions, “Empathy” (i.e., with this module), and “No-Empathy” (i.e., without this module). In order to define what aspects of HRI are affected, we used several metrics found in HRI literature including self-reported questionnaires—e.g., perceived empathy test—physiological measures—e.g., number of attentional disengagements—and objective measures—e.g., time of interaction, and performance. Dividing 36 participants into two groups and controlling the main biases inherent in subjects selection, we found that the “attention-based empathic module” seems to have affected 9 metrics: the interaction duration, how trustworthy the robot was perceived, the number of disengagements, how empathic the robot was perceived, how much participants felt they knew the robot, how the robot’s intelligence was perceived, how comfortable the interaction was perceived, how much the robot was perceived as knowledgeable, and how engaging the interaction was perceived. Due to the exploratory approach of this study, these results have to be confirmed.

Keywords—Human-Robot Interaction, Social Robots, Empathy, Empathic Display, Interaction Involvement, Attention.

I. INTRODUCTION

With small and affordable robots like Cozmo or Jibo, social robots arrive on the mass market, democratizing technologies that were up to now reserved to research institutions and companies. A social robot must be able to perceive its environment through senses [14], act on it following social norms [3], and more generally be able to perform social interactions in a Human-like way [8]. One way to improve social interactions with robots and which is greatly explored in HRI literature including self-reported questionnaires—e.g., knowledge, beliefs, intentions, thoughts, emotions, desires—to themselves and to others [7]. [34] in order to predict, adapt to, and explain their behaviors [27]. It’s important to notice that, with these definitions, being unemphatic is to be unable to understand and predict the mental states of others whereas doing incongruent empathy is about misunderstanding others’ mental states and displaying an inappropriate use of empathy. As in several studies—e.g., [23], [24], [13], [20]—we chose to focus our study on cognitive empathy only, more precisely on the robot’s relevance in adapting its behaviors to the user’s interaction involvement. Interaction involvement can be considered as a mental state and its understanding is part of cognitive empathy.

According to literature definitions, interaction involvement can be conceptualized as how much an individual partakes in a social environment [9] and, more precisely, to the extent participants are immersed and engaged in an ongoing social communication [10], [12]. It is a characteristic way of processing information and respond to messages during face-to-face communication that can be decomposed into three factors [17]:

Responsiveness: the need to respond to the situation in order to lead to a meaningful conversation.

Perceptiveness: an individual’s ability to assign meaning to others’ behaviors and interpret the meanings others assign to one’s own behaviors.

Attentiveness: one’s degree of cognitive involvement in an interaction, as attentional commitment.

In face-to-face settings, someone being involved in the interaction is attentive to the other and responsive to the evolving circumstances of conversation. In this pilot study, we used a robotic module called “attention-based empathic module” that measures the user’s attentiveness using a custom deep-learning model which has images from the robot camera as inputs. When the module detects a loss of attention, the robot reacts in order to get the user’s attention back, i.e., it triggers an animation, and it makes the robot say sentences such as “You must be attentive”.

The goal of this pilot study is to identify which measures are the most relevant when studying the influence of the empathic display on the interaction. This will help to give insights on how to setup further experiments, and on which tools and
measures to use, rather than drawing solid conclusions about the algorithm’s effects on the interaction.  

We first detail our methodology in Section II including the material used (Section II-A), the participants’ selection and evaluation (Section II-B), the experiment procedure (Section II-C), the measures used (Section II-D), and how we analyzed the data (Section II-E). In Section III, we present our results regarding the self-reported (Section III-A), objective (Section III-B), and physiological measures (Section III-C). We then discuss our results in Section IV.

II. METHODOLOGY

A. Material

We chose to use Pepper—a 1.2 meter high humanoid robot made by SoftBank Robotics. This mass-market robot has the ability to communicate with Humans through speech, gestures, and its tablet.

A quiz in English has been implemented with 32 questions in various knowledge fields, 4 answers per question are given (Fig 1), and each question is followed by an anecdote. It is very much like the “Who Wants to Be a Millionaire?” TV show. The quiz aims at entertaining the user as long as possible, but the user is asked every four questions to continue or not. The first condition is the quiz as-is—the No-Empathy condition—and the second one is the quiz with the attention-based empathic module—the Empathy condition. In the two conditions, Pepper was animated to make the interaction more natural, with a face tracking and gestures when speaking.

In the Empathy condition, when the user is not attentive to what the robot is saying, Pepper makes large and exaggerated gestures and calls back for the attention of the user using sentences like “You must be attentive” as could be done by a professor in front of an inattentive student. This simple behavior makes it easier to understand the causes behind its effects on the user’s perception of both the robot and the interaction. Pepper was placed in a calm and closed room with good lighting conditions, and always in the same spot to avoid biases. A NAO robot was put on a cabinet next to Pepper to record the experimentation without participants knowing they were filmed, and all its lights were turned off for this purpose. GStreamer was used to get the video stream from the NAO’s camera and VLC player was used to record it.

B. Participants

All the questionnaires were in English and we wanted to avoid biases with translations So we chose to also do the quiz in English for coherence. We also evaluated the English understanding level of each participant, to keep the questionnaires and the quiz’s questions in English to avoid biases with translations, we evaluated the English understanding level of each participant, i.e., only people that could at least make simple sentences and understand the main points of a conversation in English were included in the experiment. Then, robot experience, acceptability, and personality were tested to ensure that each group was balanced as much as possible in term of bias sources: each bias could indeed lead to a different way to interact. Participants proceeded to the mini IPIP test [16] to measure five sides of their personality, including their extroversion, and a homemade questionnaire inspired by the Eurobarometer 382 [42] to measure their robot experience. To complete these questionnaires, acceptability was measured with the Negative Attitudes towards Robots Scale (NARS) [40] and the Robot Anxiety Scale (RAS) [26]. Personal empathy level was tested with the Interpersonal Reactivity Index (IRI) [15].

The 36 participants were subdivided randomly into two groups to perform one of the two experimental conditions. Each group was composed of 18 persons—9 males and 9 females—with most of them being in the 18–27 years-old range. In addition to these participants, 8 more participants in the Empathy condition initially achieved the experiment but did not trigger the interaction involvement detector and consequently did not experience the robot’s empathic behaviors. This is due to two factors: 1) the attention-based empathic module did not detect any loss of attention, or; 2) the participant was attentive during the whole experiment. These subjects’ results were removed from the study because we wanted to only measure the potential effects of the empathic displays on how the interaction and the robot are perceived.

C. Procedure

First, participants responded to an online questionnaire about demography, personality, empathy, robot experience
and acceptability. The results from these questionnaire were used to make the two groups while limiting biases as much as possible. They were told that the experiment was about studying the effects of personality on how the social robots is used to limit response bias. They then came to test our application at Hoomano’s office for a session of about 30 minutes. After giving their authorization to use video and signing consent forms, the experiment began without them knowing it. Participants were told that they had to train first with the quiz on Pepper, during as much time as they wanted, and that they will then perform the experiment. We did that to limit experimental bias with people knowing they were looked at and not acting naturally [18], [29]. After being sure that the subject well understood the instructions, the experimenter left the subject alone with the robot, launching the data gathering when coming out of the room. After the participants came back from the quiz task, we told them that they actually performed the experiment and that they had to fulfill a post-experiment questionnaire. This questionnaire was used to evaluate how the participants felt about the interaction and the robot.

D. Measures

We wanted to test different kinds of measures so we decided to mix self-reported, physiological, and objective measures. In total, we chose to test eight metrics to evaluate the effect of the attention-based empathic module, each of these have already been used in HRI studies and, more generally, in social studies. The self-reported metrics were:

- The Godspeed test [4]: a robot acceptance questionnaire.
- Bickmore’s test items [6]: a robot and an interaction evaluation questionnaire.
- The Barrett-Lennard Reactivity Index (BLRI) [2]: an empathy questionnaire used to evaluate the robot’s empathy.

As a physiological measure, we used the number of disengagements of the subject calculated with the video. At last, we tested four objective measures of the interaction: the distance between the robot and the subject, the number of questions answered, the number of good answers given, and the duration of the interaction. We evaluated the distance using the front sonar of the robot, while we recovered the duration of usage and number of disengagements on the robot after every play. For the distance, we removed the last 15 percents of the data to avoid noise deriving from the experiment’s end. We then calculated the mean distance and its standard deviation.

E. Statistical analysis

Since we are using an exploratory experimental method, we decided to set the significance level $\alpha$ to 20% for the metrics analysis, i.e., we considered a metric as being potentially affected by the attention-based empathic module if the difference in this metric’s results between the groups leads to a $p$-value below $\alpha$. We did that to select measures of interest for future experiments, rather than drawing solid conclusions. To assess the difference in a measure between the two groups, we used a T-test [39] if and only if the data from both groups were close enough to a normal distribution—i.e., the Shapiro-Wilk tests [32] were not significant—and they had fairly equal variances—i.e., the F-test of equality of variances was not significant either. If a measure did not follow these conditions, we used a Mann-Whitney test [25]. We measured correlations using Spearman’s rank correlation coefficient [36]. All statistical analysis were processed using the R software\(^3\).

III. RESULTS

A. Self-reported measures

We subdivided the self-reported measures into three sections: perceived empathy, robot acceptance, and interaction evaluation. Due to the important number of tested items, we just reported significant items here. A list of all non-significant items is available in Table I.

For the interaction evaluation, there were only two significant differences between Empathy and No-Empathy groups. First, a significant difference has been found between the two groups in term of empathy ($U = 213.5, p = .106$) in the BLRI (see Fig 2).

Indeed, the robot has been perceived as more empathic in the Empathy condition with a mean of ($M = -0.89, SD = 6.88$) whereas the No-Empathy condition had a mean of ($M = -3.56, SD = 6.44$).

For the Godspeed test [4], the perceived intelligence in the No-Empathy condition and the Empathy condition may be considered as normally distributed—($S = .965, p = .707$)

\(^3\)See https://www.r-project.org/
and \((S = .945, p = .349)\) respectively—and having equal variances \((F = 1.638, p = .319)\). This was the only scale which was significantly different between the Empathy and the No-Empathy groups \((t(35) = -1.528, p = .136)\) (see Fig 3). Pepper with the empathy module has been considered less intelligent than without empathy with respective means of \((M = 17.00, SD = 3.65)\) and \((M = 18.67, SD = 2.85)\), and less knowledgeable with respective means of \((M = 7.11, SD = 2.08)\) and \((M = 8.11, SD = 1.53)\). These two scales have a moderate positive correlation: \((r_S = .52, p = .027)\) for the Empathy condition, and \((r_S = .731, p = .001)\) for the No-Empathy condition. It is interesting to note that this difference in perceived intelligence did not affect likability nor anthropomorphism results.

Bickmore’s test items [6] were analyzed separately. Some items reported interaction evaluation and others robot evaluation. First, the comfortability \((U = 116.5, p = .142)\), with an interaction perceived as less comfortable with a mean of \((M = 6.67, SD = 1.91)\) in the Empathy condition whereas in the No-Empathy condition it had a mean of \((M = 7.50, SD = 1.20)\) (see Fig 4). Second, its engaging aspect \((U = 203.5, p = .188)\), with an interaction perceived as more engaging with the empathy algorithm \((M = 7.22, SD = 1.56)\) than without \((M = 6.28, SD = 2.32)\) (see figure 5). In term of how the robot was perceived, the participants significantly felt that they knew better the robot \((U = 212, p = .113)\) in the Empathy condition \((M = 6.39, SD = 2.09)\) than in the No-Empathy condition \((M = 5.39, SD = 2.03)\). They also significantly perceived \((U = 103.5, p = .059)\) the robot as less untrustworthy in the Empathy condition \((M = 3.17, SD = 2.33)\) than in the No-Empathy condition \((M = 3.61, SD = 1.20)\). Finally, there is a significant difference \((U = 212, p = .113)\) between the two groups in term of perceived knowledge of the robot. The robot has been perceived as less knowledgeable with the empathy module \((M = 7.11, SD = 2.08)\) than without \((M = 8.11, SD = 1.53)\).
IV. Discussion

We observed that the “attention-based empathic module” seemed to have affected 9 metrics—7 if we remove the two correlations:

1) the interaction duration;
2) how trustworthy the robot was perceived;
3) the number of disengagements (positively correlated to the interaction duration);
4) how empathic the robot was perceived;
5) how much participants felt they knew the robot;
6) how the robot’s intelligence was perceived;
7) how comfortable the interaction was perceived;
8) how much the robot was perceived as knowledgeable (positively correlated to the perceived intelligence); and
9) how engaging the interaction was perceived.

We observed that even if there was a slight increase in perceived empathy, we also observed a reduction of interaction quality: users perceived the interaction as less comfortable with the empathy module. This might be a sign of an inappropriate use of empathy. Indeed, previous results showed that a robot is perceived as less safe and credible than a neutral robot when displaying incongruent empathy [13]. And since an empathic display not adapted to the situation may lead to a worse interaction perception, there is a need to think about the way our robots can show effectively that they understand Human behaviors in order to avoid increasing frustration facing actual robots. However, [22] has shown that empathy can improve a lot the Human-Computer interactions even in case of dysfunction of the robot.

Another line of thought is that maybe we cannot just apply Human-Human empathy to a social robot because we may not accept the same behavior coming from a Machine than from another person. Moreover, [5] has shown that interacting at a personal distance, with 46cm to 122cm between Human and robot, only small and medium gestures were appropriate. And in our study the users stood at an average distance of 50cm, this could explain why the users may have wrongly interpreted the exaggerated movements that Pepper did to get back their attention. So the empathy display might have to be more subtle to do not disturb the user. One way to achieve that would be for the robot to mimic the user’s posture and gestures. Many studies demonstrated the relationship between mimicry and liking in Human-Human interaction. Indeed, [28] demonstrated a better satisfaction with the interaction if a robot mimics the upper body gestures of the user than if it does not. [37] shown an increased empathy towards the mimicked while being intentionally or spontaneously mimicked. [38] also compared liking after mimicking an a priori disliked or liked person. They concluded among others that when a person mimics a disliked person, liking for them was not improved. However, mimicking a liked person improved liking. If there is no information about the liking or disliking toward the person, mimicry also improves the liking for the mimicker [11]. This conclusion shows the importance for a robot to be at least positively seen to lead to a beneficial effect of mimicking. Mimicking could allow better emotional contagion [38] and make robots more lively. Other forms of empathy displays could be interesting to use like face emotion displays [19], use of light effects on the robot [35], and semantic adaptations to the context [24].

On the other hand, [43] manipulated the description of a NAO robot, giving it more of less cognitive abilities. The robot was always the same and always acted the same way, but it was perceived differently due to the initial description. The more the robot is said to have greater cognitive abilities, the less it is perceived as smart, trustful or as having true emotions. Participants also felt less understood.

When the user thinks that the robot has higher cognitive behaviors, this leads to a disappointment because the user expects the robot to act like a Human.
It might explain why our robot is perceived as less intelligent and less knowledgeable because, noticing that the robot is interpreting their behaviors, subjects may have attributed more cognitive abilities to the robot and, doing so, had more expectations. According to [8], user’s expectations are one of the main issues in HRI. They “can mitigate the person’s disappointment or frustration when interacting with the robot”, and they “can also gently steer the person to interact with the robot in the way it was intended”. We have to think about the effect of empathy displays on users expectations to avoid depreciation of interaction quality due to higher expectations. However, the “Intelligent” item from Bickmore’s test did not show a significant difference between the two groups, so maybe the difference in Godspeed’s item was a false positive.

The fact that the robot has been perceived as more trustworthy in the Empathy condition can be explained by giving trust to Pepper knowing it better as mechanical and as not willing to try to manipulate their mind because of a lack of cleverness. Attention has to be paid to the features that create expectations and the way the user sees the robot, as a tool or as a sociable partner distinguished on the base of the mental model a Human has of the robot when interacting with it [8].

An interesting phenomenon is the interaction duration that is on average about 3 minutes and a half longer in the Empathy group—about +32%—while responding to the same number of questions. It can partly be explained by the time taken by the robot to ask for attention. Moreover, in the Empathy condition, the robot asked back the attention of the users when they did not focus anymore on the game, so the users may be more focused and could spend more time to think about the question because of the feeling to be called to order. Having the robot asking for attention would also explain why the interaction felt more engaging for them.

In addition to the low number of subjects, this experiment suffers from limitations such as metrics fetching that can be improved. Videos were bad in recording quality leading to a slight loss of interaction details that could be interpreted. A better quality could help to evaluate more precisely our metrics. Another limitation is the use of English language based questionnaires and quiz questions on a French-speaking population. We chose to keep all the questionnaires in English, but it has been proved that word perception and meaning, such as emotion meaning, could vary with culture and the language in which the word is displayed [30], [21]. Misunderstanding of the meaning of some English words in the questionnaire could lead to biases.

V. CONCLUSION AND FUTURE WORK

In this 36-person pilot study, we explored how the HRI may be affected by the robot displaying its empathic understanding, and we selected nine measures that seem to be good candidates for such studies. In the Empathy condition, the participants perceived the robot as more empathic and the interaction as more engaging, the participants also they felt they knew the robot better and interacted with for about 3.5 minutes more. Furthermore, in the No-Emmpathy condition, the robot was perceived as more intelligent and more knowledgeable, and the interaction felt more comfortable. Moreover, participants made three kinds of comments about the study after completion:

a) The way Pepper gets back the attention is exaggerated: A need to carefully analyze Human behavior to adapt empathy displays from the robot is one of the main conclusions of this study.

b) The questionnaires are too long: Because the pre-questionnaires are used to evaluate the demography of our sample and to limit biases while assigning participants to groups, they cannot really be changed. However, with this pilot study, we found that some of the measures we used—self-reported, objective, or physiological—are either not important or—due to correlations—redundant in our case. They will be removed in future studies.

c) The Barrett-Lennard Reactivity Index is not adapted: While this measure was influenced as expected in the Empathy condition, some questions—such as “Pepper doesn’t avoid or go round anything that matters between us”—seemed weird to the participants. This kind of questions should probably be removed because changing them may affect the final score in ways we do not expect.

We removed 8 participants for whom the experiment did not go as planned: they did not experience the empathy display. This means that they had the same experience as if they were in the No-Emmpathy condition. So we also analyzed the data by considering them as being in the No-Emmpathy condition, and the measures that were significant before still are after this change, with—for most of them—a lower p-value. Moreover, new measures appeared below the 20% threshold: “How do you characterize your relationship with Pepper?” in favor of the Empathy condition, and “Perceived safety” and “Respectful – Disrespectful” in favor of the No-Emmpathy condition.

For a next study, we should change the empathic display, translate material in the native language of the observed population, select fewer hypotheses, decrease α to 10 or 5%, apply multiple testing corrections, and increase the number of participants. Based on our results and our definitions, we should make new metrics to evaluate perceived empathy with two scales—Empathic Understanding and Empathic Response. To evaluate interaction quality and perception of the robot, we should work on two questionnaires with different scales to avoid the multiplication of hypotheses. What follows from our study is that interaction quality should be evaluated in term of comfortableness and engagingness whereas the perception of the robot should be assessed regarding its perceived intelligence and trustworthiness.

ACKNOWLEDGEMENTS

This research was supported by the by the French Association Nationale de la Recherche et de la Technologie (ANRT) contract 2017/0226 and the joint laboratory BEHAVIORS.AI funded by the French Agence Nationale de la Recherche (ANR) contract ANR-16-LCV2-0003-01.

\*Website: https://behaviors.ai
REFERENCES

[1] M. Asada, "Development of artificial empathy," Neuroscience research, vol. 90, pp. 41–50, 2015.

[2] G. Barrett-Lennard, "The relationship inventory: A technique for measuring therapeutic dimensions of an interpersonal relationship," in Micrographed paper read at the Annual Conference of the Southwestern Psychological Association in St. Augustine, Florida, 1959.

[3] C. Bartneck and J. Forlizzi, "A design-centred framework for social human-robot interaction," in Robot and Human Interactive Communication, 2004. ROMAN 2004. 13th IEEE International Workshop on. IEEE, 2004, pp. 591–594.

[4] C. Bartneck, D. Kulic, E. Croft, and S. Zoghbi, "Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots," International journal of social robotics, vol. 1, no. 1, pp. 71–81, 2009.

[5] C. L. Bethel and R. R. Murphy, "Survey of non-facial/non-verbal affective expressions for appearance-constrained robots," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 1, pp. 83–92, 2008.

[6] T. W. Bickmore, "Relational agents: Effecting change through human-computer relationships," Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[7] R. J. B. Blair, "Responding to the emotions of others: dissociating forms of empathy through the study of typical and psychiatric populations," Consciousness and cognition, vol. 14, no. 4, pp. 698–718, 2005.

[8] C. Breazeal, "Social interactions in hri: the robot view," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 34, no. 2, pp. 181–186, 2004.

[9] J. A. Cegala, "Interaction involvement: A cognitive dimension of communicative competence," Communication Education, vol. 30, no. 2, pp. 109–121, 1981.

[10] D. J. Cegala, G. T. Savage, C. C. Brunner, and A. B. Conrad, "An elaboration of the meaning of interaction involvement: Toward the development of a theoretical concept," Communications Monographs, vol. 59, no. 4, pp. 229–248, 1992.

[11] T. L. Chartrand and J. A. Bargh, "The chameleon effect: the perception–behavior link and social interaction," Journal of personality and social psychology, vol. 76, no. 6, p. 893, 1999.

[12] D. A. Coker and J. Burgoon, "The nature of conversational involvement and nonverbal encoding patterns," Human Communication Research, vol. 13, no. 4, pp. 463–494, 1987.

[13] J. H. Cramer, J. Goddijn, B. Wielinga, and V. Evers, "Effects of (in)accurate empathy and situational valence on attitudes towards robots," in Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on. IEEE, 2010, pp. 141–142.

[14] K. Dautenhahn and A. Billard, "Bringing up robots or—the psychology of socially intelligent robots: From theory to implementation," in Proceedings of the third annual conference on Autonomous Agents. ACM, 1999, pp. 366–367.

[15] M. H. Davis, Interpersonal reactivity index. Edwin Mellen Press, 1980.

[16] M. B. Donnellan, F. L. Oswald, B. M. Baird, and R. E. Lucas, "The mini-IPPM scales: tiny-yet-effective measures of the big five factors of personality," Psychological assessment, vol. 18, no. 2, p. 192, 2006.

[17] J. W. Duran and P. H. Spitzeberg, "Toward the development and validation of a measure of cognitive communication competence," Communication Quarterly, vol. 43, no. 3, pp. 259–275, 1995.

[18] K. L. Fode, "The effect of non-verbal and non-verbal interaction on empathy and human-robot interaction (hri): A preliminary study," in Adaptive and Emergent Behaviour and Complex Systems, 2009.

[19] S. Tisseron, Le jour où mon robot m’aimera : vers l’empathie artificielle [The day my robot will love me: towards artificial intelligence]. Albin Michel, 2015.

[20] TNS Opinion & Social, “Special Eurobarometer 382: public attitudes towards robots.” Tech. Rep., 2012.

[21] J. Vallverdu, T. Nishida, Y. Ohmoto, S. Moran, and S. Lazare, “Fake empathy and human-robot interaction (hri): A preliminary study,” International Journal of Technology and Human Interaction (IJTHI), vol. 14, no. 1, pp. 1–48, 2019.

[22] P. A. Van Lange et al., Bridging social psychology: Benefits of transdisciplinary approaches. Psychology Press, 2006.

[23] J. Terken, “Attitude of elderly towards a robotic game-and-train-buddy: An evaluation of empathy and objective control,” in Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction. ACM, 2018, pp. 224–232.

[24] D. Spearman, "The proof and measurement of association between two things," The American journal of psychology, vol. 15, no. 1, pp. 72–101, 1904.

[25] S. Stel, R. B. Van Baaren, and R. Vonk, "Effects of mimicking: Acting prosocially by being emotionally moved," European Journal of Social Psychology, vol. 38, no. 6, pp. 965–976, 2008.

[26] S. Stel and R. Vonk, "Mimicry in social interaction: Benefits for mimickers, mimickers, and their interaction," British Journal of Psychology, vol. 101, no. 2, pp. 311–323, 2010.

[27] Student, "The probable error of a mean," Biometrika, pp. 1–25, 1908.

[28] D. S. Syrdal, K. Dautenhahn, K. L. Koay, and M. L. Walters, "The negative attitudes towards robots scale and reactions to robot behaviour in a live human-robot interaction study," Adaptive and Emergent Behaviour and Complex Systems, 2009.

[29] S. Tisseron, Le jour où mon robot m’aimera : vers l’empathie artificielle [The day my robot will love me: towards artificial intelligence]. Albin Michel, 2015.

[30] TNS Opinion & Social, “Special Eurobarometer 382: public attitudes towards robots.” Tech. Rep., 2012.

[31] J. Vallverdu, T. Nishida, Y. Ohmoto, S. Moran, and S. Lazare, “Fake empathy and human-robot interaction (hri): A preliminary study,” International Journal of Technology and Human Interaction (IJTHI), vol. 14, no. 1, pp. 1–48, 2019.

[32] P. A. Van Lange et al., Bridging social psychology: Benefits of transdisciplinary approaches. Psychology Press, 2006.

[33] J. Terken, “Attitude of elderly towards a robotic game-and-train-buddy: evaluation of empathy and objective control,” in Proceedings of the Doctoral consortium, in the scope of ACII2007 Conference, 2007.
A ROS architecture for a Cognitive Social Robot

Ignazio Infantino, Agnese Augello, Umberto Maniscalco, Giovanni Pilato, Pietro Storniolo, Filippo Vella
ICAR-CNR, Istituto di Calcolo e Reti ad Alte Prestazioni,
Consiglio Nazionale delle Ricerche, Via U. La Malfa 153, 90146, Palermo, Italy

Abstract—The paper presents a framework allowing a robot to socially interact with human beings, sharing with them some basilar cognitive mechanisms. Robust sensing of the environment and people is strongly linked with the cognitive perceptive low level and influences its motivation. Both long-term memory and short-term memory store relevant data to detect and recognize the social context (and social practice), and the human social behavior. Using both internal and external evaluations, the robot learns and improves its social skills, which take into account its physiological and emotional demands (affiliation, competence, certainty). Social interaction is encoded in the cognitive architecture by considering at the same level the human understanding and the robot communicative actions. This is done by implementing a suitable ROS architecture that allow the robot to use the same human interaction channels (both verbal and nonverbal).

Keywords—Cognitive Architectures, Social Robots, Human-Robot Interaction.

I. INTRODUCTION

In the future, social robots will effectively collaborate with people if they will be able to build a real social connection [9][10]. In such a case, both robots and humans have to create a stable and positive relationship (also including an interpersonal influence) based on mutual attentiveness and responsiveness. Humans are fundamentally cognitive emotional beings, and robots have to recognize and interpret affective signals and build suitable cognitive representations that include emotions, motivations, expectations, and also the effects of the internal physical states [6][17][24][25]. Naturally, the environment can strongly influence the social interaction and it often determines the social context and behavior. Then, the robot has to interpret and interact with humans within the correct social practice, switching also among different social contexts and roles through a proactive behavior [13]. In the past, many approaches have simply exploited the causal connections between cognition and emotion using the classical psychology models such as appraisal theory. But to assure a deep understanding and interpretation of the social-emotional displays, we suppose that a whole cognitive framework should inspire the design of the robot software architecture. A suitable cognitive architecture could allow the social robot to develop their socio-cognitive skills within a sort of Theory of Mind, (also known as mental perception, social commonsense, folk psychology, social understanding). In fact, the robot needs the ability to recognize, understand, and predict the human behavior regarding the underlying mental states such as beliefs, intents, desires, feeling [12]. Moreover, the artificial cognitive modules could determine a social behavior of the robot that adheres to the people expectations, their reasoning and their manner of acting. In a specular way, given that people apply a theory of mind to understand the robot regarding these mental states as well, the social interaction could be more natural and support the building of a robust social rapport.

II. THE COGNITIVE ARCHITECTURE

In previous works, we introduced and explained various modules of a cognitive architecture based on emotions and motivations [1][2][4][5][7][14][15]. Such architecture has been employed with success in different domains and experimentations in real environments. It allowed the robot to interact naturally with normal people, to learn by human examples, to improve its performances considering human feedbacks, to show high-level cognitive capabilities such as creativity. In the present paper, we explain the whole architecture by considering all the abstraction levels and dealing with the whole loop perception-reasoning-acting during social interaction. Figure 1 shows the schema of the proposed cognitive architecture for the fully interactive social robot.

A. Sensing Capabilities and Demands

The robot is an embodied artificial agent that, firstly, has to perceive itself to evaluate the external environment. Many robotic applications avoid considering such aspect given that the focus is on the execution of a task (often manipulation, and navigation). The state of the robot is monitored to guarantee the successful execution of the sequence of commands and the robot integrity, and alerts cause an immediate stop. Before reaching such critical conditions, the internal state the robot could be used to modulate its behavior and to obtain a simple self-representation with respect the extern useful to build artificial feelings and emotions. In [23] we proposed a somatosensory system that processes different variables of the robot: the temperatures and currents of joints, the battery charge level, the values from the gyroscope, sonars, laser, and other sensors. The proposed model, based on a soft sensor-based approach, allows the robot to own its roboceptions, that, such as human sensations and feelings, are the basis for computing a more sophisticated model of emotions. Roboceptions are strictly related to wellness state of the robot and naturally contribute to influence the motivation.
The internal sensing is also coupled with some physical feelings caused from external environment: for example, the presence or absence of people, the detection of noise or silence, the presence/absence of obstacles, the level of illumination. External sensing, before being processed to understand and manipulate the environment, causes some physical reaction, as is in living beings. Both internal and external roboceptions determine the physiological state of the robot and constitute one of the four drives (or demands) of the architecture.

For example, in our experimentation of social interaction, we implemented the following roboceptions: pain (from current flowing in the joints), fatigue (from the temperature of the joints), social space or proxemics discomfort (from sonar), caress pleasure (from touch sensors), noise discomfort (from microphones). The algorithm that realizes the soft sensor tries to somehow emulate a biological mechanism by taking into account: the temporal evolution, the memory and the cumulative effects, the latency. A suitable normalization and a weighted sum allows the system to compute various global functions such as well-being, general discomfort, or general distress [23].

The social robot, as a human that someone defines as a social animal, has an innate need to socialize. The affiliation component manages such inclination, and elementary tasks could be automatically executed to satisfy this demand. For example, the robot could look for faces and people, or it could detect and localize human voices and sounds. Each drive has a set of predefined actions to do for increasing the level of satisfaction required by the corresponding demand. Finally, competence and certainty demands consider the capability of the robot to manipulate the environment and its inclination to act. The competence demand is associated to the level of expertise or proficiency with respect an action, a task, or a goal. The robot has to build a collection (a repertoire) of robust and efficient procedures to accomplish various tasks and store them in its Long Term Memory (LTM). The competence demand could be generic or referred to a specific domain and could be measured by counting the number of different procedures available. Competence has to include external evaluation to select good procedures and to discard non satisfactory ones. In the case of an artistic performance, the audience feedback will influence the future executions [3].

The certainty demand deals with the confidence to accomplish a task and it depends from the previous successes or failures with respect some internal evaluation mechanisms (for instance, the expectation arising from the differences with a simulated model). By using suitable weights on the parameters related to the drives, we define the personality of the artificial embodied agent: shy, introverted, gruff, friendly, curious, expansive, sociable, and so on.

B. Knowledge Representation

The collection of procedures in LTM that constitute the expertise of the robot arises from learning phases. An instructor drives the learning by explicitly explaining or by showing examples (learning by example or imitation learning) [22]. Moreover, the instructor evaluates the executions to
allow the robot to improve over the time. In previous works, we used Interactive Genetic Algorithms to guarantee the evolution of the learning level of the robot [20]. Elementary predefined action modules could be combined (following given rules) to generate various execution planes and they could be selected by a genetic algorithm. To represent the linguistic knowledge of the robot, we use the Artificial Intelligence Markup Language (AIML), that is an XML instance for creating natural language software and used in the implementation of some popular chat-bots. Social procedures (or social practices) could be organized for different social contexts and learned by a social instructor. The working memory (or Shor Term Memory, STM) has to detect the features useful to classify and detect the social context. We employ neural networks such as Self Organizing Maps (SOM) to learn to associate contexts to a set of feature values.

In the case of the interaction with a human, the robot detects the social context by recognizing people, objects, facial expression, social signals, and so on. STM, in our architecture, also infers human emotions and compute robot motivation from the drives.

C. Relevant Issues on Social Interaction

Depending on the behavior, Breazeal [9][10] proposed four classes of social robots: socially evocative, social interface, social receptive, sociable. Robots belonging to the first two classes, rely on the human tendency to anthropomorphize, and perform just predefined actions perceived as natural, but avoiding a real interaction. Socially receptive robots are considered socially passive, but they can benefit from interaction (e.g., learning skills by imitation). Only at the level of a sociable robot, the artificial embodied agent pro-actively engages with a human to satisfy internal social aims (drives, emotions). Naturally, such a robot requires a deep model of social cognition. In our architecture, we use natural and intuitive communication channels, both to interpret the human behavior and to transfer knowledge to the human. Using natural communication channels includes: the use of natural language and a realistic speech synthesizer; a robust natural speech processing unit; detection and classification of non-verbal cues such as social signals; the generation of non-verbal robot actions to convey emotions and intentions (see the animated say module in the architecture).

A satisfactory verbal interaction [24] requires processing unstructured sentences to infer grammatical and semantic content, searching the appropriate reply in the large repository of knowledge. Considering an active interaction, the robot should drive the evolution of the verbal interaction, looking for acquiring some information from the person such as preferences, desiderata, demands. The robot could ask the person to confirm its understandings or it could require more details (verbally) react in an appropriate way (for example in the case of ambiguity). To fully understand the human, in the future, the robot has to manage the highest level of a situated language that includes abstract things and concepts both in time and space. Moreover, to assure a robust and not ambiguous social interaction, the robot has to connect the situated language to the physical context performing the so-called symbol grounding, and in some context to perform a meaning negotiation process. At present, we use standard cloud application for robust Natural Language Processing (NLP), and a simple ALICE chat-bot that allow the robot to own a minimalist, stimulus-response language. For instance, the experiment depicted in figure 2 shows a simple social interaction task (based on AIML) to perform a drawing collaborative task. The robot verbally interacts with the user, detects a face, and uses data from a social app to propose a digital collage.

An important dimension of cognition is the affective/emotional one. The affective dimension is very important in human interaction because it is strongly intertwined with learning, persuasion, and empathy, among many other functions. For the case of speech, affect is marked both in the semantic/pragmatic content as well as in the prosody of speech and the execution of non-verbal movements (head nods, deictic gestures, gaze movements). The Mood Modulator is responsible in the architecture of the affective modulation of the robot communication.

D. Reasoning and Robot Social Intelligence

A probabilistic reasoning approach allows the robot to manage uncertainty and the lack of knowledge that is typical in real social interactions. Powerful representations and processing formats are Bayesian Networks (BN) and Markov Decision Processes (MDP), that in the field of knowledge representation and reasoning adopt Bayesian probability for managing evidence and approximation. At present, we exploited a Hidden Markov Model (HMM) based approach, previously used for creative execution of dance [11][19][21].

Reasoning capabilities are the same required both in generic tasks execution and in social interactions: prediction (often called temporal projection), by inferring what will (probably) happen if the intended course of social action is executed; envisioning, by inferring (all) possible events and effects that will happen if a social practice (as a given sequence of social actions) gets executed in a hypothetical detected social context; diagnosis, by inferring what caused a particular reaction in social practice execution. The social intelligence should allow the robot deciding what is the most appropriate way of interacting with the human in the detected
positive and stable social relationship between humans and robots can provide social support in different domains (e.g., education, mental health, physical health, aging). Such a social robot could give relevant social support in different ways: instrumental, informational, emotional, and companionship. Through complex social capabilities, these robots, for instance, could assist people in various ways (by providing information, monitoring performance, incentivizing and sustaining motivation, giving encouragement).

III. THE ROS ARCHITECTURE

We are experimenting with the proposed framework by a suitable ROS architecture presented in Figure 3. The architecture defines main modules and topics that allow the robotic system to assure a cognitive control of the social interaction with a human user. The various scripts belong to the following classes: People Perception, Verbal Interaction, and Non-verbal Interaction. Robot motivation and mood influence the social interaction that requires to adapt to a specific person (People_X), to the detected social context (SocialContext_Y), and to pursue a particular social goal (e.g., reception, affective support, entertainment).

Modules run on a distributed computational architecture, allowing the system to overcome the limitations of the robot computational resources and performances. In fact, the verbal interaction takes advantage of the use of robust speech recognition API such as those provided by Google or similar. Non-verbal interaction uses a deep-learning approach to recognize and classify human action and to produce similar social signs performed by the robot. The core module of the architecture, the Cognitive Controller, is coherent with the framework proposed in previous sections and determines the social behavior. Also, the social interaction requires various levels of evaluation to assure an acceptable behavior and social practice.

The presented system describes a general-purpose architecture for social robots that can exhibit human-like cognitive capabilities. The architecture can integrate various approaches to perform decision making, planning, and reasoning that can take advantage of a simplified cognitive model that explicit motivation, emotion, and feelings.

The aim is to combine perceptions of inner physical states, of the external environment, of peoples with internal and external evaluation mechanisms, allowing the robot to interact with the human showing complex behaviors useful to execute “social practices” and to assure “social support.” We are developing a suitable ontology and formalisms to support the knowledge representation based on processing perceptions. In preliminary experiments, since we utilized the humanoid robot Softbank's Pepper, we extend the NaoQi framework based on shared memory (memory) that stores sensors values and events and can represent the working memory. Long-term memory arises from multiple self-organizing maps to classify and represent actions, postures, facial expressions, verbal cues, and all entities that influence the social behavior of the robot. Learning and the evolution of the robot capabilities depend on evaluation based on interactive genetic algorithms.
References

[1] A. Augello, I. Infantino, A. Manfré, G. Pilato, F. Vella, Analyzing and discussing primary creative traits of a robotic artist, Biologically Inspired Cognitive Architectures, vol. 17, 22-31, 2016

[2] A. Augello, I. Infantino, A. Lieto, G. Pilato, R. Rizzo, F. Vella, Artwork creation by a cognitive architecture integrating computational creativity and dual process approaches, Biologically Inspired Cognitive Architectures, vol. 15, pp. 74-86, 2016

[3] A. Augello, I. Infantino, A. Manfré, G. Pilato, F. Vella, A Chella, Creation, and cognition for humanoid live dancing, Robotics and Autonomous Systems, vol. 88, pp 107-114, 2016.

[4] A. Augello, I. Infantino, G. Pilato, R. Rizzo, F. Vella, Creativity evaluation in a cognitive architecture, Biologically Inspired Cognitive Architectures, vol 11, pp. 29-37, 2015

[5] A. Augello, I. Infantino, G Pilato, R Rizzo, F Vella, Robotic creativity driven by motivation and semantic analysis, Proc. of Semantic Computing (ICSC), 2014 IEEE International Conference on, pp. 285-289, 2014

[6] A. Augello, I. Infantino, G Pilato, R Rizzo, F Vella, Combining representational domains for computational creativity, Proc. of the 5th international conference on computational creativity, ICCC 2014, Ljubljana, Slovenia, 2014.

[7] A. Augello, I. Infantino, G. Pilato, R. Rizzo, F. Vella, Introducing a creative process on a Cognitive Architecture, Biologically Inspired Cognitive Architectures, Vol. 6, pp. 131-139, Elsevier, 2013.

[8] A. Augello, I Infantino, G Pilato, R Rizzo, F Vella, Binding representational spaces of colors and emotions for creativity, Biologically Inspired Cognitive Architectures, vol. 5, pp. 64-71, 2013

[9] C. Breazeal, Toward sociable robots, Robotics and Autonomous Systems 42 (2003) 167-175.

[10] C. Breazeal, K. Dautenhahn, T., Social Robotics, Springer Handbook of Robotics, Edited by B. Siciliano and O. Khatib, chapter 73, pp. 1935-1960, 2016

[11] G. Città, S. Arnab, A. Augello, M. Gentile, S.I. Zielonka, D. Ifenthaler, I. Infantino, D. La Guardia, A. Manfré, M. Allegra, Move Your Mind: Creative dancing humanoid as support to STEAM activities, KES-HMSS 2017, Cognitive Systems and Robotics, Algarve, Portugal, 2017

[12] K. Dautenhahn: Getting to know each other – Artificial social intelligence for autonomous robots, Robotics and Autonomous Systems 16, 333–356 (1995)

[13] T. Fong, I. Nourbakhsh, K. Dautenhahn, A survey of socially interactive robots, Robotics and Autonomous Systems 42 (2003) 143–166

[14] G. Gaglio, I. Infantino, G. Pilato, R. Rizzo, F. Vella (2011). Vision and emotional flow in a cognitive architecture for human-machine interaction. Frontiers in Artificial Intelligence and Applications, Vol. 233: Biologically Inspired Cognitive Architectures 2011, pp. 112 – 117

[15] I. Infantino, Affective Human-Humanoid Interaction Through Cognitive Architecture, The Future of Humanoid Robots - Research and Applications, Dr. Riadh Zaier (Ed.), InTech, 2012

[16] I. Infantino, A. Manfré, U. Maniscalco, Robot Navigation Based on an Artificial Somatosensory System, First International Early Research Career Enhancement School on Biologically Inspired Cognitive Architectures, pp. 72-77, Springer, 2017

[17] I. Infantino, G. Pilato, R. Rizzo, F. Vella, I Feel Blue: Robots and Humans Sharing Color Representation for Emotional Cognitive Interaction, Biologically Inspired Cognitive Architectures 2012, pp 161-166

[18] I. Infantino, G. Pilato, R. Rizzo, F. Vella, I Feel Blue: Robots and Humans Sharing Color Representation for Emotional Cognitive Interaction, Biologically Inspired Cognitive Architectures 2012, pp 161-166

[19] I. Infantino, A. Augello, A. Manfré, G. Pilato, F. Vella, ROBODANZA: Live Performances of a Creative Dancing Humanoid, Proceedings of the Seventh International Conference on Computational Creativity, Paris, 2016

[20] A. Manfré, A. Augello, G. Pilato, F. Vella, I. Infantino, Exploiting interactive genetic algorithms for creative humanoid dancing, Biologically Inspired Cognitive Architectures, vol. 17, 12-21, 2016

[21] A. Manfré, I. Infantino, F. Vella, S. Gaglio, An automatic system for humanoid dance creation, Biologically Inspired Cognitive Architectures, vol.15, pp.1-9, 2016

[22] A. Manfré, I. Infantino, A. Augello, G. Pilato, F. Vella, Learning by demonstration for a dancing robot within a computational creativity framework, IEEE International Conference on Robotic Computing, Taichung, Taiwan, 2017.

[23] U. Maniscalco, I. Infantino, A. Manfré, Robust mobile robot self-localization by soft sensor paradigm, Robotics and Intelligent Sensors (IRIS), 2017 IEEE International Symposium on, Pages 19-24, IEEE, 2017

[24] Nikolaos Mavridis, A Review of Verbal and Non-Verbal Human-Robot Interactive Communication, https://arxiv.org/pdf/1401.4994.pdf

[25] B. Murthi, N. Roy, S. Šabanović, Cognitive Human-Robot Interaction, Springer Handbook of Robotics, Edited by B. Siciliano and O. Khatib, chapter 71, pp. 1935-1960, 2016

[26] Mary-Anne Williams, Robot Social Intelligence, in Proc. of Social Robotics: 4th International Conference, ICSR 2012, Chengdu, China, October 29-31, 2012
Orthographic Vision-based Interface for Robot Arm Teleoperation

Waleed Uddin, Maram Sakr, Camilo Perez Quintero and H.F. Machiel Van der Loos

Abstract—Robot teleoperation is crucial for many hazardous situations like handling radioactive materials, underwater exploration and firefighting. Visual feedback is essential to accurately teleoperate a robot. Existing solutions to improve teleoperation involve the use of multiple cameras, expensive sensors, depth cameras or VR/AR headsets. These systems, however, have some limitations including: safety hazards, complexity, cost, and inadaptability. Contrary to the existing work, we provide a simple, cost-effective and intuitive teleoperation system by visualizing the remote environment in an effective way to provide depth information using only one inexpensive webcam. To validate our system we perform a pilot study where users teleoperate 6-DOF arm and gripper to complete a pick and place task. We compare our proposed interface with a binocular camera setup. In addition, we test three input modalities with our interface: joystick, keyboard and Leap Motion. We use completion time and object manipulation accuracy as evaluation metrics. Results from the pilot study suggest that our interface in comparison with the binocular camera visualization improves completion time by 64%, 43% and 41% for the joystick, keyboard and Leap Motion, respectively. Furthermore, the number of errors declined using our vision system regardless of the control modality used.

I. INTRODUCTION

Teleoperation refers to a task done by a robot which is remotely controlled by a human operator. Over the years, the use of teleoperation has become popular in several areas such as military [1], space exploration [2], underwater exploration [3] and tele-surgery [4]. The teleoperation system’s performance is directly affected by the sensory information, visualization of the remote environment, control interface and operator capabilities. Teleoperating a robot is a cumbersome task for non-experts if the system is unintuitive. Humans prefer natural communication and control interfaces with the robot. To get such a system in a teleoperation setup, an informative and simple visual interface with an intuitive control system is crucial.

Thus, our paper focuses on these two key components of the teleoperation system. Conventional input modalities such as a joystick are considered non-intuitive and are detrimental to overall task efficiency [5]. With the advent of motion-sensing devices, HRI researchers have shifted focus on devising more intuitive teleoperation interfaces [6]. Quintero et al. [7] recently developed a novel semi-autonomous means to control a robotic arm. Using the Kinect skeleton tracker, they mapped the human arm joints to robot joints, providing fast but coarse positioning of the arm. To mitigate this effect they introduce a visual servoing interface for fine positioning of the arm. Kim et al. [8] presented a master-slave direct control interface for an excavator using sensors placed on the operator’s hand. The results of that study in comparison with joystick control were promising but not conclusive. These input strategies have not successfully been put into commercial use primarily due to safety concerns, design complexity, equipment costs and inadaptability.

In addition to the input modality adapted for teleoperation, the level of enrichment and information provided in a feedback modality is crucial. A camera is typically used to provide visual feedback of the teleoperation environment to the operator. However, the 2D video seen on a screen fails to provide depth information for the environment. Without depth perception, the operator is likely to make errors during teleoperation [9]. To solve this problem, people have tested teleoperation with multiple cameras placed at different locations to acquire depth perception for the environment and improve task efficiency [10], [11]. However, the option of using multiple cameras is costly, adds complexity to the system and requires additional space. Moreover, using another camera can also cause object occlusion [12]. Vision-based high-fidelity depth cameras like the Microsoft Kinect in teleoperation systems have also been proposed [7] along with recent advances in AR and VR devices, which provide immersion and telepresence to the operator. Peppoloni et al. [13] implemented a 3D augmented reality-based visual feedback system to teleoperate a Baxter robot.

These efforts to solve the problem of depth perception are commendable; however, the efficacy of these systems relies on expensive sensors or cameras, and the design itself may pose additional complexity. Furthermore, these systems may require special training to gain familiarity. Prolonged use of VR or AR headsets in teleoperation systems may also cause VR sickness [14].

Building on this existing work, we provide a simple, cost-effective and intuitive teleoperation system by focusing our efforts on visualizing the remote environment in an effective way to provide depth information using only one inexpensive webcam. In addition, we provide a comparison between three control modalities: joystick, keyboard and the Leap Motion. Thus we aimed for finding the best modality combined with our proposed visual system to achieve a balance in task completion time and object manipulation accuracy. Our main contributions in this paper are the following:

- Providing depth information along with visual feedback in teleoperation system using only one inexpensive ordinary camera.
- Comparing between different input modalities for robot
In the rest of this paper, we describe the proposed system in section II. Then, we present the pilot user study that we conducted to evaluate our system and compare between different modalities in section III. Section IV provides the results of the study, then a discussion of the results is provided in section V. Lastly, we conclude the paper and propose some future directions in section VI.

II. SYSTEM DESCRIPTION

A. Overview

We propose a direct unilateral and cartesian-based control of a 6-DOF robot in real-time. The operator can control the arm using one of three modalities: joystick, keyboard or Leap Motion [15] (hand motion tracking device). Fig. 1 shows our system in its simplified form when Leap Motion controller is being used as input method. This Leap Motion controller is connected to a local computer using a serial link. An external camera is responsible for providing a view of the remote environment and sending it to the local computer. Using computer-vision techniques, depth information of the remote location is added to the camera view to make it easier for the user to teleoperate the robot.

Fig. 2 shows the physical setup of our teleoperation system. In the remote location, a robotic arm is used to accomplish a pick and place task. The object to be picked is placed on top of a box that has a QR code to help in capturing its position with the camera. While in the operator’s location, a computer is used to process the camera views and to handle the communication between the robot and the control interface. Distances between camera and the arm as well as the arm end-effector with respect to its origin are shown in Fig. 3. The camera is fixed at a distance such that its field of view captures the robot and its surrounding environment. These distances are crucial for creating an effective visualization.

Our system is evaluated on a 6-DOF Kinova Jaco2 arm. Actuators are geared DC servomotors, which operate at 24VDC and have built-in encoders for sensing joint angles. For our research, we operate it in Cartesian control mode. Communication with the Kinova arm is done through a USB port. Commanded Cartesian commands are sent to the robot. The robot controller calculates and rotates respective joints through its inverse kinematics module to achieve the commanded Cartesian pose. The arm’s internal joint encoders provide actual pose information back to the computer, where it can be visualized along with the commanded pose. Our software is composed of four modules that interact with each other simultaneously, as shown in Fig. 4.

The main communications with the robot’s DSP using the Kinova API are functions that send a desired Cartesian trajectory and get an actual Cartesian pose. Parameters that are passed are in the form of a data structure. The arrays ”ActualPose” and “CommandedPose” hold Cartesian information about actual and commanded pose. They contain the following float variables:

\[
\text{ActualPose} = [x_p, y_p, z_p, \theta_{xp}, \theta_{yp}, \theta_{zp}, f_{1p}, f_{2p}, f_{3p}] \quad (1)
\]

\[
\text{CommandedPose} = [x_t, y_t, z_t, \theta_{xt}, \theta_{yt}, \theta_{zt}, f_{1t}, f_{2t}, f_{3t}] \quad (2)
\]

These structures are passed to the Jaco2’s API functions. The \( f \) variables represent fingers of the arm. Variables \( x, y \) and \( z \) are the Cartesian coordinates while \( \theta \) variables are wrist coordinates.

B. Input Modalities

We control the arm in Cartesian mode, i.e., moving the end-effector in \( x \), \( y \) and \( z \) directions directly. Using our teleoperation system, arm can be controlled via joystick, keyboard or the Leap Motion controller as follows:
**Joystick:** A joystick is available from the manufacturer to control the robot. Rotating it left or right moves the robot’s end-effector sideways in the x-axis while forward and backward moves it in the y-axis. For movement along z-axis (up and down), the joystick handle is rotated clockwise or counter-clockwise.

**Hand Motion Tracking:** We use the Leap Motion controller for real-time marker-less hand motion tracking. This controller is geared to small-scale VR development applications. Fig. 1 shows the Leap Motion in our system. Hand movements are captured by the Leap Motion controller at a rate of approximately 115 Hz. Hand coordinates x, y, z in meters are sent to a local computer. A program script in C# handles the incoming data from the Leap Motion and communicates with the Kinova arm through its API over a separate USB link. The processed x, y, z values are finally sent to the robot controller in the form of a Cartesian command data structure.

**Keyboard:** The robot end-effector moves forward with the W key, backward with the S key, left and right with the A and D keys, and up and down with the E and Q keys. The Space key is to grasp an object, i.e., the gripper closes, while the X key opens the gripper to drop the object. We chose these keys because the mapping has been used extensively with video games, making it easier for users to memorize.

**C. Visualization**

Our visual interface consists of an orthographic visualization of the scene as shown in Fig. 5. The environment is set up using the Unity 3D engine. The normal front view is provided to the user along with a vision-based top view. This top view is created using vision-based depth information.

A black ball represents the robot end-effector’s Cartesian movement in x, y, z directions in real-time in both views. A green ball represents the operator’s hand movements in x, y, z directions if the Leap Motion device is being used as the input modality. The green ball serves as the Cartesian command and the black ball as the follower. The blue cube in the top view is the representation of the physical box upon which the toy object is placed at center, calculated by the vision-based camera. The blue cube is not the representation of the toy rather it is the representation of its physical location. If the box is moved, the 2D representation of the box moves accordingly. Similarly, the green cube represents the target where the object is to be dropped. The problem of depth perception is solved using the marker-based vision system described as follows.

In our system, a single webcam is placed at a set distance from the robot as shown in Fig. 3 earlier. The camera has two purposes:

1) To provide visual feedback: a normal 2D frontal view of the robot and environment in x and z axis.

2) To use computer-vision for depth calculation: as shown in Fig. 6, vision-based camera computes distance of marker attached to the box, relative to itself based on a natural phenomenon of object perception. This depth information can be visualized on the screen and the operator can then control the arm accurately with perceptive information about all three dimensions.

The apparent size of the marker depends upon the visual angle experienced by the camera and not the actual size. As this visual angle seems to be proportional to the apparent size [16], we can detect changes in size of the marker.

As the real size of the marker is constant and known, any change in the apparent size of the marker would mean that the marker is either coming closer or moving away from the camera. If we move the marker towards camera, the apparent
size of the marker experienced by camera gets bigger. As the distance correlates with the apparent size [17], we can use it to compute the actual distance of object from the camera. Fig. 6 shows the described concept.

Distance of the marker relative to the camera is computed through computer-vision based API inside our 3D visualization software that calculates distance using the following function:

\[
 d_{\text{cam}} = f(S, p)
\]  

(3)

where, \( S \) is width of marker which is 25 cm, attached to a box. \( d_{\text{arm}} \) and \( d_{\text{cam}} \) are distances of marker relative to the arm’s base origin and the camera origin. \( p \) is the marker image size perceived by the camera. The camera and arm bases are fixed. \( d_{\text{cam}} \) is calculated first using camera vision API functions then \( d_{\text{arm}} \) is simply calculated as an offset.

The depth information \( d_{\text{arm}} \) is used to visualize the distance of the object in the y-axis with respect to the robot’s base location as shown in the top view of Fig. 5. By presenting this top view using single camera along with a normal frontal view to the operator, we provide perceptive information regarding all three dimensions to the operator.

Although researches involving marker-based object detection through computer vision have been conducted for Augmented-Reality (AR) systems in past [18], yet provision of depth perception utilizing such approach specifically for teleoperation systems is something that has not been explored before to the best of our knowledge.

III. USER STUDY

In order to assess the overall efficacy of our vision-based teleoperation system, we compared our system against a conventional teleoperation system consisting of two cameras displaying front and top view lacking any visualization aids or computer vision [19]. In addition, we compare between three input modalities: joystick, keyboard and leap motion using the two visual systems. We recruited three participants (two males, one female) from the University of British Columbia who had no prior experience with robot teleoperation. A consent form approved by the ethics committee was provided to each participant for signature prior running the experiment which included details of experiment. The study lasted 50-60 minutes. The participants were asked to complete a pick and place task which consists of picking up a small toy in a gentle way and moving towards the destination while avoiding obstacle and then dropping the toy in the target container. The obstacle is a wall, placed in the way to the target position. In addition, the toy was placed in the beginning on a box that can be opened if the participant pressed hard on it while picking the object up. This allows us to judge if the participants pick up the object gently or not. Each participant went through the following steps to accomplish the task:

A. Procedure

- Participants were introduced to the experimental setup via verbal briefing. Then, each participant was provided five minutes to familiarize himself with the system which includes the three input modalities and two visual interfaces. The participant was only allowed to look into the screen that showed either the top and front view using standard vision system or one camera view with marker vision-based top view using our proposed system.
- Using the standard vision system, participants were asked to complete the pick and place task using Joystick, Leap controller and Keyboard three times each, with total of nine trials, provided in a random order.
- Then using our vision-based system, participants completed the same task using the three input modalities for nine trials in a random order to fairly compare between the three modalities.
- At the end, participants were asked to complete a survey involving qualitative metrics to rate the performance of each input modality subjectively.

B. Performance metrics

To evaluate our system, we used the following as performance metrics:

- Task Completion time: this is the time taken by the participant to complete the task. We recorded the completion time in each trial for all three input modalities and the two visual interfaces.
- Number of errors: the errors we refer to in this context are the cases in which the participant pressed the box underneath the object to pick up (tough pickup). We also counted the attempts to pick up the object, if the participant did not successfully pick it up the first time. Also, when the participant hit the wall between the source and destination. Lastly, the case in which the participant did not drop the object in its target position.

Using the above metrics, we compared between the three modalities using both standard vision system (two cameras) and our proposed orthographic vision system. In addition,
we measured the performance improvement rate for each modality.

IV. RESULTS

In this section, we present results for task completion time and total number of errors collected from all 3 participants during pilot experiments. Fig. 7 shows the completion time averaged across all participants and trials for the three modalities: joystick, keyboard, and leap motion using the standard vision system and our proposed system. It is clearly shown that, our proposed vision system outperforms the standard system regardless the input modality used. Our system shows a significant decrease in task completion time with 64%, 43%, and 41% compared to the standard system using joystick, keyboard, and leap motion, respectively. In addition, with our proposed vision system using leap motion shows the lowest completion time with an average of 28.67 ± 6.58 compared to 39.78 ± 6.28 using keyboard and 48.33 ± 12.59 using joystick. Using the same performance metric, we also noted the general trend of the performance improvement within the training trials of each input modality with both visual interfaces as shown in Fig. 8. Overall, the task completion time decreased with the trials increased except with the case of using joystick with the standard vision system. In addition, it is clearly shown that joystick improvement rate is higher than the rate of both keyboard and Leap Motion. Also, Leap Motion achieves the lowest completion time in all trials compared to the other two modalities.

In terms of committed errors as shown in Fig. 9, the number of errors is comparable among the three modalities. Similar to completion time, our proposed vision system shows higher performance than using the standard system among all modalities. Using our vision system has average number of errors around 1 ± 1, 0 ± 1, and 1 ± 1 using joystick, keyboard, and leap motion, respectively. Compared to 4 ± 4, 2 ± 1, and 2 ± 1 using the same modalities, respectively with the standard vision system.

To assess the system performance in terms of perceived cognitive and physical workload, the participants were asked to fill out a NASA-TLX Load index form after the experiment had finished. Regarding satisfaction of task completion using each input modality, two participants rated Leap Motion controller above keyboard while the third participant rated keyboard first following the joystick commenting that even though Leap Motion-based control was intuitive yet it was physically more demanding than keyboard or joystick specially if the task is too long. Leap Motion outperforms other input modalities in terms of mental demand for all participants.

V. DISCUSSION

Results from the pilot study suggests that our orthographic vision-based system outperforms the conventional teleoperation system with binocular camera setup. From experiments, we learnt that the inferior performance of the standard system as manifested in results, was mainly due to poor depth perception and object occlusion visually experienced at the operator side. Although there are two cameras that cover the front and top views, the participants found that it is really hard to accurately reach the target position. In addition, the participants perceived the binocular camera system as more mentally-demanding than ours because they need to switch between the two views to accomplish the task. While using the marker-based camera view provided a sort of guidelines that helped to finish the task easier. Furthermore, using the Leap Motion controller with the marker-based vision system leverage its capabilities and helped the user to finish the task with higher performance. Rodriguez et al. [20] used the Leap Motion for teleoperating a WAM robot and they found that
the Leap Motion is faster than the other interfaces but has poor precision. This is because the lack of visual feedback about the hand position which was proven to be effective with our experiments. In comparison between various input modalities, we found that joystick seemed to perform poorly due to its discrete movements and non-intuitive control while the Leap Motion controller seemed to perform slightly better than keyboard providing a continuous control to the operator. In addition, the task completion time improvement rate among the three trials using joystick is higher than the other two modalities. This suggests that joystick needs more time for training and it is not intuitive for non-experts to use. While keyboard and Leap Motion showed a slight improvement which suggested that the users can easily use them. Regarding our single-camera system, one drawback is its inability to construct the top-view if something is blocking the camera’s view.

Conventional approaches in teleoperation such as binocular camera system, are more mentally-demanding than ours because they need to do mental calculations to perceive the depth compared to our system that presents the depth information in a simple way. This was proved from what Marble et al. [21] concluded from their study, in which most of their participants indicated a desire to have visual feedback in the teleoperation system presented with depth indicators rather than to have to deduce the depth from the interface.

This pilot study compellingly verifies the possibility of using hand motion-capture system coupled with a simple yet effective orthographic vision-based interface to greatly enhance the efficacy of teleoperation tasks.

VI. CONCLUSION AND FUTURE WORK

In this paper, we addressed the fundamental problems of perception and control experienced by the operator related to teleoperation systems. We put efforts on providing a simple, cost-effective and intuitive teleoperation system of a 6-DOF robot arm in Cartesian mode. We focused on visualizing the remote environment in an effective way by providing depth information using only one inexpensive webcam. In addition, we provided a comparison between three control modalities: joystick, keyboard and Leap Motion. Our pilot study involved three participants and consisted of a ‘pick and place’ task. Experiments involved comparison of our vision-based camera system with a conventional system consisting of two cameras that provide visual feedback to the operator. We tested task performance of both systems using joystick, keyboard and Leap controller. Results from Pilot studies showed that our vision-based system outperforms the conventional teleoperation system in terms of efficiency and accuracy. Among three input modalities, Leap Motion controller slightly outperformed the keyboard while joystick performed poorly. Since our pilot study results favor our proposed interface, we therefore in future, plan to conduct an extensive user study by involving more participants and evaluate our visual system and control modalities with different tasks. In addition, we plan to address problems related to the Leap Motion interface in terms of its physical demand in the long tasks.

REFERENCES

[1] T. Kot and P. Novák. Application of virtual reality in teleoperation of the military mobile robotic system taros. International Journal of Advanced Robotic Systems, 15(1):1729881417751545, 2018.
[2] G. Liu, X. Geng, et al. Haptic based teleoperation with master-slave motion mapping and haptic rendering for space exploration. Chinese Journal of Aeronautics, 2018.
[3] R. Saltaren, R. Aracil, et al. Field and service applications-exploring deep sea by teleoperated robot-an underwater parallel robot with high navigation capabilities. IEEE Robotics & Automation Magazine, 14(3):65–75, 2007.
[4] J. Marescaux, J. Leroy, et al. Transatlantic robot-assisted telesurgery. Nature, 413(6854):379, 2001.
[5] N. Mavridis, G. Pierris, et al. Subjective difficulty and indicators of performance of joystick-based robot arm teleoperation with auditory feedback. In Advanced Robotics (ICAR), 2015 International Conference on, pages 91–98. IEEE, 2015.
[6] B. Fang, F. Sun, et al. A novel data glove for fingers motion capture using inertial and magnetic measurement units. In Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on, pages 2099–2104. IEEE, 2016.
[7] C. P. Quintero, R. T. Fomena, et al. Interactive teleoperation interface for semi-autonomous control of robot arms. In 2014 Canadian Conference on Computer and Robot Vision (CRV), pages 357–363. IEEE, 2014.
[8] D. Kim, J. Kim, et al. Excavator tele-operation system using a human arm. Automation in construction, 18(2):173–182. 2009.
[9] J. S. Tittle, A. Roesler, et al. The remote perception problem. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, volume 46, pages 260–264. SAGE Publications Sage CA: Los Angeles, CA, 2002.
[10] M. Fischer and D. Henrich. Surveillance of robots using multiple colour or depth cameras with distributed processing. In Distributed Smart Cameras, 2009. ICDCS 2009. Third ACM/IEEE International Conference On, pages 1–8. IEEE, 2009.
[11] D. Saakes, V. Choudhary, et al. A teleoperating interface for ground vehicles using autonomous flying cameras. In 2013 23rd International Conference on Artificial Reality and Telexistence (ICAT), pages 13–19. IEEE, 2013.
[12] E. Hsiao and M. Hebert. Occlusion reasoning for object detection under arbitrary viewpoint. IEEE transactions on pattern analysis and machine intelligence, 36(9):1803–1815, 2014.
[13] L. Peppoloni, F. Brizzi, et al. Augmented reality-aided tele-presence system for robot manipulation in industrial manufacturing. In Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology, pages 237–240. ACM, 2015.
[14] D. M. Romano. Virtual reality therapy. Developmental medicine and child neurology, 47(9):580–580, 2005.
[15] L. Motion. Leap motion controller. URL: https://www.leapmotion.com, 2015.
[16] T. Konkle and A. Oliva. Canonical visual size for real-world objects. Journal of Experimental Psychology: human perception and performance, 37(1):23, 2011.
[17] W. Epstein. The known-size-apparent-distance hypothesis. The American Journal of Psychology, 74(3):333–346, 1961.
[18] Y. S. Villegas-Hernandez and F. Guedea-Elizalde. Markers position estimation under uncontrolled environment for augmented reality. International Journal on Interactive Design and Manufacturing (IHD-M), 11(3):727–735, 2017.
[19] T. Fong and C. Thorpe. Vehicle teleoperation interfaces. Autonomous robots, 11(1):9–18, 2001.
[20] D. Rodriguez, C. Perez, et al. A comparison of smartphone interfaces for teleoperation of robot arms. In Computer Conference (CLEI), 2017 XLIII Latin American, pages 1–8. IEEE, 2017.
[21] J. L. Marble, D. J. Bruemmer, et al. Lessons learned from usability tests with a collaborative cognitive workspace for human-robot teams. In Systems, Man and Cybernetics, 2003. IEEE International Conference on, volume 1, pages 448–453. IEEE, 2003.
Towards pedestrian-AV interaction: method for elucidating pedestrian preferences

Fanta Camara\textsuperscript{1,2}, Serhan Cosar\textsuperscript{2}, Nicola Bellotto\textsuperscript{2}, Natasha Merat\textsuperscript{1} and Charles W. Fox\textsuperscript{1,2,3}

Abstract— Autonomous vehicle navigation around human pedestrians remains a challenge due to the potential for complex interactions and feedback loops between the agents. As a small step towards better understanding of these interactions, this Methods Paper presents a new empirical protocol based on tracking real humans in a controlled lab environment, which is able to make inferences about the human’s preferences for interaction (how they trade off the cost of their time against the cost of a collision). Knowledge of such preferences if collected in more realistic environments could then be used by future AVs to predict and control for pedestrian behaviour. This study is intended as a work-in-progress report on methods working towards real-time and less controlled experiments, demonstrating successful use of several key components required by such systems, but in its more controlled setting. This suggests that these components could be extended to more realistic situations and results in an ongoing research programme.

I. INTRODUCTION

The potential future deployment of full Autonomous Vehicles (AVs) is currently creating much enthusiasm, as such vehicles would change our daily life through making transportation more efficient. Huge improvements have been made on robotic localisation and mapping problems using Simultaneous Localisation And Mapping (SLAM) algorithms \cite{26, 5} together with new, cheap sensors and computation technologies \cite{15} \cite{30}. ‘Self-driving’ cars can navigate safely on roads, promising a future society with a better mobility system with less accidents and traffic in cities.

But before the fully autonomous driving (SAE Level 5) revolution happens, AVs must share space with and will be challenged by human drivers and pedestrians, who are much harder to model and act upon than passive environments. Decades’ of research in the fields of Transport Psychology and Human Factors have not yet been translated into robotic control systems, and leave many questions still unanswered. For safety and legal reasons, pedestrians are considered as obstacles, such that the vehicle always stops for them, in most current ‘self-driving’ systems. Recent on-road studies have shown that pedestrians may then take advantage over AVs due this predictable behaviour \cite{22} \cite{20} \cite{13} \cite{8} \cite{4}, pushing in front of them for priority eventually in every negotiation such that the vehicles can then make no progress. This has become known as the ‘Freezing Robot Problem (FRP)’. Real human driving is massively more complex than simply mapping, localising and path planning. It is considered an art form by advanced practitioners such as members of the Institute for Advanced Motorists and other advanced drivers such as high-speed police and ambulance drivers. In their training, these practitioners generally emphasise the human psychological processes involved in predicting the behaviours of other road users as the most important skill of human drivers. Can you tell if a pedestrian is assertive enough to risk stepping out in front of you from their body language, their facial expressions, even their clothes and demographics? Road users have different utility functions, ranging from timid pedestrians likely to give way to all oncoming traffic, though to business-people late for a meeting or patients for an urgent medical appointment becoming much more assertive and risk-taking. Drivers must also consider the psychological effects of their own actions. Speeding up and slowing down are not just ways to control one’s progress but also send information about our own personality and risk preferences to pedestrians engaged in such negotiations for priority, along with other possible signals including lateral road positioning, and more conventional signals such as flashing indicator lights and headlights, and driver face and arm expressions.

Fig. 1: Two agents negotiating for priority at an intersection

The new EU H2020 interACT project has been created with a consortium of European partners \cite{14} to investigate the role of interaction in future deployment of AVs in mixed traffic environments with human drivers, cyclists and pedestrians. The project will aim to understand the behaviour of other road users, and how AVs could interact with them in a safe and efficient manner, and propose HMI solutions that could facilitate the communication between AVs and people.

As first steps towards these goals, we recently proposed \cite{10} and solved a very simple game-theoretical mathematical model of the unsigned road-crossing scenarios represented in figs. 1 and 2, based on the famous game of ‘chicken’ and called ‘sequential chicken’. In this simplest-possible model, two agents (which may be pedestrians and/or vehicles) compete for space at an unsigned intersection, using only their positions to signal information to one another. Time, space and actions are discretized and it is assumed that both players have equal utility functions and know this to be the case. The model leaves open free parameters specifying the utility function for human players. We proposed \cite{10} only as a mathematical model but suggested that its parameters could be found via human experiments. In \cite{6}, we experimented the model with human participants and asked them to play the sequential chicken game as a board game to measure their behavioral parameters.

The present study extends this idea to present a new protocol using physical subjects’ bodies in a semi-structured interaction scenario together with person tracking and Gaussian Process Regression analysis, to infer their preferences in a slightly more

\textsuperscript{1} Institute for Transport Studies (ITS), University of Leeds, United Kingdom

\textsuperscript{2} Lincoln Centre for Autonomous Systems (LCAS), School of Computer Science, University of Lincoln, United Kingdom

\textsuperscript{3} Ibex Automation Ltd, United Kingdom

This project has received funding from the European Union’s Horizon 2020 Research and Innovation programme under grant agreements No 723395 (InterACT) and No 645376 (FLOBOT).
realistic setting. As such it represents additional progress moving the model closer to the real world, though of course there are many more needed in the future as the work progresses, until we reach a real-time continuous version suitable for use in AVs. First, it makes use of pedestrian tracking to estimate the trajectories of the agents involved in semi-structured human-human interactions while playing the sequential chicken model. Second, it computes the optimal strategies using a meta-strategy convergence method [10]. Lastly, it infers the parameters of the interactions using Gaussian Process Regression. To our knowledge there is no previous work fitting tracking-based semi-structured game-theoretic models to human motions and to infer behaviour parameters.

A. Related work

The problem of self-driving cars interacting with other road users is raising interest in both the Robotics and Transport Studies communities. Game Theory offers a framework to model conflict and cooperation between rational decision-makers. It was developed in the 1940s by von Neumann and Morgenstern. Its core concept is (Nash) equilibrium which is the pair of strategies (probability distributions over actions to be played) such that none of the players would change their strategy if they knew the other’s strategy. Previous work in Transport Studies and highway design has applied game theory to several driver behaviour modelling tasks as reviewed in [9], [16] and [21] developed game theoretic methods for lane changing manoeuvres. In [16] for example, a mixed-motive game theory model is used for deciding the strategy made by two AVs equipped with Adaptive Cruise Control (ACC). Their simulation has shown that game theory provides better results as payoffs obtained are larger and the differences smaller for the two cars. Similar to our work, the model in [19] computed Nash equilibria using Fictitious Play. Their method differs from ours that not only their model takes into account pedestrians’ position from a single image but also some visual features from their appearance as part of the utility function to improve their trajectory prediction. [1] presents an algorithm for intersection management involving up to four self-driving cars communicating with each other. Two motions choices are available for each player (move forward or stop) and an optimised solution using game theory to solve the discrete intersection problem is presented. [2] makes use of game theory model such as the Prisoner’s Dilemma to propose a decision making system for AVs in a roundabout. Alternative variants of the game of chicken are proposed in [22], [24] and [7] to solve conflicts between agents at intersections. A cellular automata-based approach is implemented in [24] and [7] to represent the conflict between two agents. [22] focuses on the interaction between an AV and a pedestrian. [23] proposes a game theory approach for intersection conflicts management with reactive agents (the automated vehicles) equipped with Adaptive Cruise Control systems and a manager agent is used to decide the optimal strategy that increases the overall performance of all the agents. This approach prevents from a crash to occur and it also minimises the time delay in the intersection. [29] and [28] proposed a non-cooperative game theoretic approach to human collision avoidance. Their method differs from ours that they used a motion capture system to record human motions, a Bootstrap algorithm to compute the confidence intervals and applied a Dynamic Time Warping (DTW) algorithm to measure similarity between the trajectories. Gaussian process models of continuous trajectories have been applied to pedestrian path modelling and FRP in [27].

When multiple equilibria are present in games, standard game theory does not specify how the players should choose the best one. In the above studies, no method is proposed for players to select which equilibrium to use. Typically this is because Transport Studies seeks to describe macroscopic flows of traffic rather than prescribe actions for individual vehicles, and considers that any possible equilibrium is a good description of observed data. For example in [22] the choice for the best solution depends on ‘local social norms’ which assumes that drivers should have prior knowledge of local customs. Unusually, [10] proposed a novel approach for optimal strategy prescription, called meta-strategy convergence. This method begins by choosing an equal-weighted mixture of strategies from all rational equilibria (after removing dominated and asymmetric equilibria where possible). The resulting strategies do not in general form an equilibrium themselves, but by applying fictitious play until convergence, a single equilibrium is obtained upon which it is argued that two rational players should agree without communication. Most of the game theory models reviewed above outperform non-game theoretic predictive models [29], [7], [19], [23].

Pedestrian tracking plays an important role in many commercial applications. Several tracking approaches are described in [12] [11]. Tracking is still a challenge for computer vision systems because of the multiple uncertainties (e.g. occlusions) due to complex environments. Tracking of pedestrians requires the estimation of non-linear, non-Gaussian problems due to human motion, pedestrian scales, and posture changes. Monte-Carlo methods such as particle filtered-based approaches draw a set of samples assigned to a target and perform the data association for multiple targets using probabilistic techniques such as Nearest Neighbor (NN), Multi-Hypothesis-Tracking (MHT), JPDAF and PHD-filter [18]. Pedestrian tracking is composed of two steps: (i) a prediction step to determine the expected position and motion state and (ii) an update step to refine the prediction using sensor observations.

Tracking has been often combined with game theory for problems of multi-robot system coordination. [25] used the approach of non-cooperative games to control a team of mobile robots for a target tracking. When multiple equilibria are present, an arbiter module based on the min-max method is used to fairly distributes the costs among the robots. [17] applies cooperative game theory to improve tracking performance for a group of robots. Their method allows communication between the robots in order to minimise tracking costs and maximise the interests of the overall system of robots. [32] proposes a cooperative nonzero sum game approach for the problem of multi-target tracking for a multi-robot system in a dynamic environment.

II. METHODS

The present study shows how to fit parameters of the Sequential Chicken model to human behaviour collected from a semi-structured experimental environment. This environment is designed to enable the simplest possible mapping of physical human motions onto the model, as a step towards more naturalistic interaction modelling based on extensions of the model.

A. Sequential chicken model

In Sequential Chicken, two agents (e.g. pedestrian and/or human or autonomous driver) called Y and X are driving straight towards each other at an unmarked intersection as in fig. 1. In the model this process occurs over discrete space as in fig. 2 and discrete times (‘turns’) during which the agents can adjust their discrete speeds, simultaneously selecting speeds of either 1 square per turn or 2 squares per turn, at each turn. Both agents want to pass the intersection as soon as possible to avoid travel delays, but if they collide, they are both bigger losers as they both receive a negative utility $U_{crash}$. Otherwise if the players pass the intersection, each receives a time delay penalty $-TU_{time}$, where $T$ is the time from the start of the game and $U_{time}$ represents the value of saving one turn of travel time. The model assumes that the two players choose their actions (speeds) $\alpha_Y, \alpha_X \in \{1,2\}$ simultaneously then implement them simultaneously, at each of several discrete-time turns. There is no lateral motion (positioning within the lanes of the roads) or communication between the agents other than via their visible positions. The game is symmetric, as both players are assumed to know that they have the same utility functions.
(U_{\text{crash}}, U_{\text{time}})$, hence they both have the same optimal strategies. These optimal strategies are derivable from game theory together with meta-strategy convergence, via recursion. Sequential Chicken can be viewed as a sequence of one-shot sub-games, whose payoffs are the expected values of new games resulting from the actions, and are solvable by standard game theory.

The (discretized) locations of the players can be represented by $(y, x, t)$ at turn $t$ and their actions $a_Y, a_X \in \{1, 2\}$ for speed selection. The new state at turn $t + 1$ is given by $(y + a_Y x + a_X, t + 1)$. Define $v_{y,x,t} = (v_{y,x+1,t}, v_{y,x-1,t+1}, v_{y-1,x,t+1}, v_{y+1,x,t+1})$ as the value (expected utility, assuming all players play optimally) of the game for state $(y, x, t)$. As in standard game theory the value of each $2 \times 2$ payoff matrix can then be written as,

$$v_{y,x,t} = v(y-1,x-1,t+1) v(y-1,x-2,t+1)$$

which can be solved using dynamic programming assuming meta-strategy convergence equilibrium selection. Under some approximations based on the temporal gauge invariance described in [10], we may remove the dependencies on the time $t$ in our implementation so that only the locations $(y, x)$ are required in computation of $v_{y,x}$ and optimal strategy selection.

In the sequential chicken model, if the two players play optimally, then there must exist a non-zero probability for a collision to occur. Intuitively, if we consider an AV to be one player that always yields, it will make no progress as the other player will always take advantage over it, hence there must be some threats of collision.

### B. Human experiment

Eighteen human volunteer subjects (University of Lincoln Computer Science staff and students) were divided into 9 pairs, one designated as player $Y$ and the other as player $X$. Each pair was asked to play a physical version of the Sequential Chicken game on a plus-maze shaped playing area drawn on an indoor floor as 0.4m grid squares as shown in fig. 2. Player $Y$ was starting from $y = 10$ and player $X$ from $x = 10$ such that they were both starting 10 squares away from the intersection. Players were instructed that their objective was to pass the intersection as soon as possible, ‘as if they were trying to reach their office entrance in a busy pedestrian area’. Each pair played 5 games. In each game, the players were each given two cards containing the numbers 1 and 2. To prevent cheating, they were instructed that at each turn, called by the experimenter about every 2 seconds, they should select one card in secret, then both hold them up together (as in the game ‘scissors, paper, stone’) then both move together by that number of squares towards or beyond the intersection.

The players’ motions were recorded using a Velodyne 3D lidar while an experimenter called the turns. Fig. 5 shows examples of the lidar output during the games.
Kalman Filter (UKF) and Nearest Neighbour (NN) data association to deal with multiple detections simultaneously [3]. The tracker estimates the 2D coordinates and velocities of each pedestrian using a standard prediction-update recursive algorithm. The prediction step is based on the following constant velocity model,

\[
\begin{align*}
    x_k &= x_{k-1} + \Delta t \dot{x}_{k-1} \\
    \dot{x}_k &= \dot{x}_{k-1} \\
    y_k &= y_{k-1} + \Delta t \dot{y}_{k-1} \\
    \dot{y}_k &= \dot{y}_{k-1}
\end{align*}
\]  

where \( x_k \) and \( y_k \) are the Cartesian coordinates of the target at time \( t_k \), \( \dot{x}_k \) and \( \dot{y}_k \) are the respective velocities, and \( \Delta t = t_k - t_{k-1} \). (The variables \( x, y, t \) in this section measure the same quantities in the game theory model, but here take continuous values while the game theory model uses quantised values.) The update step of the estimation use a 2D polar observation model to represent the position of a detected cluster,

\[
\begin{align*}
    \phi_k &= \tan^{-1}(y_k/x_k) \\
    \chi_k &= \sqrt{x_k^2 + y_k^2}
\end{align*}
\]  

where \( \phi_k \) and \( \chi_k \) are, respectively, the bearing and the distance of the cluster’s centroid with respect to the sensor. For sake of simplicity, noises and coordinate transformations are omitted in the above equations. Tracks set were then filtered to locate the two longest tracks in the game grid area as shown in fig. 6. More details can be found in [3], [33]. Discrete player positions (grid squares) were then extracted from the tracks for each turn.

**D. Gaussian Process parameter posterior analysis**

We use Gaussian Processes Regression [31] to fit the posterior belief over the behavioural parameters of interest, \( \theta = (U_{\text{crash}}, U_{\text{time}}) \) from the observed data. D. Under the Sequential Chicken model, \( M \), these are,

\[
P(\theta|M,D) = \frac{P(D|\theta,M)P(\theta|M)}{\sum_{\theta'} P(D|\theta',M)P(\theta'|M)}. \tag{4}
\]

We assume a flat prior over \( \theta \) so that,

\[
P(\theta|M,D) \propto P(D|\theta,M), \tag{5}
\]

which is the data likelihood, given by,

\[
P(D|\theta,M) = \prod_{\text{game turn}} P(d_{\text{game turn}}^x|y,x,\theta,M')P(d_{\text{game turn}}^y|y,x,\theta,M'), \tag{6}
\]

where \( d_{\text{game turn}} \) are the observed action choices, and \( y \) and \( x \) are the observed player locations at each turn of each game. Here \( M' \) is a noisy version of the optimal Sequential Chicken model \( M \), which plays actions from \( M \) with probability \( (1-s) \) and maximum entropy random actions (0.5 probability of each speed) with probability \( s \). This modification is necessary to allow the model to fit data where human players have made deviations from optimal strategies which would otherwise occur in the data with probability zero. Real humans are unlikely to be perfectly optimal at anything as they may make mistakes of perception and decision making. This is a common method to weaken psychological models to allow non-zero probabilities for such mistakes if present.

For a given values of \( \theta \) we may compute the optimal strategy for the game by dynamic programming as in Algorithm 1. Optimal strategies are in general probabilistic, and prescribe the \( P(d_{\text{game turn}}^x|y,x,\theta,M),P(d_{\text{game turn}}^y|y,x,\theta,M) \) terms to compute the above data likelihood. We then use a Gaussian Process with a Radial Basis Function (RBF) kernel to smooth the likelihood function over all values of \( \theta \) beyond a sample whose values are computed explicitly. In practice this is performed in the log domain to avoid numerical computation problems with small probabilities. The resulting Gaussian Process is then read as the (un-normalized, log) posterior belief over the behavioural parameters \( \theta = (U_{\text{time}},U_{\text{crash}}) \) of interest.

**Algorithm 1 Optimal solution computation**

\[
\begin{align*}
    &\text{for } U_{\text{crash}} \text{ in range}(U_{\text{min \_ crash}}, U_{\text{max \_ crash}}) \text{ do} \text{ for } U_{\text{time}} \text{ in range}(U_{\text{min \_ time}}, U_{\text{max \_ time}}) \text{ do} \\
    &\quad S \leftarrow \text{strategy matrix \( NY \times NX \times 2 \) for} \ P(\text{player X chooses speed 2|y,x) do} \\
    &\quad \text{loglik} = 0 \text{ for each game in data do} \\
    &\quad \quad \text{loglik} - \prod_{\text{game turn}} P(d_{\text{game turn}}^x|y,x,\theta,M)P(d_{\text{game turn}}^y|y,x,\theta,M') \\
    &\quad \quad \text{end for} \\
    &\quad \text{end for} \\
    &\quad \text{maxloglik} \leftarrow \text{max of loglik(U_{\text{crash}}, U_{\text{time}})} \\
    &\text{end for} \\
    &\text{end for}
\end{align*}
\]

**III. Results**

After applying Gaussian Process Regression and optimising \( s \) to maximise the likelihood at the Maximum A Posteriori (MAP) point of \( \theta \), the posterior distribution over \( \theta = (U_{\text{crash}},U_{\text{time}}) \) is shown in fig. 7. The MAP estimate of the parameters is then around \( U_{\text{crash}} = -30, U_{\text{time}} = 45 \), at \( s = 0.11 \). The -2:3 ratio in the utilities means that assuming the noisy model \( M' \) the subjects value a 2/3 time delay equally to a crash, and the \( s \) value means that the subjects make mistakes from optimal behaviour in 11% of actions. Significance of the results can be seen by inspection of the thin

---

(Fig. 6: Pedestrian trajectories extracted from the 3D lidar before (a) and after (b) filtering)
standard deviation widths of 1D slices through the 2D posterior as in fig. 8.

Fig. 7: Gaussian Process log-posterior over behavioural parameters. (Un-normalized)

![Gaussian Process log-posterior over behavioural parameters.](image)

Fig. 8: Slices through the Gaussian Process showing 1 standard deviation log-posterior confidence.

The low (for Psychology models) deviation rate from optimal behavior, \(s = 11\%\), suggests that the model is a good fit to what human pedestrians actually do in priority negotiations. The behavioral parameter results then show that in the semi-structured scenario the participants have a preference for time saving rather than collision avoidance. This was unexpected – in real life, a collision is intuitively much worse than almost any time delay to almost everyone. While the game was structured as a sequence of discrete turns to simplify model fitting, it was designed to closely resemble a real-world interaction in continuous time as much as possible. This high-risk appetite of the pedestrians is perhaps best explained by: (1) the high safety of the environment – the players know the study is set up in a laboratory environment operating under health and safety policies, and that all the other subjects are also just playing a game, so they are less concerned about colliding than they would be with strangers in a real public place such as an office corridor intersection; (2) the environment of the experiment may lead some players to view it as a zero-sum competition (a race) rather than attempting to maximise only their own utility; and (3) the utility of colliding with other experimental subjects is less bad than colliding with real strangers or with robots or autonomous vehicles, which is also harder to emulate in a safe laboratory environment that simulates pedestrian-pedestrian interactions.

IV. CONCLUSION

Despite showing an unexpected and unrealistic result as a consequence of the lab environment used, this Methods Paper has demonstrated successful use of a new Method for elucidating pedestrian preferences in the Sequential Chicken model from a real-world scenario and empirical data. It was conducted in a deliberately simplified, semi-structured environment, designed to simplify and test model-fitting whilst still demonstrating all the essential components required for future more realistic experiments: protocol, tracking, parameter fitting, and posterior parameter analysis. Building on the components from this protocol demonstration, future experiments could now take further steps towards elucidation for realistic environments, including replacing the semi-structured discrete turn-taking with unstructured, continuous time and space movements but also using signalling methods (gestures, sounds etc.). They should also move away from the simplifying assumption of shared known and symmetric utility functions, for example by using augmented reality to safely simulate interactions with heavy vehicles (e.g. in 3D driving simulations) whose damage in a collision is less to themselves than to the pedestrian. This current model would not be affected by conflicts related to driving conventions (such as left- and right-hand driving) as only one single driving convention can run at a time in a precise location. The proposed approach could be easily extended to multi-lane roads and it would become a multi-target tracking problem for the AV. Future work should consider less computational optimization methods than the gaussian process (GP) regression for the real-time behavioural parameter fitting.

REFERENCES

[1] R. Adkins. Autonomous vehicle intersection management. Technical report, 2016.
[2] L. Banjanovic-Mehmedovic, E. Halilovic, I. Bosankic, M. Kantardzic, and S. Kasapovic. Autonomous vehicle-to-vehicle (v2v) decision making in roundabout using game theory? International Journal of Advanced Computer Science and Applications (ijasca), 7(8), 2016.
[3] N. Bellotto and H. Hu. Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of Bayesian filters. Autonomous Robots, 28:425–438, 2010.
[4] R. Brooks. The big problem with self-driving cars is people. In IEEE Spectrum, 27 Jul, 2017., 2017.
[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. IEEE Transactions on Robotics, 6(6), 2016.
[6] F. Camara, R. Romano, G. Markkula, R. Madigan, N. Merat, and C. W. Fox. Empirical game theory of pedestrian interaction for autonomous vehicles. In Measuring Behavior 2018: 11th International Conference on Methods and Techniques in Behavioral Research, Manchester Metropolitan University, March 2018.
[7] P. Chen, C. Wu, and S. Zhu. Interaction between vehicles and pedestrians at uncontrolled mid-block crosswalks. Safety Science, 82:68 – 76, 2016.
[8] J. Condliffe. Humans will bully mild-mannered autonomous cars, 2016.
[9] R. Elvik. A review of game-theoretic models of road user behaviour. Accident Analysis & Prevention, 62:388 – 396, 2014.
[10] C. W. Fox, F. Camara, G. Markkula, R. Romano, R. Madigan, and N. Merat. When should the chicken cross the road?: Game theory for autonomous vehicle - human interactions. In Proceedings of VEHITS 2018: 4th International Conference on Vehicle Technology and Intelligent Transport Systems, January 2018.
[11] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(7):1239–1258, July 2010.
[12] L. Guo, L. Li, Y. Zhao, and Z. Zhao. Pedestrian tracking based on camshift with kalman prediction for autonomous vehicles. Transactions on Robotics, 6(6), 2016.
[13] K. Hurley. Autonomous vehicles the “game of chicken” which could be a serious problem for driverless cars., 2017.
[14] InterACT. Eu h2020 interact project, 2018.
[15] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada. An open approach to autonomous vehicles. IEEE Micro, 35(6):60–68, Nov 2015.
[16] R. Kim, Changwon & Langari. Game theory based autonomous vehicles operation. International Journal of Vehicle Design (IJVD), Vol. 65, 2014.
[17] Y. Li, M. Li, L. Dou, Q. Zhao, Z. Wang, and J. Li. A socially multi-robot target tracking method based on extended cooperative game theory. In 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), pages 247–252, Dec 2014.

[18] W. Luo, X. Zhao, and T. Kim. Multiple object tracking: A review. CoRR, abs/1409.7618, 2014.

[19] W. Ma, D. Huang, N. Lee, and K. M. Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. CoRR, abs/1604.01431, 2016.

[20] R. Madigan, T. Louw, M. Dziennus, T. Graindorge, E. Ortega, M. Graindorge, and N. Merat. Acceptance of automated road transport systems (arts): An adaptation of the utaut model. Transportation Research Procedia, 14:2217 – 2226, 2016. Transport Research Arena TRA2016.

[21] F. Meng, J. Su, C. Liu, and W. H. Chen. Dynamic decision making in lane change: Game theory with receding horizon. In 2016 UKACC 11th International Conference on Control (CONTROL), pages 1–6, Aug 2016.

[22] A. Millard-Ball. Pedestrians, autonomous vehicles, and cities. Journal of Planning Education and Research, 38(1):6–12, 2018.

[23] H. A. Rakha, I. Zohdy, and R. K. Kamalanathsharma. Agent-based game theory modeling for driverless vehicles at intersections. Technical report, 2013.

[24] A. Rane, S. Krishnan, and S. Waman. Conflict resolution of autonomous cars using game theory and cellular automata. In 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), pages 326–330, Feb 2014.

[25] K. Skrzypczyk. Game theory based target following by a team of robots. In Proceedings of the Fourth International Workshop on Robot Motion and Control (IEEE Cat. No.04EX891), pages 91–96, June 2004.

[26] S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics. Intelligent robotics and autonomous agents. MIT Press, 2005.

[27] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 797–803, Oct 2010.

[28] A. Turnwald, D. Althoff, D. Wollherr, and M. Buss. Understanding human avoidance behavior: Interaction-aware decision making based on game theory. International Journal of Social Robotics, 8(2):331–351, Apr 2016.

[29] A. Turnwald, W. Olszowy, D. Wollherr, and M. Buss. Interactive navigation of humans from a game theoretic perspective. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 703–708, Sept 2014.

[30] Udacity. Open source self-driving car, 2017.

[31] C. K. Williams and C. E. Rasmussen. Gaussian processes for regression. In Advances in neural information processing systems, pages 514–520, 1996.

[32] M. Yan. Multi-robot searching using game-theory based approach. International Journal of Advanced Robotic Systems, 5(4):44, 2008.

[33] Z. Yan, T. Duckett, and N. Bellotto. Online learning for human classification in 3d lidar-based tracking. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pages 864–871, 2017.
Automatically Learning Driver Behaviors for Safe Autonomous Vehicle Navigation

Ernest Cheung\textsuperscript{1}, Aniket Bera\textsuperscript{1}, Emily Kubin\textsuperscript{2}, Kurt Gray\textsuperscript{2}, and Dinesh Manocha\textsuperscript{1}

Abstract—We present an autonomous driving planning algorithm that takes into account neighboring drivers’ behaviors and achieves safer and more efficient navigation. Our approach leverages the advantages of a data-driven mapping that is used to characterize the behavior of other drivers on the road. Our formulation also takes into account pedestrians and cyclists and uses psychology-based models to perform safe navigation. We demonstrate our benefits over previous methods: safer behavior in avoiding dangerous neighboring drivers, pedestrians and cyclists, and efficient navigation around careful drivers.

I. INTRODUCTION

There are different kinds of drivers in urban environments, and an expert human driver will identify dangerous drivers and avoid them accordingly. However, existing autonomous driving systems often treat all neighboring vehicles the same and do not take actions to avoid the dangerous drivers. This problem has been studied in transportation and urban planning works [31]. This line of works map drivers’ behaviors with background information like age, gender, driving history, etc., but this information is not available to autonomous vehicles. Therefore, to allow autonomous driving algorithms to account for driving behaviors, a mapping between sensor data and driving behaviors must be available.

Previous studies in transportation and urban studies [16], [31] usually study the difference between aggressive drivers, careful drivers and typical drivers. In particular, Guy et al. [20] and Bera et al. [4], [5], [3] applied psychological theory to capture human behaviors. Autonomous driving systems that are on the roads right now uses a range of different algorithms to interpret the sensor data: trajectory data computation using semantic understanding or object detection methods [18]. Some uses an end-to-end approach to compute driving actions directly from sensor data[8].

Main Results: Our approach takes into account behaviors of neighboring entities and plans accordingly to perform safer navigation. We leverage the results of an extensive user study that learned the relationship between vehicular trajectories and the underlying driving behaviors: Trajectory to Driver Behavior Mapping [12]. This work allows our navigation algorithms to classify the driving behaviors of neighboring drivers, and we demonstrated simulated scenarios with vehicles, pedestrians, and cyclist where navigation with our approach is safer.

Compared to prior algorithms, our algorithm offers the following benefits:

1. Driving Behavior Computation: Trajectory to Driver Behavior Mapping established a mapping between five features and six different driving behaviors, and conducted factor analysis on the six behaviors, which are derived from two commonly studied behaviors: aggressiveness and carefulness. The results show that there exists a latent variable that can summarize these driving behaviors and that can be used to measure the level of awareness that one should have when driving next to another vehicle. The same study examined how much attention a human pays to such a vehicle when it is driving in different relative locations. We leverage the results of this study and develop a proximity cost that reacts to aggressive drivers more appropriately.

2. Improved Realtime Navigation: We enhance an existing Autonomous Driving Algorithm [7] to navigate according to the neighboring drivers’ behaviors. Our navigation algorithm identifies potentially dangerous drivers in real-time and chooses a path that avoids potentially dangerous drivers. In particular, our approach accounts for pedestrians and cyclists, and avoids them by considering their velocity relative to the ego-vehicle. Our method can offer saver navigation and plan more appropriately to avoid dangerous drivers than prior works. We refer the readers to read [11] for the technical details.

An overview of our approach is shown in Figure\textsuperscript{1}. The rest of the paper is organized as follows. We present a detailed overview of previous work in Section II. We describe the mapping from trajectories to driving behaviors in Section III and our autonomous driving algorithm in Section IV.

II. RELATED WORKS

A. Driving Behaviors Studies

Psychology, transportation, and urban planning researchers have been studying human driving behaviors. Aljaafreh et al. [1] classified drivers into four different levels of aggressiveness with accelerometer data. Feng et al. [16] categorized drivers into three different level of aggressiveness according to drivers’ background information (age, gender, experience, etc.), and environmental factors (weather, traffic, etc.). Apart from that, social psychologist have also studied the correlation between driver background information and driving behaviors [29], [2], and previous driving behaviors [9]. Besides, Meiring et al. [31] pointed out that careless drivers, including drunk and distracted drivers, are also dangerous. Despite the fact that these works have found mappings between driving behaviors and a lot of other different factors, most of these factors are unknown to autonomous vehicles during navigation. We use neighboring vehicles’ trajectories,
discovered that human drivers’ behaviors can be affected when they observe an autonomous vehicle and that they will react in certain ways when they observe different actions of the autonomous vehicle [37]. Huang et al. [24] proposed a technique to make autonomous car actions more easily understand by humans, so that their reactions are more predictable. Besides, an active learning approach [14] using examples of expert human driver’s preferences has been to model human driving behaviors. These works show the importance of having autonomous vehicles navigating according to human behaviors.

C. Autonomous Car Navigation

There is a significant number of works on navigating autonomous vehicles [25], [38], [44], [28], [23], [42]. During the DAPRA Urban Grand Challenge and the Grand Cooperative Driving Challenge, the participating research teams proposed different navigation approaches [10], [17], [27], [15]. Recently, Best et al. [7] proposed a novel navigation algorithm, AutonoVi, which also considers steering and acceleration planning, dynamic lane changes, and several other scenarios. We proposed a new approach that takes into account driving behavior, which is complimentary to these previous work and can be combined with them.

III. TRAJECTORY TO DRIVER BEHAVIOR MAPPING

In this section, we describe the trajectory features that are used to identify driver behaviors, the driving behavior metrics, and the attention metrics used in a detailed user study, Trajectory to Driving Behavior Mapping [12].

A. Features

The goal of Trajectory to Driving Behavior Mapping is to leverage a set of trajectory features that map to driving behaviors, assuming that the trajectories have already been extracted from the raw sensor data. As described in the previous section, a lot of features (e.g., drivers’ backgrounds, throttle opening, environmental factors, etc.) that have been mapped to driving behavior are not available for autonomous vehicles. Therefore, the user study has derived a set of variants and performed feature selection to select the most relevant ones to use in the mapping.

| Notation | Description |
|----------|-------------|
| \( v_{rel} \) | Relative speed to neighbors |
| \( d_{avg} \) | Distance with front car |
| \( s_{front} \) | Lane following metric |
| \( j_c \) | Longitudinal jerk |
| \( s_{center} \) | Lane following metric |

TABLE I: Five Features selected in Trajectory to Driving Behavior Mapping

1) Acceleration: Previous works [33], [32], [39], [41] have shown that acceleration can be used to identify driver aggressiveness. This study [33] found out that longitudinal jerk can reflect aggressiveness better than progressive jerk, and this has been further verified during the feature selection in the user study.
2) Lane following: The metric proposed in this work [6] measures the extent of lane following using the mean and standard deviation of lane drifting and lane weaving. Trajectory to Driving behavior proposes a feature that also depends on lane drifting, but further differentiates drivers who keep deviating from the center of the lane to the left and right, and those drivers who are driving stably off the center of the lane. Furthermore, when a vehicle is performing lane changing, the effect on this metric of these trajectory segments is nullified and will not impact this metric.

Let \( y_i \) and \( y(t) \) be the center longitudinal position of the lane in which the targeted car is in and the longitudinal position of the car at time \( t \), respectively. Also suppose a set of lane changing events happened at time \( t_i \), \( C = \{ t_1, t_2, \ldots, t_n \} \), the lane drift metric \( s_C(t) \) is given by:

\[
s_C(t) = \begin{cases} 
0, & \text{if } \exists t \in C \text{ s.t. } t \in [t - k, t + k], \\
(y(t) - y_i), & \text{otherwise}. 
\end{cases}
\]

where \( k \) is the amount of time that we nullify the impact of lane changing to this metric.

Trajectory to Driving Behavior Mapping measures the rate of change in drifting in \( \tau \) seconds, so that this metric can highlight those drivers who are drifting more frequently from the center of the lane. The overall lane following metric is therefore defined as below. It is also illustrated in Figure 2.

\[
s_{\text{center}} = \int |s_C(t)| \left[ \mu + \int_{t-\tau}^{t} |s'_C(t)| dt \right] dt,
\]

where \( \mu \) is a parameter that differentiates drivers who are driving stably off the center of the lane, and those who are driving along the center of the lane.

![Fig. 2: Lane following metric illustration. The lane following metric, \( s_{\text{center}} \), is given by the sum of the area under the plot \( s'_C \). The example shows that the lane following metric can differentiate drivers from drifting left and right (i iii), driving along the center of the lane (ii), changing lanes (iv), and consistently driving off the center of the lane (v).](image)

3) Relative Speed: Trajectory to Driving Behavior Mapping designed the following metric to capture the relationship between a given driving behavior and the relative speed of the car with respect to neighboring cars:

\[
v_{\text{nei}} = \int \sum_{n \in N} \max(0, \frac{v(t) - v_n(t)}{\text{dist}(x(t), x_n(t))}) dt,
\]

where \( N \) is the set containing all neighboring cars within a reasonably huge range. \( v(t), x(t), v_n(t), x_n(t) \) are the speed and the position of the targeting car, and the position and the speed of the neighbor \( n \), respectively.

This metric relies merely on the speed and position of the neighbors, and it can represent the actual driving speed of the targeted vehicle with respect to it’s neighbor better than simply using relative speed.

B. Driving Behavior Metrics and Attention Metrics

Aggressiveness [16], [1], [22] and Carefulness [31], [35], [30] are two metrics that are commonly used to identify dangerous drivers. In typical social psychology studies, related items are introduced into user evaluation to ensure the robustness of the results. Therefore, Trajectory to Driving Behavior mapping evaluated four more driving behaviors apart from Aggressiveness and Carefulness, and those are listed in Table II.

When an aggressive or careless driver is observed, depending on the position of that driver with respect to the targeted vehicle, the amount of attention that the driver of the targeted vehicle pays would still vary. Therefore, when evaluating the users’ responses when driving as the targeted vehicle, the users are also asked to rate the four attention metrics listed in Table II.

| Symbol | Description | Symbol | Level of Attention when |
|--------|-------------|--------|------------------------|
| \( b_0 \) | Aggressive | \( b_6 \) | following the target |
| \( b_1 \) | Reckless | \( b_7 \) | preceding the target |
| \( b_2 \) | Threatening | \( b_8 \) | driving next to the target |
| \( b_3 \) | Careful | \( b_9 \) | far from the target |
| \( b_4 \) | Cautious | | |
| \( b_5 \) | Timid | | |

**TABLE II: Six Driving Behavior metrics (\( b_0, b_1, \ldots, b_5 \)) and Four Attention metrics (\( b_6, b_7, b_8, b_9 \)) used in user evaluation in obtaining the mapping**

C. Data-Driven Mapping

Trajectory to Driving Behavior Mapping conducts a user study that has 100 participants identifying driver behaviors from videos. The trajectories of the videos are extracted from the Interstate 80 Freeway Dataset [21]. The users were asked to rate the metrics we listed in Table II on a 7-point scale and a 5-point scale for driving behavior and attention metrics, respectively.

After that, feature selection was applied to the results using least absolute shrinkage and selection operator (Lasso) analysis. In addition, the five features that are most appropriate for mapping to driving behaviors are extracted from ten potential ones. It concluded that using \( \{s_{\text{center}}, v_{\text{nei}}, s_{\text{front}}, v_{\text{avg}}, \hat{t}\} \) in mapping between features and driving behavior, and \( \{s_{\text{center}}, v_{\text{nei}}, v_{\text{avg}}\} \) in the mapping between features and attention metrics can produce best regression models.

Using \( \{s_{\text{center}}, v_{\text{nei}}, s_{\text{front}}, v_{\text{avg}}, \hat{t}\} \) and \( \{s_{\text{center}}, v_{\text{nei}}, v_{\text{avg}}\} \) as the features, linear regression is applied to obtain the mapping between these selected
features and the drivers’ behaviors. The results we obtained are below. For $B_{\text{behavior}} = [b_0, b_1, ..., b_5]^T$,

$$
B_{\text{behavior}} = \begin{pmatrix}
1.63 & 4.04 & -0.46 & -0.82 & 0.88 & -2.58 \\
1.58 & 3.08 & -0.45 & 0.02 & -0.10 & -1.67 \\
1.35 & 4.08 & -0.58 & -0.43 & -0.28 & -1.99 \\
-1.51 & -3.17 & 1.06 & 0.51 & -0.51 & 1.39 \\
-2.47 & -2.60 & 1.43 & 0.98 & -0.82 & 1.27 \\
-3.59 & -2.19 & 1.75 & 1.73 & -0.30 & 0.61
\end{pmatrix} \mathbf{s}_{\text{center}} \begin{pmatrix}
\mathbf{s}_{\text{max}} \\
\mathbf{s}_{\text{front}} \\
\mathbf{s}_{\text{neg}} \\
\mathbf{s}_{\text{avg}} \\
\mathbf{j}_i \\
1
\end{pmatrix} \tag{4}
$$

Moreover, for $B_{\text{attention}} = [b_6, b_7, b_8, b_9]^T$,

$$
B_{\text{attention}} = \begin{pmatrix}
B_{\text{back}} \\
B_{\text{front}} \\
B_{\text{adj}} \\
B_{\text{far}}
\end{pmatrix} = \begin{pmatrix}
0.54 & 1.60 & 0.11 & -0.8 \\
-0.73 & 1.06 & 0.63 & -0.07 \\
-0.14 & 1.73 & 0.25 & 0.15 \\
0.25 & 1.47 & 0.17 & -1.43
\end{pmatrix} \mathbf{s}_{\text{center}} \begin{pmatrix}
\mathbf{v}_{\text{neg}} \\
\mathbf{v}_{\text{avg}} \\
\mathbf{v}_{\text{ego}} \\
1
\end{pmatrix} \tag{5}
$$

We refer the readers to read [11] for more technical details and analysis.

IV. Navigation

In this section, we describe how we leverage the benefits of identifying driver behaviors and ensure safe navigation. TDBM [12] extends an autonomous car navigation algorithm, AutonoVi [7], and shows improvements in its performance by using our driver behavior identification algorithm and TDBM. AutonoVi is based on a data-driven vehicle dynamics model and optimization-based maneuver planning, which generates a set of favorable trajectories from among a set of possible candidates, and performs selection among this set of trajectories using optimization. It can handle dynamic lane-changes and different traffic conditions.

The approach used in AutonoVi is summarized below: The algorithm establishes a graph of roads from a GIS database and computes the shortest global route plan using A* algorithm. Taking into account traffic rules and real-time traffic, the plan is translated to a static guiding path, which consists of a set of $C^1$ continuous way-points. AutonoVi then samples the speed and steering angle in a favourable range of values to obtain a set of candidate trajectories. Using the Control Obstacles approach, AutonoVi eliminates the trajectories that would lead to a possible collision. With the set of collision-free trajectories, AutonoVi selects the best trajectory using an optimization approach. It selects trajectories that avoid: i) deviating from the global route; ii) unnecessary lane changes; ii) sharp turns, breaking, and acceleration, which lead to discomforting experiences for passengers; and iv) getting to close to other road entities (including vehicles, pedestrians, and cyclists).

A. Neighboing Vehicles

AutonoVi proposed a proximity cost function to differentiate entities by class to avoid getting too close to other objects. It considers all vehicles as the same and applies the same penalization factor, $F_{\text{vehicle}}$, to them. Similarly, it applies higher factors : $F_{\text{ped}}$ and $F_{\text{cyc}}$ to all pedestrians and all cyclists, respectively. The original proximity cost used in AutonoVi is:

$$
c_{\text{prox}} = \sum_{n=1}^{N} F_{\text{vehicle}} e^{-d(n)} \tag{6}
$$

This cost function has two issues: i) it cannot distinguish dangerous drivers to avoid driving too close to them, and ii) it diminishes too rapidly due to its use of an exponential function. Therefore, TDBM proposed a novel proximity cost that can solve these problems:

$$
c'_n = \sum_{n=1}^{N} c(n) \tag{7}
$$

where $d(n)$ is the distance between the car navigating with TDBM and the neighbor $n$; $d_t$ is a threshold distance beyond which neighbors will be applied with the ‘far away’ metric $B_{\text{far}}$; and $d_{t2}$ is a threshold distance beyond which neighbors would not have any impact on TDBM’s navigation. $B_{\text{far}}$ and $B_r$ refers to the attention metrics in Equation $B$.

This proximity cost used in TDBM discouraged the optimizer from picking any candidate whose path is close to these dangerous drivers. However, this approach has a drawback: when the ego-vehicle and the neighboring vehicle are both slow, some unnecessary lane changing may occur. To avoid this, we add the relative velocity of the neighboring vehicle in relation to the ego-vehicle into the cost function. The new cost function also nullifies the effect of the cost on vehicles that are driving away from the ego-vehicle. The new cost function for vehicles is:

$$
c'_n = \sum_{n=1}^{N} \max(0, v_{\text{ego}} - v_n) c(n) \tag{9}
$$

where $v_{\text{ego}}$ and $v_n$ are the current progression speed along the lane of the ego-vehicle and the neighbor $n$ respectively.

B. Pedestrians and Cyclists

The proximity costs for pedestrians and cyclists in AutonoVi and TDBM are still diminishing rapidly and do not take into consideration the velocity of the pedestrian/cyclist. We propose accounting for the current velocity in order to better predict and represent the zones to be avoided by the navigation algorithm:

$$
c'_{\text{obs}} = \sum_{n=1}^{N} F(n) \max(0, v_n - \frac{\mathbf{s}_{\text{ego}} - \mathbf{s}_n}{||\mathbf{s}_{\text{ego}} - \mathbf{s}_n||}) \tag{10}
$$
where $F(n)$ returns $F_{ped}$ or $F_{cyc}$ depending on the type of obstacle $n$. $v_n$ represents the current normalized velocity of the pedestrian/cyclist. $\tilde{s}_{ego}$ and $\tilde{s}_n$ are the position of the ego-vehicle and the obstacle $n$, respectively.

Using these new cost functions, we can avoid drivers that are potentially riskier, stay away from pedestrians and cyclists more appropriately, and select a better navigation path. Examples of scenarios are illustrated in Figure 3.

V. Conclusion and future works

We present a new navigation approach leveraging the estimation of neighboring human drivers’ behaviors and react to them accordingly. Using our approach, the navigation algorithm can more accurately estimate the level of awareness the ego-vehicle should have about neighboring vehicles, pedestrians and cyclists, and more effectively avoid those that require a higher level of awareness. Our approach can provide safer navigation among aggressive drivers, pedestrians, and cyclist and more efficient navigation when facing careful drivers.

The trajectory data that is currently available in the autonomous driving research community are limited, as labeling raw images are expensive. Currently, pedestrian and vehicle detection methods are advancing, and soon will be able to extract trajectory data reliably from raw data. The Trajectory to Driving Behavior Mapping applied in this work is based on highways, and the driving behaviors could be different in urban environment as there are pedestrians and cyclists involved. Furthermore, driving and pedestrians behaviors are different across countries and regions. With more data available, we would like to evaluate our approach on urban environments. Besides, there are works conducted to predict pedestrians trajectories (e.g., SocioSense [5]), and we can combine them to navigate even safer around pedestrians and cyclists in the future.

REFERENCES

[1] A. Aljaafreh, N. Alshabatat, and M. S. N. Al-Din. Driving style recognition using fuzzy logic. In Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on, pages 460–463.

[2] K. H. Beck, B. Ali, and S. B. Daughters. Distress tolerance as a predictor of risky and aggressive driving. Traffic injury prevention, 15(4):349–354, 2014.

[3] A. Bera, T. Randhavane, E. Kubin, A. Wang, K. Gray, and D. Manocha. Classifying group emotions for socially-aware autonomous vehicle navigation. 2018.

[4] A. Bera, T. Randhavane, and D. Manocha. Aggressive, tense, or shy? identifying personality traits from crowd videos. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017.

[5] A. Bera, T. Randhavane, R. Prinja, and D. Manocha. Sociosense: Robot navigation amongst pedestrians with social and psychological constraints. In Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on, pages 7018–7025.

[6] L. M. Bergasa, D. Alméría, J. Almazán, I. J. Yebs, and R. Arroyo. Drivesafe: An app for alerting inattentive drivers and scoring driving behaviors. In Intelligent Vehicles Symposium Proceedings, 2014 IEEE, pages 240–245. IEEE, 2014.

[7] A. Best, S. Narang, L. Pasqualin, D. Barber, and D. Manocha. Autonovi: Autonomious vehicle planning with dynamic maneuvers and traffic constraints. arXiv:1703.08561, 2017.

[8] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp,
R. Kianfar, B. Augusto, A. Ebadighajari, U. Hakeem, J. Nilsson, A. Kesting, M. Treiber, and D. Helbing. General lane-changing model

S. H. Huang, D. Held, P. Abbeel, and A. D. Dragan. Enabling robots

P. B. Harris, J. M. Houston, J. A. Vazquez, J. A. Smither, A. Harms, J. Halkia and J. Colyar. Interstate 80 freeway dataset.

S. J. Guy, S. Kim, M. C. Lin, and D. Manocha. Simulating heterogeneous crowd behaviors using personality trait theory. In Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation, pages 43–52, 2011.

J. Halkia and J. Colyar. Interstate 80 freeway dataset. Federal Highway Administration, U.S. Department of Transportation, 2006.

P. B. Harris, J. M. Houston, J. A. Vazquez, J. A. Smither, A. Harms, J. A. Dahlke, and D. A. Sachau. The prosocial and aggressive driving inventory (padi): A self-report measure of safe and unsafe driving behaviors. Accident Analysis & Prevention, 72(Supplement C):1 – 8, 2014.

G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In American Control Conference ACC’07, pages 2296–2301. IEEE, 2007.

S. H. Huang, D. Held, P. Abbeel, and A. D. Dragan. Enabling robots to communicate their objectives. arXiv:1702.03465, 2017.

C. Katrakazas, M. Qud dus, W.-H. Chen, and L. Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. Transportation Research Part C: Emerging Technologies, 60:416–442, 2015.

A. Kesting, M. Treiber, and D. Helbing. General lane-changing model mobility for car-following models. Transportation Research Record: Journal of the Transportation Research Board, (1999):86–94, 2007.

R. Kianfar, B. Augusto, A. Ebadighajari, U. Hakeem, J. Nilsson, A. Raza, R. S. Tabar, N. V. Inukulapat, C. Englund, P. Falcone, et al. Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge. IEEE transactions on intelligent transportation systems, 13(3):994–1007, 2012.