Second Language Tutoring using Social Robots

Project No. 688014

L2TOR
Second Language Tutoring using Social Robots

Grant Agreement Type: Collaborative Project
Grant Agreement Number: 688014

D4.1 Input Module for Number Domain

Due Date: **30/06/2017**
Submission Date: **05/05/2016**

Start date of project: **01/01/2016** Duration: **36 months**

Organisation name of lead contractor for this deliverable: **Plymouth University**

Responsible Person: **Tony Belpaeme** Revision: **1.0**

Project co-funded by the European Commission within the H2020 Framework Programme		
Dissemination Level		
PU	Public	**PU**
PP	Restricted to other programme participants (including the Commission Service)	
RE	Restricted to a group specified by the consortium (including the Commission Service)	
CO	Confidential, only for members of the consortium (including the Commission Service)	

Contents

Executive Summary

This document outlines the current implementation and evaluations leading to the input module for the L2TOR number domain. Specifically, evaluations of Automatic Speech Recognition (ASR), face detection, face recognition, and object recognition in the context of child-robot interaction are described. The implementation of components for the L2TOR system is derived from these evaluations and will be detailed here.

Principal Contributors

The main authors of this deliverable are as follows (in alphabetical order):

Tony Belpaeme Plymouth University
James Kennedy, Plymouth University
Fotios Papadopoulos, Plymouth University
Christopher Wallbridge, Plymouth University

Revision History

Version 0.1 (J.K. 24-01-2017)
First draft of structure.
Version 1.0 (T.B. 30-06-2017)
Final version.

1 Overview of the Number Domain Input

Work Package 4, multimodal input processing, aims to leverage existing software and methods for social signal processing. The solutions devised as part of this work package will provide the input from sensing data to enable the lessons for the L2TOR evaluations. The intention is that wherever possible, the input module should utilise current state-of-the-art software following evaluation of its suitability for the specific scenarios in L2TOR. This approach encourages efficient use of resources, whilst simultaneously providing substantial information for use by the robotic platform from the world (and specifically, interacting partners) around it.

The work in this deliverable describes the research undertaken in the design and development of the software prototype for the multimodal input interpretation for the number domain (Milestone 7). The specific input modalities considered here are **speech** (Section 2), in the form of automatic speech recognition (ASR) and voice activity detection (VAD), and **vision**, in the form of face detection and face recognition (Section 3), and object tracking (Section 4). These modalities are included in Tasks 4.1, 4.2 and 4.6 from the Description of Action. Some of this work additionally serves as preliminary investigation for, or overlaps with the requirements of, D4.2 which is concerned with the spatial domain; these aspects will not be discussed here, but in D4.2, due in M27.

The software prototype associated with this deliverable comes in the form of a series of components (available from the L2TOR Git repository). Section 5 describes the implementation of these components, and also provides a brief overview of how these components fit into the main L2TOR system that will be used in the number domain evaluations.

2 Speech Input

Speech is manifestly key in language learning, an application reliant upon verbal channels of communication. This is particularly apposite when interacting with children as young as those under consideration in the L2TOR project (aged 3-5 years old), who have limited reading and writing capabilities. This includes not only appropriate speech production by robots, but detecting, transcribing and understanding speech from young users as well. A prerequisite to this interpretation of speech is having a sufficiently accurate transcription of what is being said. For this reason, high-quality Automatic Speech Recognition (ASR) is a vital component for producing autonomous human-robot interaction; this is discussed in Section 2.1. It is also useful to know when a child is speaking. This would enable a robot to generate autonomous behaviour contingent on that of the child, e.g., the robot could focus its attention towards the child at appropriate moments in the interaction. Voice Activity Detection (VAD) in the context of child speech is therefore discussed in Section 2.2.

2.1 Automatic Speech Recognition

ASR engines have undergone significant improvements in recent years, particularly following the introduction of new techniques such as deep learning [1]. However, these engines are commonly evaluated against standardised datasets of adult speech [2]. One might naively assume that these improvements will also translate to child speech, and will cope relatively well with noisy (i.e., real world) environments, such as those experienced in applied HRI. However, this is often observed to not be the case, cf. [3].

As part of the research undertaken for T4.1 for the L2TOR project, we systematically evaluate a number of state-of-the-art Automatic Speech Recognition engines under a variety of environmental conditions. Using a combination of restricted and free speech from 11 children, we explore the impact

of background noise, microphone quality, microphone placement, and providing a *grammar* to ASR engines. The findings in full can be seen in [4]; included as Annex 6.1 here. A summary of the outcomes from this evaluation is reproduced below:

- Constrain the interaction by leading the child to a limited set of responses. This typically works well for older children, but carries the risk of making the interaction stale.
- Use additional input/output devices. A touchscreen has been found to be a particularly effective substitute for linguistic input [5], but also other devices –such as haptic devices– should be considered.
- Place the young user in the optimal location for ASR. The location and orientation relative to the microphone (and robot) has a profound impact on ASR performance. A cushion, stool or chair can help children sit in the optimal location.
- Constrain the grammar of the ASR. While not all ASR engines allow for this, some will allow constraints or "hints" on what is recognised. This proves to be valuable in constrained interaction settings, for example, when listening only for numbers between 1 and 10.
- Background noise appears to be less of an issue than initially anticipated. It appears that the current ASR engines have effective noise cancelling mechanisms in place. Nevertheless, "the less noise, the better" remains true, particularly when interacting at a distance from the robot.
- A lack of ASR performance does not mean that the robot should not produce speech, as speech has been found to be particularly effective to engage children.

To summarise, ASR performance is not currently sufficient to understand open speech from children. The robot will need to direct the child towards answers with a small number of responses to maximise the likelihood of accurate recognition. Using an external microphone and a cloud-based recogniser can lead to improvements in recognition when using a restricted grammar, however, this adds potentially impractical requirements for interaction scenarios (such as placing large and expensive, studio-grade microphones beside the robot, with a reliable internet connection). Simply using a small off-board microphone does not provide a significant improvement above the built-in robot microphone, so the built-in microphone should be preferred for speech recognition tasks. Nevertheless, numbers from 1 to 10 are only accurately recognised 61% of the time (95% CI [48%,73%]). Possible responses must be restricted to a smaller set of options, or alternatives to speech input must be offered, such as touchscreen input.

These evaluations were conducted in English, which is likely to have the largest amount of raw data for training speech recognisers. Other languages used in the project, such as Dutch and Turkish will likely experience even lower recognition rates. Through small-scale pilots to verify whether this was the case, we see that the performance for Dutch numbers from 1 to 10 spoken by a native child achieves 20% recognition using the built-in NAO recogniser, whilst Turkish is not currently supported as a language. For off-board microphones and cloud-based ASR engines, results for Turkish and Dutch were similar to that of English in number word (translations of 1 through 10) recognition.

2.2 Voice Activity Detection

Voice Activity Detection (VAD) separates speech segments from non-speech segments in an audio signal by using signal processing. VAD plays an important role in a variety of applications such as voice recognition, speech enhancement, and speech coding where there is a requirement to classify audio

Implementation	VAD method
openSmile[1]	LSTM RNN
webrtcvad[2]	Gaussian Mixture Model
VAD-python[3]	Energy Based

Table 1: VAD implementations

	True positive	False positive	Failed detections
OpenSmile	62%	23%	15%
webrtcvad	43%	50%	7%
VAD-python	26%	63%	11%

Table 2: VAD Results

data based as containing speech or non-speech data. The most common VAD algorithms are based on an energy threshold, however some rely on more sophisticated models such as pattern recognition.

For Task 4.2, we reviewed and tested several state of the art VAD implementations as shown in table 1.

In order to compare the VAD algorithms, we used pre-recorded samples of childrens voices from Task 4.1. To simplify the evaluation, we used one female and one male voice of 30 seconds duration that included some natural background noise from the school environment. Table 2 shows the results divided into true or false positives and failed detections (failed to detect voice).

Based on the results, it is clear that the OpenSmile VAD, which is based on a LSTM Recurrent Neural Network, outperforms the other two implementations, especially in situations with dynamic levels and different types of background noise as found in school environments. For that reason, we decided to utilise openSmile as the voice activity detector in L2TOR, as its trained model copes well with childrens voices in noisy environments. Additionally, OpenSmile allows us to parametrize its options to finely tune to the environment.

3 Identifying and Interacting with Children

WP4 is responsible for perceiving and analysing the environment through a variety of sensors and provide meaningful information to the InteractionManager module (see WP5). Part of the perceptions capabilities of the robot is the Face detection which is required for believable and contingent social interaction between the robot and the humans. In addition, face recognition allows the robot to personalise its behaviour and tutoring based on the recognised user. Both recognizers inform task 4.2 and have been combined into the Perception module that handles the information from the sensors and the robots as described in the next two sections.

3.1 Face Recognition

Face recognition is one of the most challenging task for the computers. That is true especially when the users to be recognized are 4 to 5 years old as the recognisers have been optimised for adult faces. In L2TOR, we utilise the embedded ALFaceRecognition approach that is installed on the Nao robots and is based on OMRON's OKAO libraries. This recogniser works by comparing the user's face with a

Figure 1: Setup used to evaluate the effectiveness of face detection in a spontaneous interaction between two children. Children are facing each other and sit on cushions. Each child wears a bright sports bib, either purple or yellow, to facilitate later identification. Cameras are mounted at position where the robot would be observing the children.

preloaded user database and returns a confidence value of the closest match(es). Tests in a university environment showed that accuracy drops significantly when the lighting conditions change or when a large number of users are added to the database. We conclude that face recognition is currently not mature enough for use in real-world environments.

For that reason, a backup option will be used which allows manual identification by an operator. This will allow the robot to correctly recognise the user, even if the system fails to perform the automatic face recognition.

3.2 Face Detection and Tracking

Face detection is the process of finding a human face and keep tracking it until it moves out of the viewing point of the camera. NAO robots provide the ALFaceDetection module that uses the frontal camera on the head to detect and track a user. While this module offers sufficient face tracking, it inherently restricts the operation of the tracker as the camera is mounted on the head which is constantly moving while its performing behaviours. To overcome this issue, we used a Microsoft Kinect V2 sensor on a fixed location in front of the user that captures depth and image information (RGBD). The face tracking on Kinect provides a stable reference point of the head in 3D space along with its orientation. The Perception module receives face information from both Kinect and the robot and distributes them to the corresponding modules.

We evaluated face detection using the setup shown in figure 1. We use the face detection software from the gazr library[4], which relies on the dlib face tracker [5], which in turn is an implementation of [6]. We recorded 64 children between the ages of 4 and 8. Interactions lasted from 5 to 40 min (M=21m58s, SD=11m2s), which resulted in 2,232,978 video frames. The children's faces were detected in 1,208,309 of these frames, equating to M=54% of the frames (SD=22%, meaning that there is considerable variation between children). This illustrates the challenge of solely relying on face detection to drive the interaction: children are very dynamics, frequently looking away from the

[4] https://github.com/severin-lemaignan/gazr
[5] http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html

screen, the cameras and from each other. This negatively impacts the possibility to reliably detect their faces. Ensuing from that, the reliability of measures relying on face analysis (such as the detection of emotions or engagement) is restricted by the ability to detect faces.

4 Identifying and Tracking Objects

As part of the work undertaken for T4.1 for L2TOR we conducted a review of available technology for real-world object recognition and tracking. The findings in full have been submitted to HAI 2017 [6] and the paper has been included as Annex 6.2 in this document.

Our review took into account factors that would affect the studies we intend to conduct, and present challenges to deployment into multiple schools. These factors include changing backgrounds,lighting conditions and occlusion. A number of techniques were considered based on 2D video stream, RGB-D streams (colour and depth), and non-vision based.

The following is a summary of the findings from the paper:

- Fiducial markers provided highly stable and accurate pose information. However it is highly vulnerable to occlusion. It also requires the use of additional blocks for affixing the markers.
- Feature tracking methods were unable to handle varying backgrounds. This causes issue when the camera is moving, or if a participant enters the view.
- Template matching was robust, but does not scale well with multiple objects.
- Machine learning performs well for the objects it has been trained for, but is unable to handle iconic representations of objects it has learned. This means a specific dataset for training, which would require a lot of time to prepare. Preparation of our own data set would also be necessary to obtain proper depth information.
- The implementation of tabletop segmentation that was tested was unstable, producing a lot of false positives. The planar segmentation itself could be useful when implemented with another method.
- Intel's Realsense SDK performed well, but would sometimes lose an object, requiring the object to be moved before it was re-acquired. While not common this would still cause too many problems during a study, or when deployed at schools.
- None of the vision based techniques were fully capable of performing to the required standard on their own. Time spent developing a pipeline of vision techniques would be required to develop a robust system.
- Magnetic sensors perform well for tracking a single object. However to distinguish between multiple objects requires the use of RFID tags. The sensors can also only measure a few centimetres above them.
- An NFC mat is highly reliable and accurate. Similar to the magnetic sensor however it can only measure a few centimetres above the actual mat itself. Still for many of the situations required for the lessons this may be the most accurate and reliable. The mats themselves however cost roughly €1400 each.

[6] http://hai-conference.net/hai2017/

Work by Vlaar et al. [7] has shown that in the context of learning a second language, that real world objects present no advantage to learning new words than using objects displayed on a tablet. Based on these findings, and the review of object tracking techniques it has been decided that the focus will be on the use of a virtual scene on a tablet.

Underworlds [7] is a software framework for tracking geometric and temporal representations for robots. Work has been completed to enable the use of this software for the representation of the position of objects for the L2TOR project. Further details can be found in Section 5.

5 Software Prototype

Figure 2 represents the agreed architecture of the L2TOR project for both number and spatial domains. WP4 is responsible for Underworlds, Perception Manager and the Tablet Game.

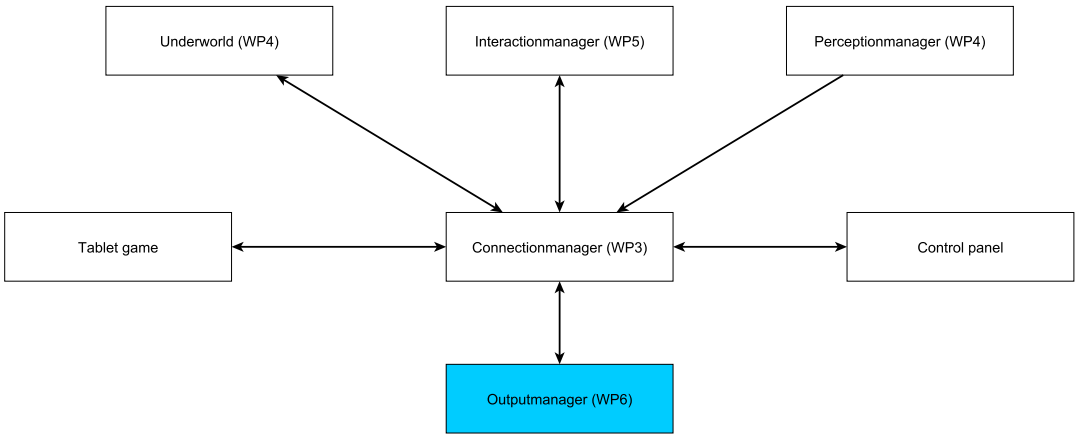

Figure 2: L2TOR architecture

5.1 Underworlds

Underworlds is a software framework for tracking geometric and temporal representations for robots. Underworlds is currently coded in Python, but future plans include the addition of C++ wrappers. Work has been completed for L2TOR to allow Underworlds to communicate with other modules in the project via the use of JSON messages. The interaction manager module is able to send a message to Underworlds to load a scene that is to be used for the current lesson. Messages can also be sent from the tablet game to update Underworlds with the new locations of objects after they have been moved. This framework also allows for messages from software that is providing the location of physical objects, if this is still introduced at a later date. Underworlds can also calculate the basic spatial relations necessary for the lessons in the number domain and communicate these to the interaction manager. The calculation of spatial relations will be further updated for the spatial domain.

[7] https://github.com/severin-lemaignan/underworlds

5.2 Tablet Game

Tablet Game is a HTML based 3D game (Figure 3) that allows object manipulation through touch and provides a basic collision detection model to enable spatial reasoning. We decided to use a 3D version of a game as it enhances the fidelity of digital representations and spatial abilities of the user. This game allow the users the drag a number of objects in X and Y axis (we disabled the Z axis as it is difficult to manipulate through touch and simplifies the interaction) according to robot's instructions. The game is capable of loading numerous 3D objects dynamically with textures. Each scene is loaded via JSON messages that inform the locations and characteristics of each object (e.g., shadows, collisions, rotation, colour etc.). The game is connected to the rest of the system via the Web Socket protocol and sends updates for the location of the objects on every touch. The game engine does not include any rules as that is handled by the InteractionManager in WP5.

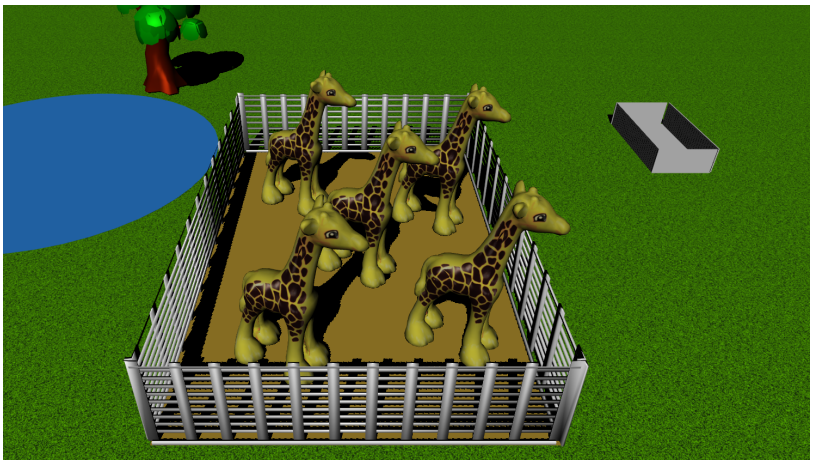

Figure 3: Tablet Game

5.3 Perception Manager

The Perception Manager is the centralised module that handles all the incoming data about the surroundings through the sensors. This module comprises of the VAD detector, Kinect module and NAO perception. Perception Manager initialises and starts the VAD detector which can run either on the tablet or on the operator's computer as long as the selected microphone is close to the user. The manager filters the messages and forwards them to the rest of the system with the following structure: startVAD and stopVAD with timestamps. At this stage, Kinect module retrieves the position and orientation of the user's head and calculates the approximate gaze of the user. This in turn will allow InteractionManager to facilitate more natural interactions through direct gaze. Additionally, head location will be used by Output Manager to track the user's face in real time. NAO perception is a python module that sits as a service on the robot and manipulates the data from both ALFaceDetection and ALFaceRecognition.

References

[1] Dong Yu and Li Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Springer, 2015.

[2] Michael L Seltzer, Dong Yu, and Yongqiang Wang. An investigation of deep neural networks for noise robust speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7398–7402. IEEE, 2013.

[3] Jill Fain Lehman. Robo fashion world: a multimodal corpus of multi-child human-computer interaction. In *Proceedings of the 2014 workshop on Understanding and Modeling Multiparty, Multimodal Interactions*, pages 15–20. ACM, 2014.

[4] J. Kennedy, S. Lemaignan, C. Montassier, P. Lavalade, B. Irfan, F. Papadopoulos, E. Senft, and T. Belpaeme. Child Speech Recognition in Human-Robot Interaction: Evaluations and Recommendations. In *Proceedings of the 12th ACM/IEEE International Conference on Human-Robot Interaction*, 2017.

[5] Paul Baxter, Rachel Wood, and Tony Belpaeme. A touchscreen-based 'sandtray' to facilitate, mediate and contextualise human-robot social interaction. In *Proceedings of the 7th annual ACM/IEEE international conference on Human-Robot Interaction*, pages 105–106. ACM, 2012.

[6] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[7] Rianne Vlaar, Josje Verhagen, Ora Oudgenoeg-Paz, and Paul Leseman. Comparing L2 word learning through a tablet or real objects: What benefits learning most? In *Proceedings of ACM HRI conference, Vienna, Austria, March 2017 (R4L workshop at HRI 2017)*, 2017.

6 Annex Descriptions

6.1 Kennedy, J. et al. (2017), Child Speech Recognition in Human-Robot Interaction: Evaluations and Recommendations

Bibliography - Kennedy, J., Lemaignan, S., Montassier, C., Lavalade, P., Irfan, B., Papadopoulos, F., Senft, E., Belpaeme, T. (2017) Child Speech Recognition in Human-Robot Interaction: Evaluations and Recommendations. In *Proceedings of the 12th IEEE/ACM International Conference on Human Robot Interaction*. DOI: 10.1145/2909824.3020229

Abstract - An increasing number of human-robot interaction (HRI) studies are now taking place in applied settings with children. These interactions often hinge on verbal interaction to effectively achieve their goals. Great advances have been made in adult speech recognition and it is often assumed that these advances will carry over to the HRI domain and to interactions with children. In this paper, we evaluate a number of automatic speech recognition (ASR) engines under a variety of conditions, inspired by real-world social HRI conditions. Using the data collected we demonstrate that there is still much work to be done in ASR for child speech, with interactions relying solely on this modality still out of reach. However, we also make recommendations for child-robot interaction design in order to maximise the capability that does currently exist.

Relation to WP - This work directly contributes to Task T4.1.

6.2 Wallbridge, C.D. et al. (2017), Qualative Review of Object Recognition Techniques for Tabletop Manipulation

Bibliography - Wallbridge, C.D., Lemaignan, S., Belpaeme, T. (2017) Qualative Review of Object Recognition Techniques for Tabletop Manipulation. Submitted to HAI

Abstract - This paper provides a qualitative review of different object recognition techniques relevant for near-proximity Human-Robot Interaction. These techniques are divided into three categories: 2D correspondence, 3D correspondence and non-vision based methods. For each technique an implementation is chosen that is representative of the existing technology to provide a broad review to assist in selecting an appropriate method for tabletop object recognition manipulation. For each of these techniques we give their strengths and weaknesses based on defined criteria. We then discuss and provide recommendations for each of them.

Relation to WP - This work directly contributes to Task T4.1.

A Annexes

Child Speech Recognition in Human-Robot Interaction: Evaluations and Recommendations

James Kennedy*
Plymouth University, U.K.

Séverin Lemaignan
Plymouth University, U.K.

Caroline Montassier
INSA Rouen, France

Pauline Lavalade
Université Pierre et Marie Curie, France

Bahar Irfan
Plymouth University, U.K.

Fotios Papadopoulos
Plymouth University, U.K.

Emmanuel Senft
Plymouth University, U.K.

Tony Belpaeme
Plymouth University, U.K.
Ghent University, Belgium

ABSTRACT

An increasing number of human-robot interaction (HRI) studies are now taking place in applied settings with children. These interactions often hinge on verbal interaction to effectively achieve their goals. Great advances have been made in adult speech recognition and it is often assumed that these advances will carry over to the HRI domain and to interactions with children. In this paper, we evaluate a number of automatic speech recognition (ASR) engines under a variety of conditions, inspired by real-world social HRI conditions. Using the data collected we demonstrate that there is still much work to be done in ASR for child speech, with interactions relying solely on this modality still out of reach. However, we also make recommendations for child-robot interaction design in order to maximise the capability that does currently exist.

Keywords

Child-Robot Interaction; Automatic Speech Recognition; Verbal Interaction; Interaction Design Recommendations

1. INTRODUCTION

Child-robot interaction is moving out of lab and into 'the wild', contributing to domains such as health-care [2], education [15, 25], and entertainment [20]. An increasing amount is being understood about how to design interactions from a nonverbal behaviour perspective [13, 14], but many of these domains hinge on effective verbal communication. This includes not only appropriate speech production by robots, but transcribing and understanding speech from young users as well. A prerequisite to this interpretation of speech is having a sufficiently accurate transcription of what is being said.

*Corresponding author: james.kennedy@plymouth.ac.uk

HRI '17, March 06 - 09, 2017, Vienna, Austria

ISBN 978-1-4503-4336-7/17/03…$15.00

DOI: http://dx.doi.org/10.1145/2909824.3020229

For this reason, high-quality Automatic Speech Recognition (ASR) is a vital component for producing autonomous human-robot interaction. ASR engines have undergone significant improvements in recent years, particularly following the introduction of new techniques such as deep learning [26]. However, these engines are commonly evaluated against standardised datasets of adult speech [23]. One might naively assume that these improvements will also translate to child speech, and will cope relatively well with noisy (i.e., real-world) environments, such as those experienced in applied HRI. However, this is often observed to not be the case, cf. [19].

In this paper we seek to evaluate the state-of-the-art in speech recognition for child speech, and to test ASR engines in settings inspired by real-world child-robot interactions. We record a variety of pre-determined phrases and spontaneous speech from a number of children speaking English using multiple microphones. We separate recordings by whether they are comparatively clean, or contain noise from the real-world environment. Through consideration of the results, we highlight the limitations of ASR for child speech, and also make a number of interaction design recommendations to maximise the efficacy of the technology currently available.

2. BACKGROUND

Speech recognition has undergone significant advances, building on or moving on from the use of Hidden Markov Models (HMM) towards using deep neural networks (DNN). DNNs have been shown to outperform older HMM based approaches by some margin against standard benchmarks [12]. For example, in a Google speech recognition task a deep neural network reduced the Word Error Rate (WER) to 12.3%, a 23% relative improvement on the previous state-of-the-art [12].

However, these benchmarks are based on adult speech corpora, such as the TIMIT corpus [17]. It has been noted by other researchers that there is a lack of corpora for children's speech, leading to a lack of training data and a lack of benchmarking for children's speech recognition models [5, 9, 11]. It is commonly assumed that the recent improvements observed in adult speech recognition mean that child speech recognition improved at the same pace, and recognising children's utterances can be achieved with a similar degree of success. However, anecdotal evidence suggests that this is not the

case; Lehman et al. [19] state that recognition of children's speech "remains an unsolved problem", calling for research to be undertaken to understand more about the limitations of ASR for children to ease interaction design.

Children's speech is fundamentally different from adult speech: the most marked difference being the higher pitched voice, due to children having a shorter, immature vocal tract. In addition, spontaneous child speech is marked by a higher number of disfluencies and, especially in younger children, language utterances are often ungrammatical (e.g., "The boy *putted* the frog in the box"). As such, typical ASR engines, which are trained on adult speech, struggle to correctly recognise children's speech [8, 24]. An added complexity is caused by the ongoing development of the vocal apparatus and language performance in children: an ASR engine trained for one age group is unlikely to perform well for another age group.

There have been various attempts to remedy this, from adapting adult-trained ASR engines to the spectral characteristics of children's speech [18,22], to training ASR engines on child speech corpora [6, 8, 10], or combinations of both. For example, Liao et al. [21] have used spoken search instructions from YouTube Kids to train DNNs with some success, resulting in a WER between 10 and 20%. In [24] vocal-tract length normalisation (VTLN) and DNN are used in combination, and when trained on read speech of children aged between 7 and 13 years, result in a WER of approximately 10%. It should be noted that these results are achieved in limited domains, such as spoken search instructions, read speech, or number recognition [22]. Also, the circumstances in which the speech is recorded are typically more controlled than interactions encountered in HRI, where ambient noise, distance and orientation to the microphone, and language use are more variable.

Whilst children's speech recognition in general is a challenge, HRI brings further complexities due to factors such as robot motor noise, robot fan noise, placement and orientation of microphones, and so on. Many researchers adopt interaction approaches that do not rely on verbal interaction due to the unreliability of child ASR, particularly in 'wild' environments. Wizard of Oz (WoZ) approaches have proven popular to substitute for sub-optimal speech recognition and natural language interaction, but when autonomy is important, WoZ is impractical and the use of mediating interfaces to substitute for linguistic interaction has proven successful. Touchscreens, for example, can serve as interaction devices, they provide a focus for the interaction while constraining the unfolding interaction [1]. However, if we wish the field to continue to progress into real-world environments, then it is unrealistic to exclude verbal interaction due to the prevalence of this communication channel in natural interaction.

3. RESEARCH QUESTIONS

The previous section highlights that the current performance of ASR for child speech remains unclear. We wish to address this by exploring different variables in the context of child speech, such as the type of microphone, the physical location of the speaker relative to a robot, and the ASR engine. These variables motivate a set of research questions presented below, all in the context of child speech. Their evaluation will be conducted with the aim of producing evidenced guidelines for designing verbal human-robot interactions with children.

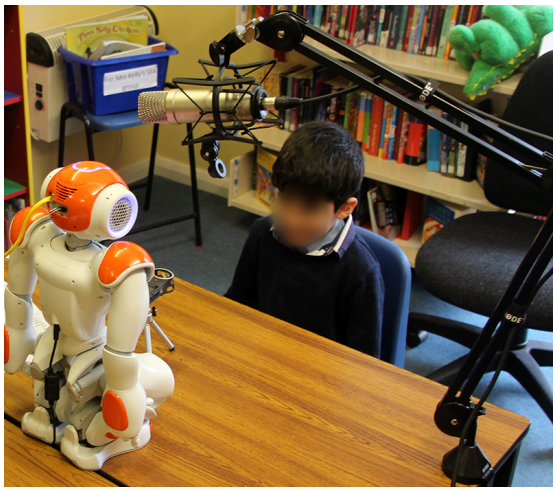

Figure 1: Equipment layout for recording children in a school. The Aldebaran NAO is turned on (but not moving) and records to a USB memory card. The studio microphone and portable microphone record simultaneously.

Q1 Do external microphones produce better results than robot-mounted microphones?

Q2 How can physical interaction setups be optimised for ASR?

Q3 Is there a benefit to using cloud-based or off-board ASR engines compared to a stock robot ASR engine?

Q4 What is the impact of 'real-world' noise on speech recognition in an HRI inspired scenario?

4. METHODOLOGY

In order to address the research questions posed in the previous section, a data collection and testing procedure was designed. At the time of writing, no corpus of child speech suitable for the intended analysis was publicly available. As such, there is a need for the collection of this data; the procedure for this will be outlined here.

4.1 Participants

A total of 11 children took part in our study, with an average age M=4.9, SD=0.3; 5F/6M. The age group is motivated by the many large-scale initiatives in the US, Europe and Japan exploring linguistic interactions in HRI [2,3,19,20,25], and the fact that this age group is preliterate, so cannot interact using text interfaces. All children had age-appropriate competency in speaking English at school. All participants gave consent to take part in the study, with the children's parents providing additional consent for participation, and recording and using the audio data. The children were rewarded after the study with a presentation of social robots.

4.2 Data Collection

In order to collect a variety of speech utterances, three different categories were devised: single word utterances, multi-word utterances, and spontaneous speech. The single word and multi-word utterances were collected by repeating

after an experimenter. This was done to prevent any issues with child reading ability. Spontaneous speech was collected through retelling a picture book, 'Frog, Where Are You?' by Mercer Mayer, which is a common stimulus for this activity in language development studies [4]. The single word utterances were numbers from 1 to 10, and the multi-word utterances were based on spatial relationships between two nouns, for example, 'the horse is in the stable'. Five sentences of this style were used; the full set can be downloaded from [16].

The English speech from children was collected at a primary school in the U.K. This served two purposes: firstly, to conduct the collection in an environment in which the children are comfortable, and secondly, to collect data with background noise from a real-world environment commonly used in HRI studies, e.g., [15]. An Aldebaran NAO (hardware version 5.0 running the NaoQi 2.1.4 software) was used as the robotic platform. This was selected as it is a commonly used platform for research with children, as well as for its microphone array and commercial-standard speech recognition engine (provided by Nuance). The robot would record directly from the microphones to a USB memory stick. Simultaneously, a studio grade microphone (Rode NT1-A) and a portable microphone (Zoom H1) were also recording. The studio microphone was placed above the robot and the portable microphone just in front of the robot (Fig. 1).

4.3 Data Processing

Encoding and Segmentation.

All audio files were recorded in lossless WAV format (minimum sampling rate of 44kHz). The audio files from each of the three microphones were synchronised in a single Audacity project. The audio files were then split to extract segments containing the speech under consideration. These segments were exported as lossless WAV files, resulting in 16 files per microphone (48 in total) per child. The spontaneous speech was transcribed and split into sentences. This produced a total of 222 spontaneous speech utterances of various lengths ($M = 7.8$ words per utterance, $SD = 2.6$). The full dataset (audio files and transcripts) is available online at [16].

Noisy vs. Clean Audio Recordings.

As the recordings of children in English were collected during the course of a school day, there is a range of background noise. To study the impact of noise on ASR performance, it is desirable to separate the recordings into those that have minimal background noise ('clean' recordings) and those that have marked background noise ('noisy' recordings). Some noise is unavoidable, or would be present in any HRI scenario, such as robot fan noise, so these were considered 'clean'. Other noise, such as birds outside, other children shouting from the adjacent room, doors closing, or coughing would be considered 'noisy'. This means that the clean recordings are not noise-free like those from a studio environment, but are a realistic representation of a minimal practical noise level in a 'wild' HRI scenario, thereby allowing us to evaluate recognition accuracy with greater veracity.

To appropriately categorise the recordings as clean or noisy, each one was independently listened to by 3 human coders with the guidance from above as to what is considered clean vs. noisy. Overall agreement levels between coders was good, with Fleiss $\kappa = .74$ (95% CI [.65,.84]) for the fixed utterances and $\kappa = .68$ (95% CI [.60,.75]) for the spontaneous

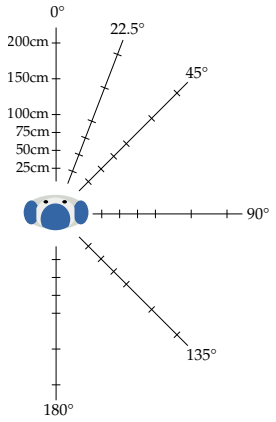

Figure 2: Locations at which speech it played to the NAO to explore how the physical layout of interactions may influence speech recognition rates.

utterances. A recording was categorised as noisy or clean if all 3 coders agreed it was respectively noisy or clean. Where there was any disagreement between coders, the recordings were omitted from analysis of noise impact (59 fixed and 54 spontaneous utterances were excluded). This resulted in 80 noisy recordings, and 37 clean recordings being analysed from the fixed utterances set and 83 clean/85 noisy recordings from the spontaneous utterances set. For some children, the NAO recording failed due to technical difficulties. Therefore, when comparing across microphones, the fixed utterance selection is reduced to 29 clean recordings and 60 noisy recordings.

Manipulation of the Sound Location.

To evaluate the impact of distance and angle on speech recognition, it was necessary to vary the distance between the robot and child, while at the same time keeping the speech utterances constant. As children struggle to exactly reproduce speech acts and over 500 utterances are needed to be recognised, we used pre-recorded speech played through an audio reference speaker (the PreSonus Eris E5) placed at different locations around the robot. In order to match the original volume levels, a calibration process was used where a recording would be played and re-recorded at the original distance between the child and the robot. The audio signal amplitudes between the original and recorded file were then compared. The speaker volume was iteratively revised until the amplitudes matched. This volume was then maintained as the speaker was moved to different distances and angles from the robot, while always facing the robot (to address, at least in part, Q2 from Sec. 3); see Fig. 2 for a diagram of these positions.

4.4 Measures

For recognition cases where a *multiple choice* grammar is used (i.e., the list of possible utterances is entirely predefined, and the recognition engine's task is to pick the correct one), the recognition percentage is used as the metric. Each word or sentence correctly recognised adds 1; the final sum is divided by the number of tested words or sentences. All Confidence Intervals calculated for the recognition percentage include continuity correction using the Wilson procedure. We

use the same metric when using template-based grammars (Sec. 5.2.1).

For the cases in which an open grammar is used, we use the Levenshtein distance as a metric at the *letter* level. This decision was made as it reduces punishment for small errors in recognition, which would typically not be of concern for HRI scenarios. For example, when using the Levenshtein distance at the word level (as with Word Error Rate), if the word 'robots' is returned for an input utterance of 'robot', this would be scored as completely unrecognised. At the letter level, this would score a Levenshtein distance of 1, as only a single letter needs to be inserted, deleted or substituted (in this example, the letter 's') to get the correct result. To compare between utterances, normalisation by the number of letters in the utterance is then required to compensate for longer inputs incurring greater possibility of higher Levenshtein distances.

5. RESULTS

This section will break down the results and analysis such that the research questions are addressed. The results are split into two main subsections concerning: 1) technical implementation details, and 2) general ASR performance. The intention is to then provide a practical guide for getting the best performance from ASR in HRI scenarios, as well as an indication of the performance level that can be expected more generally for child speech under different circumstances.

5.1 Technical Best Practices

Throughout this subsection, the ASR engine will remain constant so that other variables can be explored. In this case, the ASR engine used is the one that comes as default on the Aldebaran NAO, provided by Nuance (VoCon 4.7). A grammar is provided to this engine, consisting of numbers (as described in Sec. 4.2) and single word utterances. Longer utterances, along with open grammar and spontaneous speech will be explored in the subsequent subsection.

5.1.1 Type of microphone

Upon observation of the results it became clear that the robot-mounted microphone was vastly outperforming the portable and studio microphones. When visually comparing the waveforms, there was a noticeable difference in recorded amplitude between the NAO signal and the other two microphones. This was despite the standalone microphone input gains being adjusted to maximise the signal (whilst preventing peak clipping). To increase the signal amplitude whilst maintaining the signal-to-noise ratio, the files were normalised. This normalisation step made a significant difference to the results of the speech recognition. For the portable microphone, the recognition percentage after normalisation (70%, 95% CI [59%,79%]) was significantly improved compared to before normalisation (2%, 95% CI [0%,9%]); Wilcoxon signed-rank test[1] $Z = -7.483, p < .001, r = 0.67$. A similar improvement was observed for the studio microphone when comparing before (5%, 95% CI [2%,12%]) and after (81%, 95% CI [70%,88%]) normalisation; $Z = -7.937, p < .001, r = 0.71$ (Fig. 3). This suggests that the NAO microphones are tuned to maximise the speech level, and if

[1]Due to the recognition being binary on single word inputs, the resulting distributions are non-normal, so non-parametric tests are used for significance testing.

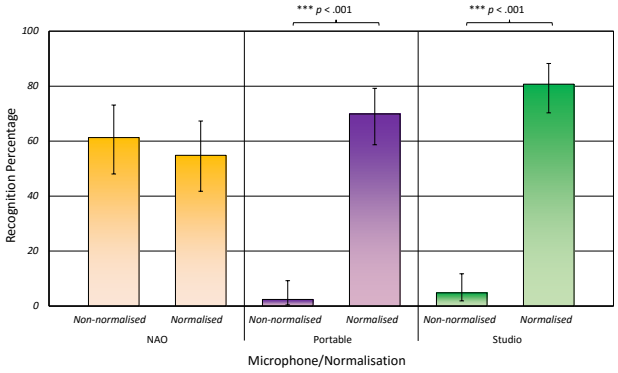

Figure 3: A comparison of recognition percentage of English words and short sentences spoken by children, split by microphone before and after normalisation. * indicates significance at the $p<.001$ level. The recognition is much improved for the portable and studio microphones following normalisation.**

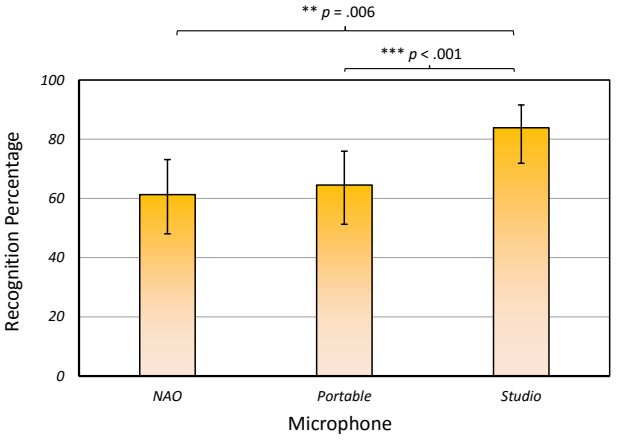

Figure 4: Recognition percentage of numbers spoken by children, split by microphone type (62 utterances). * indicates significance at the $p<.001$ level, ** indicates significance at the $p<.01$ level. The studio microphone provides the best ASR performance, but the difference between on- and lower quality off-board microphones is relatively small.**

external microphones are to be used, then normalisation of the recordings should be considered a vital step in processing prior to sending to an ASR engine. Therefore, for the remainder of the analysis here, only normalised files are used for the studio and portable microphones.

In exploring Q1, it is observed that the differences between microphones is smaller than may have been expected. The NAO microphones are mounted in the head of the robot near a cooling fan which produces a large amount of background noise. It could therefore be hypothesised that the ASR performance would greatly increase by using an off-board microphone, and that using a higher-quality microphone would improve this further. Using Friedman's test, a significant difference at the $p < .05$ level is found between the NAO (61%, 95% CI [48%,73%]), portable (65%, 95% CI [51%,76%]), and studio (84%, 95% CI [72%,92%]) micro-

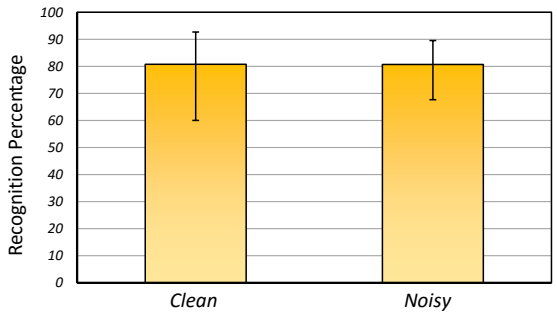

Figure 5: Recognition percentage of single word utterances spoken by children, split by background noise level (83 total utterances). Noise level does not have a significant effect on the recognition rate.

phones; $\chi^2(2) = 9.829, p = .007$. Post-hoc Wilcoxon signed-rank pairwise comparisons with Bonferroni correction reveal a statistically significant difference between the portable and studio microphones ($Z = -3.207, p < .001, r = 0.29$; Fig. 4), and between the NAO and studio microphones ($Z = -2.746, p = .006, r = 0.24$). Differences between the portable and NAO microphones ($Z = -0.365, p = .715, r = 0.03$) were not significant. This suggests that there is no intrinsic value to using an off-board microphone, but that a high quality off-board microphone can improve the ASR results. The difference between the robot microphone and the external studio grade microphone is fairly substantial, with a recognition percentage improvement of around 20%point ($r = 0.28$). It would be scenario specific as to whether the additional technical complexity of using a high-quality external microphone would be worth this gain, and indeed, in scenarios where the robot is mobile, use of a studio grade microphone may not be a practicable option.

5.1.2 Clean vs. Noisy Recording Environment

Splitting the files by whether they were judged to be clean or noisy (as described in Sec. 4.3), it was observed that the noise did not appear to have a significant impact on the results of the ASR. Using the studio microphone (i.e., the best performing microphone) for the number utterances, a Mann-Whitney U test reveals no significant difference between clean (81%, 95% CI [60%,93%]) and noisy (81%, 95% CI [68%,90%]) speech; $U = 740.5, p = .994, r = 0.00$ (Fig. 5). The apparent robustness of the ASR engine to noise is of particular benefit to HRI researchers given the increasingly 'real-world' application of robots, where background noise is often near impossible (nor desirable) to prevent.

However, this does not mean that noise does not play a role in recognition rates. In this instance, the ASR engine is restricted in its grammar; the effect of noise in open grammar situations is explored in the next subsection. Additionally, when the distance of the sound source to the microphone is varied, background noise becomes a greater factor.

5.1.3 Sound Source Location

Measurements were made as in Fig. 2 using the built in NAO microphone, with the replayed audio from the studio microphone (as described in Sec. 4.3). Due to the number of data points this generates (540 per child), the findings in full will not be produced here, but to get a high-level picture

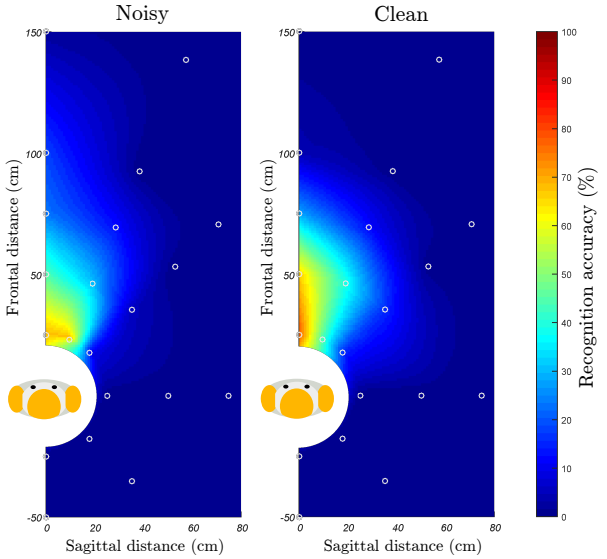

Figure 6: Interpolated heatmap of recognition percentage as a function of distance and orientation to the robot. Interpolation has been performed based on the measurements made at the small white circles. On the *left* is the heatmap for the noisy audio, whereas the *right* is for clean audio. The clean audio is better recognised at further distances from the robot, however, in both cases, recognition accuracy is 0% to the side and behind the robot.

of how the distance and orientation influences recognition rates, a heatmap can be seen in Fig. 6.

Two observations can be made from this data that have particular relevance for HRI researchers. The first is the platform-specific observation that with the NAO robot (currently one of the most widely used research platforms for social HRI) the utterance recognition rate drops dramatically once the sound source reaches a 45 degree angle to the robot head, and becomes 0 once it reaches 90 degrees. The implication of this is that when using the NAO, it is vital to rotate the head to look at the sound source in order to have the possibility of recognising the speech. This is of course dependent on the current default software implementation; four channels of audio exist, but for ASR only the front two are used, and so a workaround could be created for this. The second, broader observation, is that the background noise and distance seem to influence recognition rates when combined. Fig. 5 shows how little impact noise has when the files

Distance (cm)	Clean % [95% CI]	Noisy % [95% CI]
25	73 [52,88]	77 [64,87]
50	65 [44,82]	44 [31,58]
75	27 [12,48]	23 [13,36]
100	4 [0,22]	18 [9,30]

Table 1: ASR recognition rates for children counting from one to ten. Recordings were played frontally at different distances from the robot. Note how recognition falls sharply with distance when the speech contains noise.

are fed directly into the robot ASR, but when combined with distance, there is a marked difference beyond 50cm. Table 1 shows the measurements for the first metre directly in front of the robot; at 25cm the difference between clean and noisy files are minimal, however at 50cm, the difference is more pronounced, with recognition rates dropping fast.

5.2 ASR Performance with Children

The previous subsection addressed variables in achieving a maximal possible speech recognition percentage through modifying the technical implementation, such as different microphones, distances to a robot, orientation to a robot, and background noise levels. This subsection will provide a complementary focus on exploring the current expected performance of ASR with children under different speech and ASR engine conditions. This will include a comparison of differing length utterances, spontaneous utterances, and different ASR engines with varying grammar specifications. For all analyses in this section, the studio microphone signal is used to provide the best quality sound input to the speech engines (and provide a theoretical maximal performance).

5.2.1 Impact of Providing a Grammar

Tests on child speech in the previous subsection were performed with single word utterances, with a grammar consisting of only those utterances. This kind of multiple choice is relatively straightforward, and this carries over to slightly longer utterances too. We compare the recognition rate of the fixed multi-word utterances (34 spatial relation sentences as described in Sec. 4.2) under 3 conditions using the built-in NAO ASR: 1) with a fixed grammar containing the complete utterances, e.g., "one" or "the dog is on the shed" (i.e., multiple choice), 2) with a template grammar for the sentences (as seen in Fig. 8), and 3) with an open grammar. This progressively reduces the prior knowledge the ASR engine has about what utterances to expect. The full mix of noisy and clean utterances were used as there was no observed significant correlation in any of the three conditions between ASR confidence level and noise condition, nor between noise condition and resulting recognition rates. The grammar condition has a significant impact on the recognition percentage; Friedman's test $\chi^2(2) = 39.92, p < .001$. Post-hoc Wilcoxon signed-rank pairwise comparisons with Bonferroni correction reveal a statistically significant difference between the multiple choice (74%, 95% CI [55%,86%]) and template grammars (53%, 95% CI [35%,70%]); $Z = -2.646, p = .008, r = 0.32$. The template grammar in turn offers a significant improvement over the open grammar (0%, 95% [0%,13%]); $Z = -4.243, p < .001, r = 0.51$ (Fig. 7).

5.2.2 Comparison of ASR Engines

Finally, we look at how different ASR engines perform, under identical recording conditions. We compare the Google Speech API (as found in the Chrome web browser for instance), the Microsoft Speech API (as found in the Bing search engine), CMU PocketSphinx, and the NAO-embedded Nuance VoCon 4.7 engine; studies were run in August 2016. The audio samples are those recorded with the studio microphone; they include native and non-native speakers as well as noisy and clean samples; they include both the fixed sentences and the spontaneous speech; no grammar is provided to the engine (i.e., open grammar).

As performing recognition with an open grammar is a much harder challenge for recognition engines, the recognition percentage alone is no longer a sufficient measurement to compare between performance of ASR engines due to the very low number of exact utterance recognitions across all engines. Instead we use the Levenshtein distance (LD) at the letter level. As the utterance length for the spontaneous speech is also variable, the Levenshtein distance is normalised by utterance length (as per Sec. 4.4). This provides a value between 0 and 1, where 0 means the returned transcription matches the actual utterance, and 1 means not a single letter was correct. Values in between indicate the proportion of letters that would have to be changed to get the correct response, therefore lower scores are better. Table 2 provides one recognition example with the corresponding Levenshtein distances.

While the LD provides a good indication of how close the result is from the input utterance, the examples in Table 2

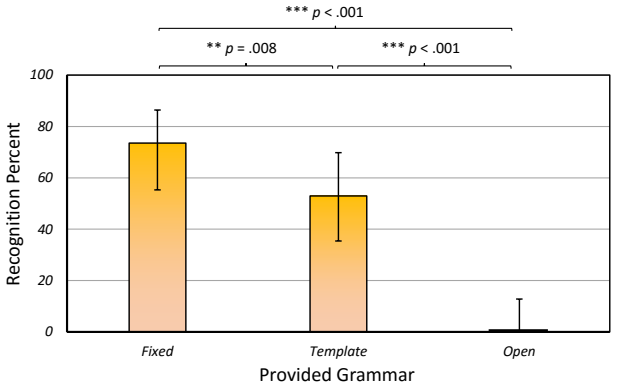

Figure 7: Recognition percentage when providing a fixed grammar, a template grammar, and an open grammar on short utterances. The fixed 'multiple choice' grammar produces the best recognition, followed by a template. The open grammar, on average, recognises almost no sentences correctly.

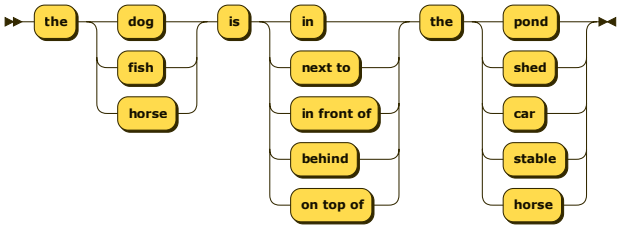

Figure 8: Template for the grammar provided to the ASR for the fixed utterances. 75 different sentences can be generated from this grammar.

Google API	*then the wraps looks at the dog*	[LD=0.17]
Microsoft API	*rat look at dogs*	[LD=0.48]
PocketSphinx	*look i personally*	[LD=0.83]

Table 2: Recognition results and Levenshtein distance for three ASR engines on the input utterance "then the rat looked at the dog". The NAO-embedded Nuance engine did not return any result.

evidence that this metric does not necessarily reflect *semantic* closeness. In this particular case, the Bing result "rat look at dogs" is semantically closer to the original utterance than the other answers. For this reason, we assess recognition performance in open grammar using a combination of three metrics: 1) the Levenshtein distance; 2) raw accuracy (i.e., the number of exact matches between the original utterance and the ASR result); 3) a manually-assessed 'relaxed' accuracy. The utterance would be considered accurate in the 'relaxed' category if small grammatical errors are present, but not semantic errors. Grammatical errors can include pluralisation, removal of repetitions, or small article changes ('the' instead of 'a'). For example, if an input utterance of "and then he found the dog" returned the result "and then he found a dog", this would be considered accurate, however "and then he found the frog" would produce a similar LD, but the semantics have changed, so this would not be included in the relaxed accuracy category.

Table 3 shows that when the input utterance set is changed to use spontaneous speech, the average normalised LD does not change much for any of the ASR engines. Nor do the LD rates change much when only clean spontaneous speech is used, providing further evidence for the minimal impact of noise as established in Sec. 5.1.2. However, there is a marked difference between Google and the other recognition engines. The average LD from Google is around half that of the other engines, and the number of recognised sentences in both the strict and relaxed categories is substantially higher. The recognition performance remains however generally low: using relaxed rules, the currently best performing ASR engine (Google Speech API) for our data recognises only about 18% of a corpus of 222 child utterances (utterances have a mean length of $M = 7.8$ words, $SD = 2.6$).

To help decide whether or not the results returned from Google would actually be usable in autonomous HRI scenarios, it is necessary to determine when the utterance is correctly recognised. This is typically indicated through the *confidence value* returned by the recognition engine. To further explore this, we assess the number recognition percentage at different thresholds within the confidence level (Fig. 9). A total of 101 results from the 222 passed to the recogniser returned a confidence level (a confidence value is not returned when the uncertainty of the ASR engine is too high). To achieve just below 50% semantically correct recognition accuracy, the confidence threshold could be set to 0.8, which would only include 36 utterances. While a clear improvement over the 18% previously achieved when not taking into consideration the confidence value, a 50% recognition rate is arguably not sufficient for a smooth child-robot verbal interaction, and would still require the system to reject nearly 2/3 of the child utterances.

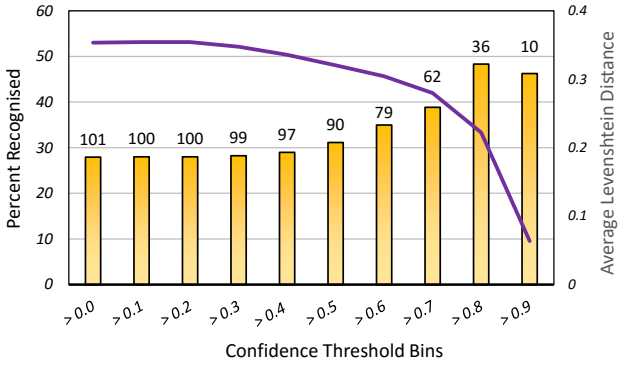

Figure 9: Histogram of recognition percentage (using the relaxed, manually coded criteria) for spontaneous speech grouped by confidence levels (indicated by the number above each bar) returned by Google ASR. The average Levenshtein Distance is also shown on the secondary axis. Recognition increases with higher confidence ranges, but few utterances have a high confidence.

6. DISCUSSION

Our results show that, at the time of writing, automatic speech recognition still does not work reliably with children, and should not be relied upon for autonomous child-robot interaction.

Speech segmentation is one aspect that we did not investigate. The segmentation of speech units and rejecting non-speech parts is an important factor in speech recognition. For example, noise can be mistakenly recognised by ASR engines as speech, or a pause in the middle of a sentence might interrupt the segmentation. Existing solutions (like a beep sound indicating when to talk) are not ideal for children of this age. Our manual segmentation likely leads to better results than would be expected with automatic segmentation.

We did not analyse if gender had an effect on ASR due to the age of the children used in the study. It has been shown that there are no significant differences in the vocal tract between genders at the age under consideration (5-6 years old) [7], so we do not expect differing performance based on gender.

Mitigation strategies for poor ASR performance depend on the ASR engine. We have specifically investigated the use of constrained grammar with the NAO's Nuance engine; and the use of the recognition confidence with the Google ASR. While severely constraining the interaction scope, none of these techniques were found to provide satisfactory results. In our most favourable test case (children speaking numbers from one to ten in front of the robot, at about 25cm; the robot having an explicit 'multiple choice' grammar), the ASR would return an incorrect result in one of four cases, and could not provide any meaningful confidence value. This result is disappointing, particularly when considering that interactions based on 'multiple choice' are difficult to rely on with children, as they tend not to remember and/or comply to the given set of recognisable utterances.

Template-based grammars (or 'slot-filling' grammars) where the general structure of the sentence is known beforehand, and only a limited set of options are available to fill the 'gaps' are a potentially interesting middle-ground between 'multiple choice' grammars and open speech. However, we show that in our test case (grammar depicted in Fig. 8), the correct utterance was recognised in only 50% of the cases, again without any useful confidence value.

In the realm of open grammars, the Google Speech API returned the most accurate results by a large margin. When run on grammatically correct, regular sentences (the ones generated from the grammar depicted in Fig. 8), it reaches 38% accuracy in recognition when minor grammatical differences are allowed. This result, while likely not yet usable in today's applications, is promising. However, when looking

	Google		Bing		Sphinx		Nuance	
	M LD [95%CI]	*% rec.*	*M* LD [95%CI]	*% rec.*	*M* LD [95%CI]	*% rec.*	*M* LD [95%CI]	*% rec.*
fixed (n=34)	**0.34** [0.24,0.44]	*11.8 [38]*	0.64 [0.56,0.71]	*0 [0]*	0.68 [0.64,0.73]	*0 [0]*	0.76 [0.73,0.80]	*0 [0]*
spontaneous (n=222)	**0.39** [0.36,0.43]	*6.8 [17.6]*	0.64 [0.61,0.67]	*0.5 [2.4]*	0.80 [0.77,0.84]	*0 [0]*	0.80 [0.78,0.82]	*0 [0]*
spontaneous clean only (n=83)	**0.40** [0.35,0.45]	*6.0 [16.9]*	0.63 [0.58,0.68]	*1.2 [1.2]*	0.78 [0.72,0.85]	*0 [0]*	0.78 [0.75,0.81]	*0 [0]*

Table 3: Comparison between four ASR engines using fixed, all spontaneous, and clean spontaneous speech utterances as input. Mean average normalised Levenshtein Distance (*M* LD) indicates how good the transcription is. *% rec* indicates the percentage of results that are an exact match for the original utterance, with the values in square brackets *[ ]* indicating matches with 'relaxed' accuracy.

at children's spontaneous speech, the recognition rate drops sharply (to around 18% of successful recognition). This difference can be explained by the numerous disfluencies and grammatical errors found in natural child speech. To provide an example, a relatively typical utterance from our data was "and... and the frog didn't went to sleep". The utterance has a repetition and disfluency at the start, and is followed by grammatically incorrect content. This is, in our opinion, the real challenge that automatic child speech recognition faces: the need to account for the child-specific language issues, beyond the mere differences between the acoustic models of adults vs. children. This is a challenge not only for speech-to-text, but as well for later stages of the verbal interaction, like speech understanding and dialogue management.

Our results allow us to make a number of recommendations for designing child-robot interaction scenarios that include verbal interaction. Most of these are also applicable to adult settings and would be expected to contribute to a smoother interaction.

- Constrain the interaction by leading the child to a limited set of responses. This typically works well for older children, but carries the risk of making the interaction stale.
- Use additional input/output devices. A touchscreen has been found to be a particularly effective substitute for linguistic input [1, 14], but also other devices –such as haptic devices– should be considered.
- Place the young user in the optimal location for ASR. The location and orientation relative to the microphone (and robot) has a profound impact on ASR performance (Sec. 5.1.3). A cushion, stool or chair can help children sit in the optimal location.
- Constrain the grammar of the ASR. While not all ASR engines allow for this (cf. Bing), some will allow constraints or "hints" on what is recognised. This proves to be valuable in constrained interaction settings, for example, when listening only for numbers between 1 and 10 (Sec. 5.2.1).
- Background noise appears to be less of an issue than initially anticipated. It appears that the current ASR engines have effective noise cancelling mechanisms in place. Nevertheless, "the less noise, the better" remains true, particularly when interacting at a distance from the robot (Sec's 5.1.2 & 5.1.3).
- A lack of ASR performance does not mean that the robot should not produce speech, as speech has been found to be particularly effective to engage children.

We opted to evaluate the ASR capabilities of the Aldebaran NAO platform, as it is the most commonly used robot in commercial and academic HRI. While the NAO system under performs for child speech, some performance could be gained through using a high-quality external microphone and cloud-based ASR, with Google as clear favourite.

7. CONCLUSION

Language is perhaps the most important modality in human-to-human interaction and as such, functional natural language interaction forms a formidable prize in human-machine interaction. Speech recognition is the entry point to this and while there has been steady progress in speaker-independent adult speech recognition, the same progress is currently lacking from children's speech recognition. For various reasons –pitch characteristics of children's voices, speech disfluencies, and unsteady developmental changes– child speech recognition is expected to require a multi-pronged approach and recognition performance in unconstrained domains is currently too low to be practical.

This has a profound impact on the interaction between children and technology, especially where pre-literacy children are concerned, typically ages 6 and younger. As they have no means of entering input other than by speaking to the device, the interaction with pre-literacy children stands or falls with good speech recognition.

Our results show that natural language interactions with children are not yet practicable. Today, building rich and natural interactions between robots and children still requires a complex alchemy: a careful design of the interaction that leads the responses of the young user in such a way that restrictive ASR grammars are acceptable, the understanding and production of rich non-verbal communication cues like gaze, and a judicious use of supporting technology such as touchscreens.

8. ACKNOWLEDGEMENTS

This work has been supported by the EU H2020 L2TOR project (grant 688014), the EU FP7 DREAM project (grant 611391), and the EU H2020 Marie Sklodowska-Curie Actions project DoRoThy (grant 657227).

9. REFERENCES

[1] P. Baxter, R. Wood, and T. Belpaeme. A touchscreen-based 'sandtray' to facilitate, mediate and contextualise human-robot social interaction. In *Proceedings of the 7th annual ACM/IEEE international conference on Human-Robot Interaction*, pages 105–106. ACM, 2012.

[2] T. Belpaeme, P. Baxter, R. Read, R. Wood, H. Cuayáhuitl, B. Kiefer, S. Racioppa, I. Kruijff-Korbayová, G. Athanasopoulos, V. Enescu, R. Looije, M. Neerincx, Y. Demiris, R. Ros-Espinoza, A. Beck, L. Cañamero, A. Hiolle, M. Lewis, I. Baroni, M. Nalin, P. Cosi, G. Paci, F. Tesser, G. Sommavilla, and R. Humbert. Multimodal Child-Robot Interaction: Building Social Bonds. *Journal of Human-Robot Interaction*, 1(2):33–53, 2012.

[3] T. Belpaeme, J. Kennedy, P. Baxter, P. Vogt, E. E. Krahmer, S. Kopp, K. Bergmann, P. Leseman, A. C. Küntay, T. Göksun, A. K. Pandey, R. Gelin, P. Koudelkova, and T. Deblieck. L2tor - second language tutoring using social robots. In *Proceedings of the 1st International Workshop on Educational Robots*, Paris, France, 2015.

[4] R. A. Berman and D. I. Slobin. *Relating events in narrative: A crosslinguistic developmental study*. Psychology Press, 2013.

[5] P. Cosi, M. Nicolao, G. Paci, G. Sommavilla, and F. Tesser. Comparing open source ASR toolkits on Italian children speech. In *Proceedings of the Workshop on Child Computer Interaction*, 2014.

[6] S. Fernando, R. K. Moore, D. Cameron, E. C. Collins, A. Millings, A. J. Sharkey, and T. J. Prescott. Automatic recognition of child speech for robotic applications in noisy environments. *arXiv preprint*, arXiv:1611.02695, 2016.

[7] W. T. Fitch and J. Giedd. Morphology and development of the human vocal tract: A study using magnetic resonance imaging. *The Journal of the Acoustical Society of America*, 106(3):1511–1522, 1999.

[8] M. Gerosa, D. Giuliani, S. Narayanan, and A. Potamianos. A review of ASR technologies for children's speech. In *Proceedings of the 2nd Workshop on Child, Computer and Interaction*, pages 7:1–7:8. ACM, 2009.

[9] P. Grill and J. Tučková. Speech databases of typical children and children with SLI. *PloS one*, 11(3):e0150365, 2016.

[10] A. Hagen, B. Pellom, and R. Cole. Children's speech recognition with application to interactive books and tutors. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, 2003*, pages 186–191. IEEE, 2003.

[11] A. Hämäläinen, S. Candeias, H. Cho, H. Meinedo, A. Abad, T. Pellegrini, M. Tjalve, I. Trancoso, and M. Sales Dias. Correlating ASR errors with developmental changes in speech production: A study of 3-10-year-old European Portuguese children's speech. In *Proceedings of the Workshop on Child Computer Interaction*, 2014.

[12] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[13] C.-M. Huang, S. Andrist, A. Sauppé, and B. Mutlu. Using gaze patterns to predict task intent in collaboration. *Frontiers in Psychology*, 6, 2015.

[14] J. Kennedy, P. Baxter, and T. Belpaeme. Nonverbal Immediacy as a Characterisation of Social Behaviour for Human-Robot Interaction. *International Journal of Social Robotics*, in press.

[15] J. Kennedy, P. Baxter, E. Senft, and T. Belpaeme. Social Robot Tutoring for Child Second Language Learning. In *Proceedings of the 11th ACM/IEEE International Conference on Human-Robot Interaction*, pages 67–74. ACM, 2016.

[16] J. Kennedy, S. Lemaignan, C. Montassier, P. Lavalade, B. Irfan, F. Papadopoulos, E. Senft, and T. Belpaeme. Children speech recording (English, spontaneous speech + pre-defined sentences). Data set, 2016. http://doi.org/10.5281/zenodo.200495.

[17] L. F. Lamel, R. H. Kassel, and S. Seneff. Speech database development: Design and analysis of the acoustic-phonetic corpus. In *Speech Input/Output Assessment and Speech Databases*, 1989.

[18] L. Lee and R. Rose. A frequency warping approach to speaker normalization. *IEEE Transactions on Speech and Audio Processing*, 6(1):49–60, Jan 1998.

[19] J. F. Lehman. Robo fashion world: a multimodal corpus of multi-child human-computer interaction. In *Proceedings of the 2014 Workshop on Understanding and Modeling Multiparty, Multimodal Interactions*, pages 15–20. ACM, 2014.

[20] I. Leite, H. Hajishirzi, S. Andrist, and J. Lehman. Managing chaos: models of turn-taking in character-multichild interactions. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, pages 43–50. ACM, 2013.

[21] H. Liao, G. Pundak, O. Siohan, M. Carroll, N. Coccaro, Q.-M. Jiang, T. N. Sainath, A. Senior, F. Beaufays, and M. Bacchiani. Large vocabulary automatic speech recognition for children. In *Proceedings of Interspeech*, 2015.

[22] A. Potamianos and S. Narayanan. Robust recognition of children's speech. *IEEE Transactions on Speech and Audio Processing*, 11(6):603–616, 2003.

[23] M. L. Seltzer, D. Yu, and Y. Wang. An investigation of deep neural networks for noise robust speech recognition. In *Proccedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7398–7402. IEEE, 2013.

[24] R. Serizel and D. Giuliani. Deep-neural network approaches for speech recognition with heterogeneous groups of speakers including children. *Natural Language Engineering*, 1:1–26, 2016.

[25] F. Tanaka, K. Isshiki, F. Takahashi, M. Uekusa, R. Sei, and K. Hayashi. Pepper learns together with children: Development of an educational application. In *Proceedings of the IEEE-RAS 15th International Conference on Humanoid Robots*, HUMANOIDS 2015, pages 270–275. IEEE, 2015.

[26] D. Yu and L. Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Springer, 2015.

Qualitative Review of Object Recognition Techniques for Tabletop Manipulation

Anonymous Author
for Submission
City, Country
e-mail address

Anonymous Author
for Submission
City, Country
e-mail address

Anonymous Author
for Submission
City, Country
e-mail address

ABSTRACT

This paper provides a qualitative review of different object recognition techniques relevant for near-proximity Human-Robot Interaction. These techniques are divided into three categories: 2D correspondence, 3D correspondence and non-vision based methods. For each technique an implementation is chosen that is representative of the existing technology to provide a broad review to assist in selecting an appropriate method for tabletop object recognition manipulation. For each of these techniques we give their strengths and weaknesses based on defined criteria. We then discuss and provide recommendations for each of them.

Author Keywords

object detection; pose detection; tabletop manipulation.

INTRODUCTION

Context: near object interaction

This paper takes a practical approach to survey the technical landscape on the problem of small object identification and 6D object localisation in a cluttered environment – a context often termed as *object recognition for tabletop manipulation*. Our approach is practical: we consider a typical interaction setup (Fig. 1) where the robot needs to accurately and robustly identify and localise objects in order to manipulate them, communicate about them or reason on their geometric properties and relations. Critically, the object recognition technique needs to be suitable for actual experimental work, including field experiments: it must be reasonably easy to deploy the system in a range of dynamic human environments, without having to rely on expensive or cumbersome physical sensors, or expensive computation. We also take a short to medium horizon: not all techniques we evaluate are commonly available yet, but all have the potential to be robust implementations in the near future.

This paper tries to remedy a lack of information on deployment details in HRI contexts: many traditional assessments do not report on practical considerations. We need to take into account many different factors. For example, how robust is the detection and pose recognition when there are frequent changes to the environment, such as varying backgrounds or changing lighting conditions.

HAI'17, October 17–20, 2017, Bielefeld, Germany

ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

Figure 1. A close proximity interaction setup, typically found in human-robot interaction and cognitive robotics scenarios. While the number and nature of objects varies from one experiment to another, key scene characteristics are usually constant: relatively small objects (e.g. largest side being less than 10 cm), presence of occlusions, limited working space, and the presence of both textured and texture-less objects.

In this paper we compare across three families of techniques. The first is techniques that rely on 2D images, from which we track a selection of points. Back projection on these points allow the estimation of an object's 6D position. The second family of methods use 3D templates. 3D objects are compared against a known point cloud to find the position and orientation of an object. The final family relies on techniques that do not use traditional vision techniques, for example RFID technology.

Surveys on Object Detection

As a cornerstone of many robotic applications, research on object recognition and localisation has been reviewed in numerous past literature surveys. These surveys typically focus on one family of techniques or algorithms, typically using synthetic datasets to quantitatively compare the performances of the state of the art. We summarise hereafter the main findings for each of the localisation techniques.

Techniques based on 2D correspondences

When perceptual data consists of camera images, pre-stored templates of objects are often matched against the incoming video stream using 2D correspondence techniques. Li et al. [9] conducted a survey of visual feature detection. In the review

they categorise these techniques based on the fundamental principle by which they detect features, such as edge, blob or corner detection. Feature detection methods vary in performance based on the application context, but among them feature based techniques such as A-KAZE, ORB and SURF are popular in object recognition and tracking contexts [5].

Techniques based on 3D correspondences

The increased availability and popularity of 3D cameras has driven the need for 3D object matching techniques. Diez et al. [6] performed a qualitative review of 3D registration techniques, in which a mapping is made between 3D images or a 3D templates and an image. They specifically reviewed a variety of detectors and descriptors for 3D registration. Descriptors and detectors attempt to minimise the number of points required before using such brute force techniques to perform accurate identification. Note that while these are used to select salient points, they nearly always end up using iterative closest point (ICP) algorithms, which find corresponding points between a template and an unknown object. The more points that are used, the more accurate the detection is, but using more points has an exponential impact on computational requirements.

Non vision-based techniques

Many other reviews also focus on technologies not relying on visual perception. RFID can be used for coarse localisation, and has been shown to have an accuracy of a few centimetres [13]. The techniques used in their review are meant for localisation within a room, while our focus is on techniques that work on the scale of under a metre, for example localising objects on a tabletop. But reduced distance holds potential for increased accuracy, as objects are nearer to the RFID readers. Mautz [10] conducted a wide survey of a number of indoor positioning techniques for a range of applications. Most of the techniques reviewed are localisation for navigation, and are not practical for use in a tabletop situation. However, among the suitable methods identified for the accuracy we require for tabletop recognition was magnetic technology, which is able to reach millimetre levels of precision. Hostettler et al. [8] look at using Anoto positioning technology to localise a robot. They concluded that using a printed pattern that they are able to position a robot with high accuracy and with robustness to lighting and occlusion conditions, the technology was only restricted by the size and quality of the sheets that could be printed with the pattern.

Approach and Methodology

We compare a number of existing implementations of a wide range of techniques for object and pose detection. We chose a selection of implementations based on availability, ability to process in real-time and that could be considered representative of that technology. Each of these methods was compared against the following criteria:

1. **Degrees of Freedom**: The degrees of freedom that the method is able to measure (position and orientation).
2. **Detection Stability**: How stable was the method of detection. Would an object be lost even if nothing was happening, or were false positives generated.
3. **Rotation Invariance**: Is the method able to track the object when it is rotated.
4. **Distance Invariance**: How much does the distance of the object affect the tracking for that method.
5. **Environment Interference**: Is the method able to cope with changes to the background and lighting.
6. **Occlusion**: Can the method detect objects that are being occluded by other objects from the perspective of the robot.
7. **Practical Use**: Any additional notes such as extra equipment required that may affect the usability of the system in an experiment.

Each method is briefly described will be provided and an assessment based on the above criteria. A table of results provides a side by side comparison of each implementation. Finally we discuss and provide recommendations on each method.

ASSESSMENT OF OBJECT DETECTION METHODS

3D pose estimation from 2D images

These techniques use a standard 2D cameras. From this, image features are extracted that can be used to identify the object. These features can then be used to provide a 3D position by back projecting the 2D points to 3D reference points, using algorithms like 'perspective-n-point'(PnP) [7].

Fiducial markers

Fiducial markers look similar to 2D barcodes that can be printed out or displayed on a screen for detection. Each of these markers can be assigned an ID, and multiple markers can be attached to one object. The tags must be attached to a flat surface to allow them to be read and a pose estimation to be made. In this paper we used Chilitags [4].

Figure 2. Object with a fiducial marker, which allows it to be identified and tracked.

Changing the size of the markers can be used to affect the distance at which a marker can be read. The tracking is often lost while the objects are moving, but the objects are re-acquired quickly once they are set back down. The tags are highly susceptible to occlusion, a small amount is enough to lose tracking. The corners are particularly susceptible to

this. Because of the requirement to have a flat surface for the marker, irregularly shaped objects may be challenging to attach a marker to. We overcame this by using an additional cube attached below the object (see fig. 2).

Feature tracking

Three feature tracking methods were tested using the implementations provided by OpenCV[1]; SURF [2], A-KAZE [1] and ORB [12]. In each case an image is used as a target for the feature detection. These methods are classed as blob detection method, which look for areas of pixels that are similar to each other but contrast their surroundings. SURF approximates a Hessian matrix to rapidly find areas of interest. A-KAZE (Fig. 3) is an accelerated form of KAZE, using nonlinear diffusion filtering to detect areas. Normal Gaussian methods blur the edges of objects leading to reduced accuracy, but KAZE ensures that the blurring methods are adapted to natural boundaries. ORB uses a form of corner detection based on FAST as a keypoint detector and uses BRIEF as a descriptor.

All three of these methods struggle with changing backgrounds. A-KAZE and ORB are much more robust to the rotation of the object compared to SURF. While they all struggle with variations of distance, SURF is a marginally better than A-KAZE and ORB. While SURF is meant to be able to handle rotation, the objects in our evaluation have typically simple features and repetitive textures, which SURF struggles to handle. As these feature trackers determine which features they are going to track, they are not known in advance, this makes it more challenging to implement a PnP system for getting 3D coordinates.

Template matching

Template matching (Fig. 4), while a relatively old technique, was also considered; we tested using the implementation from OpenCV. An image is used as the target for template matching. This target image is then compared pixel by pixel against an image, and the strongest match is returned as a bounding box.

Multiple target images will be required per object to provide proper 6D pose estimation. Template matching is able to handle a range of distances well. It also has some tolerance to rotation. However it is not able to handle varying backgrounds.

Deep Learning

Deep learning relies on the training of Neural Networks on a dataset of pictures. These pictures contain the objects to be recognised and tracked with bounding boxes and classification. Here we used Faster R-CNN [11] to test Deep Learning. We used a pre-trained network[2] that was trained on the PASCAL VOC 2007 dataset. This technique provides accurate bounding boxes on target objects, though does sometimes lose an object (Fig 5).

It was also assessed if the pre-trained network would be able to detect an iconic representation of the animals that it had been trained on. It was however unable to (Fig 6), so training would be required on the specific objects to be used as part of the experimental setup.

[1] http://opencv.org/

[2] https://github.com/smallcorgi/Faster-RCNN_TF

This method only provides bounding boxes of the objects, but unlike template matching these cannot be compared against a known object. For instance the network used was trained to recognise sofas from a large number of sofas. But without knowing the dimensions of the sofa in a particular image we cannot tell if an object is large and far away, or small and nearby. This makes it difficult to provide a 6D estimation. This method also requires a lot of processing power to process in real time, and would likely need an additional computer, and not be run directly on a robot like the Nao.

3D pose estimation from 3D sensor data

In recent years RGBD cameras, which return 3D scene data in addition to a 2D image, have been widely used in HRI. The Microsoft Kinect technology or the Intel Realsense technology have proven particularly popular. Here we evaluate their software in the context of object localisation and pose reading. The techniques that look at are fairly computationally intensive, but not so intensive as to require more than a tablet or laptop to process the data. An external laptop could be added to a setup using a robot with limited processing power, such as a Nao.

For 3D object matching, we evaluate two different software implementation: "Tabletop" from the Object Recognition Kitchen (ORK)[3] implemented using ROS, and the Intel Realsense SDK [4].

Planar segmentation and iterative fitting

Tabletop uses planar segmentation to separate the surface of a table and segment objects that are on top. These objects are then compared to a database containing meshes of known objects using simple iterative fitting (related to ICP[3]). This method performed well with different object rotations and scales, and was unaffected by a change in background. However this method generated too many false positives to be considered a stable option for close proximity human-robot interaction scenarios.

Intel Realsense tracking

In the Intel Realsense SDK, Object Tracking (C++) for the SR300 was used. This method relies on having a 3D mesh of the object, which it then used for matching. During our investigation we were unable to specify the exact method used by the Intel SDK as it has not been published (see discussion session). In general the objects were recognised and tracked accurately, returning both position and orientation. However, objects were sometimes lost for no apparent reason and would need to be moved for them to be recognised again. This technique is able to handle a small amount of occlusion. At the time of writing the tracking feature for the SDK is only available for Windows, so implementing for integration with a robot may prove more challenging than other methods that can easily be integrated or already have a ROS implementation.

[3] http://wg-perception.github.io/object_recognition_core/index.html

[4] http://www.intel.co.uk/content/www/uk/en/architecture-and-technology/realsense-overview.html

Figure 3. Image showing target on the left, with an object being tracked through a rotation on the right using A-KAZE.

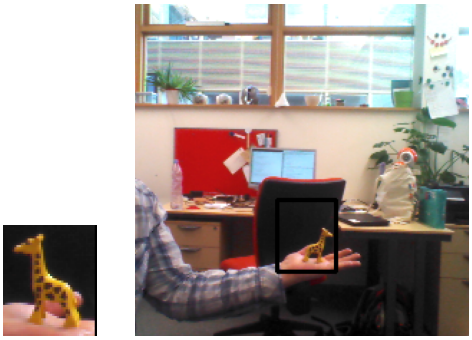

Figure 4. Left: Sample target image for template matching. Right: Test image showing a match with a flipped object.

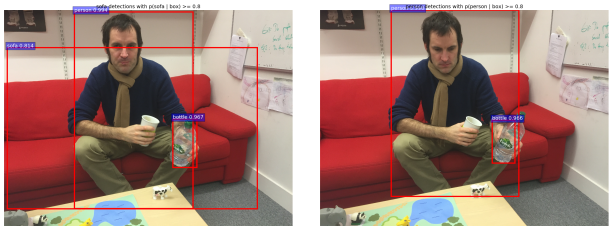

Figure 5. Images showing two similar pictures that have been processed by Faster R-CNN, on the left the sofa is detected, on the right the sofa is missed.

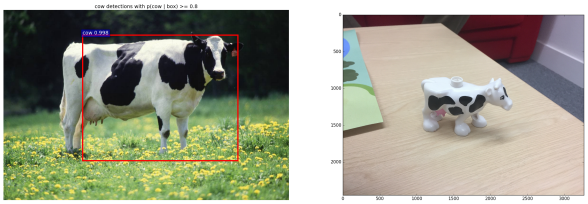

Figure 6. Images showing two pictures of cows, on the left a real cow that is detected by Faster R-CNN trained on the PASCAL VOC 2007 dataset, on the right an iconic toy cow that is missed.

Non-Vision Based Techniques

This section details methods that do not rely on the use of cameras, but instead the use of additional equipment.

Magnetic Field sensors

Magnetic Field sensors use one or more Hall effect sensors to read the position and orientation of a magnetic tag. Near Field Communication (NFC) tags can be used in addition to the magnet sensor to distinguish between different tags. We evaluated the GaussSense[5] solution, a small and affordable magnet sensor with a high degree of sensitivity. It is able to measure orientation and measures up to 3-4cm away from the sensor. It does however only cover a very small area. Many sensors would be required to cover a larger, the price may then become a consideration, with a 16x16cm board costing $350. GaussSense also requires the use of an Arduino to process the data received. The addition of the NFC sensor for object identification introduces a considerable amount of noise to pose reading. Also, due to fitting on top of the magnet sensor, the magnet and tag have to practically be in constant contact in order to be detected.

NFC solutions

Several NFC sensors a can be combined into an NFC array, allowing for detection over a larger area. We evaluated the ePawn[6] mat, an NFC sensor board covering a 32x32cm area. The ePawn mat, using a 2D matrix of sensors, can locate a tag with millimetre accuracy. Using two tags in an object allows the calculation of orientation in the plane of an object. Tags themselves are 2cm in diameter so would be able to fit on or inside small objects. Tags only really work well while in contact with the mat. The prototype we evaluated currently costs €1400.

Summary of Results

We provide a summary of results in Table 1. We give a based on the criteria defined in section 1.3.

DISCUSSION AND RECOMMENDATIONS

Of all the 2D vision based techniques fiducial markers were probably the most reliable. However its sensitivity to occlusion means it is unsuitable for a study where the objects are

[5]http://gausstoys.com/

[6]http://epawn.fr/

Method	Degrees of Freedom	Sta.	RInv.	DInv.	Env.	Occ.	Practical Use
2D w/ PnP							
Fiducial Markers	6D	Very High	Very High	High	Very High	Very Low	Markers on flat surfaces
A-KAZE	6D	Moderate	Very High	Low	Low	Moderate	
ORB	6D	Moderate	Very High	Low	Low	Moderate	
SURF	6D	Moderate	Moderate	Moderate	Low	Moderate	
Template Matching	6D	Very High	High	High	Low	Moderate	
Deep Learning (Faster R-CNN)	Planar	High	Very High	Very High	Very High	High	High Training Requirement
Depth Mapping							
ORK	6D	Very Low	High	High	High	Moderate	RGB-D Camera
Realsense SDK	6D	High	High	High	High	Moderate	RGB-D Camera
Non-Vision Based							
GaussSense	Planar w/ Rotation	Low	Very High	Very High	Very High	Very High	Sensor Board
ePawn	Planar w/ Rotation	Very High	Very High	Very High	Very High	Very High	Sensor Board

Table 1. Table showing a summary of the different object detection methods and their performance. Sta.: Detection Stability. RInv.: Rotation Invariance. DInv.: Distance Invariance. Env. Environment Interference. Occ.: Occlusion

Figure 7. Image showing the ePawn NFC Mat

frequently moved around by hand and placed behind other objects. Another challenge is often the attachment of fiducial markers onto objects: curved or irregular objects often prove challenging to attach the markers to. However, fiducial markers might bring benefits not offered by other technologies: the ease of displaying fiducial markers on a screen, or printing out markers, and the high accuracy it can provide, means that it is suitable for calibrating multiple cameras quickly in an experimental setup.

The feature tracking methods (A-KAZE, ORB and SURF) all have issues with dynamic backgrounds, which is an issue when the camera is not static or when subjects in the interaction are in view. It should be noted that the objects being used for this assessment were all relatively simple toys, which lacked rich texture. These methods may perform better on other, more textured, objects, but it may still require combining these methods with other algorithms to get a truly robust detection system.

Template matching, while relatively old, was among the most robust of the 2D methods. To provide a 6D pose estimation however this method will require a lot of templates to compare against. Therefore this method will not scale well with multiple objects. It may be better to use this method to increase the stability of other techniques where it could be used for foreground selection.

The Faster-RCNN that we tested can only provide a bounding box for our objects, this means we cannot get a full 6D pose estimation this technique alone. However its reliability means that it could be very useful as a foreground selection technique to be used in a pipeline with other methods. Recent research looks into using a CNN that is able to handle 3D pose estimation [14], but it is unlikely that a training set for specific experimental requirements exist as these networks are only just emerging. The process of generating the required training data and then training the network is a process that potentially requires months of work before being usable in an experiment.

The implementation of tabletop in ORK provided too many false positives to be feasible for use in our future studies. However we only tried one camera, the Intel SR300. Other hardware or updates to software drivers may increase performance. By making use of the planar segmentation part of the process it would be possible to subtract the background for use in other detection methods, causing this to no longer be an issue for those methods which struggle with varying backgrounds.

The Intel Realsense SDK performed better with a lot higher stability compared to ORK. However the issue where it would sometimes lose an object while not common is still enough to cause issues in a study. This however is probably the best method available if it is a requirement to track objects while they are being moved. We were unable to find the exact technique that Intel Realsense used, as it has not been published, but due to its performance it was still included in this review. It appears to identify contours in the object before we assume using ICP to match these points to the points of objects stored in the database.

None of the vision based techniques were fully capable of performing the required level of object recognition in a practical tabletop setting. However a pipeline of techniques has the potential to overcome the weaknesses that are shown with just a single method. For instance the 2D techniques could be used to provide a bounding box and classification of the

object, allowing a 3D technique to provide precision depth and pose information.

The GaussSense magnetic sensor performs well when tracking a single object. However an NFC module is required to be able to distinguish between multiple objects. For this reason it would be recommended to just use an NFC sensor when using multiple objects.

The ePawn NFC mat is probably the best method reviewed here for use in object recognition with tabletop manipulation. Its downside is that it cannot provide full 6D pose estimation, and the need for additional sensor equipment in the form of a RFID matrix. It is however suitable for many cases where objects need to be tracked, and potential interactions can be shaped around this limitation. NFC also has an advantage of being a known and reliable technique, as it used widely in contactless technology, such as debit cards and key fobs.

ACKNOWLEDGEMENTS

This work has been completed as part of the L2TOR project which is funded by the H2020 Framework Programme of the EC, grant number: 688014.

REFERENCES

1. Pablo F Alcantarilla and T Solutions. 2011. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell* 34, 7 (2011), 1281–1298.
2. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. Surf: Speeded up robust features. *Computer vision–ECCV 2006* (2006), 404–417.
3. Paul J Besl and Neil D McKay. 1992. Method for registration of 3-D shapes. In *Robotics-DL tentative*. International Society for Optics and Photonics, 586–606.
4. Quentin Bonnard, Séverin Lemaignan, Guillaume Zufferey, Andrea Mazzei, Sébastien Cuendet, Nan Li, Ayberk Özgür, and Pierre Dillenbourg. 2013. Chilitags 2: Robust Fiducial Markers for Augmented Reality and Robotics. (2013). `http://chili.epfl.ch/software`
5. Hsiang-Jen Chien, Chen-Chi Chuang, Chia-Yen Chen, and Reinhard Klette. 2016. When to use what feature? SIFT, SURF, ORB, or A-KAZE features for monocular visual odometry. In *Image and Vision Computing New Zealand (IVCNZ), 2016 International Conference on*. IEEE, 1–6.
6. Yago Diez, Ferran Roure, Xavier Lladó, and Joaquim Salvi. 2015. A qualitative review on 3D coarse registration methods. *ACM Computing Surveys (CSUR)* 47, 3 (2015), 45.
7. Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981), 381–395.
8. Lukas Hostettler, Ayberk Özgür, Séverin Lemaignan, Pierre Dillenbourg, and Francesco Mondada. 2016. Real-time high-accuracy 2D localization with structured patterns. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 4536–4543.
9. Yali Li, Shengjin Wang, Qi Tian, and Xiaoqing Ding. 2015. A survey of recent advances in visual feature detection. *Neurocomputing* 149 (2015), 736–751.
10. Rainer Mautz. 2012. Indoor positioning technologies. (2012).
11. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.
12. Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2564–2571.
13. T Sanpechuda and L Kovavisaruch. 2008. A review of RFID localization: Applications and techniques. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on*, Vol. 2. IEEE, 769–772.
14. Paul Wohlhart and Vincent Lepetit. 2015. Learning descriptors for object recognition and 3d pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3109–3118.