12-2020

Jointly optimizing sensing pipelines for multimodal mixed reality interaction

Darshana RATHNAYAKE
Singapore Management University, darshanakg@smu.edu.sg

Ashen DE SILVA
University of Moratuwa

Dasun PUWAKDANDAWA
University of Moratuwa

Lakmal MEEGAHAPOLA
Idiap Research Institute

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

See next page for additional authors

Citation

RATHNAYAKE, Darshana; DE SILVA, Ashen; PUWAKDANDAWA, Dasun; MEEGAHAPOLA, Lakmal; MISRA, Archan; and PERERA, Indika. Jointly optimizing sensing pipelines for multimodal mixed reality interaction. (2020). *2020 17th IEEE International Conference on Mobile Ad-Hoc and Smart Systems (MASS): Delhi, India, 10-13 December: Proceedings*. 309-317.

Available at: https://ink.library.smu.edu.sg/sis_research/5669

Author

Darshana RATHNAYAKE, Ashen DE SILVA, Dasun PUWAKDANDAWA, Lakmal MEEGAHAPOLA, Archan MISRA, and Indika PERERA

Jointly Optimizing Sensing Pipelines for Multimodal Mixed Reality Interaction

Darshana Rathnayake*, Ashen de Silva†, Dasun Puwakdandawa†, Lakmal Meegahapola‡§, Archan Misra*, and Indika Perera†
*Singapore Management University, Singapore
†University of Moratuwa, Sri Lanka
‡Idiap Research Institute, Switzerland
§École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract*—Natural human interactions for Mixed Reality Applications are overwhelmingly multimodal: humans communicate intent and instructions via a combination of visual, aural and gestural cues. However, supporting low-latency and accurate comprehension of such multimodal instructions (MMI), on resource-constrained wearable devices, remains an open challenge, especially as the state-of-the-art comprehension techniques for each individual modality increasingly utilize complex Deep Neural Network models. We demonstrate the possibility of overcoming the core limitation of latency–vs.–accuracy tradeoff by exploiting cross-modal dependencies–i.e., by compensating for the inferior performance of one model with an increased accuracy of more complex model of a different modality. We present a sensor fusion architecture that performs MMI comprehension in a quasi-synchronous fashion, by fusing visual, speech and gestural input. The architecture is reconfigurable and supports dynamic modification of the complexity of the data processing pipeline for each individual modality in response to contextual changes. Using a representative "classroom" context and a set of four common interaction primitives, we then demonstrate how the choices between low and high complexity models for each individual modality are coupled. In particular, we show that (a) a judicious combination of low and high complexity models across modalities can offer a dramatic 3-fold decrease in comprehension latency together with an increase ∼10-15% in accuracy, and (b) the right collective choice of models is *context dependent*, with the performance of some model combinations being significantly more sensitive to changes in scene context or choice of interaction.

***Index Terms*—sensor fusion, mixed reality, multimodal interactions**

I. INTRODUCTION

The rapid growth in the sensing capabilities of mobile and wearable devices, together with advances in machine learning-based perception, has spawned growing interest in the *Mixed Reality (MR)* applications across various domains [1], [2]. Broadly speaking, MR seeks to present users with a richer interaction and instructioning capability over a combination of both (a) synthetically-generated virtual objects, and (b) real-world objects located in the user's physical world. MR-based interaction now encompasses both unimodal [3] and multimodal [4] paradigms. In general, unimodal interaction (e.g., via touch-screen interactions) is easier to parse, but is less natural, for two reasons: (a) human instructioning and interaction is inherently multimodal, employing voice, vision, gestures and touch, and (b) the constraints of unimodal technologies often imply the use of unnaturally longer sequence of actions & instructions. Indeed, certain modalities may be better adapted for certain types of interaction tasks–for example, speech is better for describing object attributes such as type and color, whereas gestures are more powerful for disambiguating object location.

To enable truly interactive MR applications, it is important to support such multimodal instructioning and interaction capability in a real-time fashion–e.g., with a system response latency that does not exceed 1-2 secs. *Multi-modality* has thus emerged as a powerful paradigm for improving the effectiveness of real-time MR interaction, with a large body of work demonstrating that multi-modality can (a) enhance the overall expressiveness of interactions [5], (b) reduce overall interaction time [5], (c) increase interaction accuracy [6] and (d) support more natural interaction [6]. Moreover, multimodal interactions (**MMI**) are very effective in overcoming the natural ambiguity in intent expression that exists in unimodal systems. For example, in a study room scenario where several people are in the same MR realm, instructional ambiguity can arise due to imprecise perspectives [7], such as the speech command "look at *this* book" which involves the deictic expression *this*. Using further verbal elaboration of "*this*" to overcome this ambiguity is decidedly unnatural and requires more human effort. An intuitive and alternative disambiguation approach is to introduce another modality (such as a pointing gesture directed towards the object referred to by the "this" qualifier) [8]. This can also be viewed as providing more contextual evidence, with the gestural modality effectively providing additional context [9] in addition to linguistic cues.

Most unimodal sensing and comprehension pipelines, however, explicit a natural *latency vs. accuracy* tradeoff, that poses a challenge to our natural desire for low-latency, accurate interaction. This tradeoff problem has exacerbated with the recent explosion of Deep Neural Network (DNN) models for perception tasks in vision and speech [10], [11]–while such DNNs can increase accuracy significantly, they are often impossible to execute on resource-constrained wearable devices and must be offloaded to a GPU-rich, cloud infrastructure which imposes non-trivial additional network latency.

Our work in this paper explores the implications of such performance tradeoffs in MMI-based MR scenarios, with a view to developing techniques to "flatten this accuracy-vs-latency curve"— effectively, allowing wearable devices to exploit the enhanced accuracy of multimodality without suffering the penalty of significantly longer execution latency. More specifically, the work is driven by two observations:

- Current MMI designs perform algorithm selection of individual modalities in isolation, rather than jointly. In other words, the choice between a (more complex, higher-delay) DNN model vs. a (less complex, lower latency) alternative for visual object recognition is determined independently of similar choices made for other modalities. This is arguably sub-optimal as it ignores the possible interactions between different modalities, and the possibility that errors in one particular modality might be sufficiently compensated by improved capabilities of another modality.
- Besides not being optimized jointly, the current algorithm choices are also *not adaptive*–i.e., they typically tend to have a predefined fusion logic across different input modalities, independent of *context*. For example, for an MR application that combines audio and gestural cues, it will continue to execute a pre-defined fusion process even if, in certain situations (e.g., a very sparse, spatially well-separated layout of books in the aforementioned study room), the gestural input may have sufficient discriminative ability, making high verbal comprehension accuracy unnecessary.

Our contribution is to propose a sensor fusion architecture for such interactive MR applications, targeted to resource-constrained wearable devices, that can address the above-mentioned limitations. We design an architecture that is *configurable*–i.e., it allows the different sense-making pipelines for each individual modality to be configured or modified, to better match (a) different types of environmental context (e.g. classrooms, conference rooms, and industrial settings), and (b) varying performance characteristics of the underlying system (e.g. clock speed, RAM size and network latency). The architecture is designed to handle the *asynchronous* execution of each modality's perception pipeline. While our proposed architecture is generic, we specifically instantiate and evaluate it for fusing three common modalities: *vision, speech, and gestures*. Using state-of-the-art DNN-based pipelines for each modality, we experimentally show that the judicious *joint selection* of modality-specific perception pipelines helps to significantly improve the MMI latency-vs-accuracy tradeoff.

Key Contributions: This paper makes the following key contributions:

- *Configurable Multi-modal Fusion Architecture:* We propose an asynchronous sensor fusion approach for multimodal instruction comprehension (a key building block of MMI), which combines the inferences from different modalities in a flexible fashion, while allowing the sense-making pipelines of individual modalities to be adaptively modified in response to changing environmental or device context. The proposed architecture utilizes soft-synchronization via communication queues to coordinate inputs across different modalities, while accommodating the differences in processing latency for each modality-specific inferencing pipeline.
- *Demonstrate Coupled MMI Tradeoffs across Modalities:* By profiling a multiplicity of different inferencing models (varying from low to high complexity), we first characterize the accuracy vs. latency characteristics for each modality. We subsequently perform multimodal fusion, under these varying low←→high complexity alternatives, to quantify the overall impact on the performance of multimodal instruction comprehension. We show that the intelligent exploitation of compensatory synergies across different modes has significant impact: for example, (a) we can reduce latency by 10+% without any loss of accuracy, and (b) maintain high comprehension accuracy while lowering the CPU utilization by $\sim$7%.

We believe that our work provides tangible evidence of the benefits of a adaptive fusion model for pervasive multimodal instruction comprehension, and should motivate future work on developing *context-adaptive* MMI systems that can improve the efficiency of instruction comprehension by dynamically adapting the individual sense-making pipelines.

II. Related Work

We describe past work on both (a) mixed reality interactions and (b) the broader topic of multimodal sensor fusion.

A. Mixed Reality Interactions

Auditory Modality: Hughes et al. [12] emphasized the use of audio/speech as a modality to interact with non-linear MR environments, highlighting the importance of employing 3D surround technologies to enrich immersive experiences. Audio/speech interactions are vital to natural MR interactions, as it has been demonstrated [13] to be the primary modality for natural human communication. Use of different auditory-related technologies such as surround [14], binaural [15] and 3D [12], enrich the user experience of MR applications.

Gesture Modality: Gestures have been widely used to convey (or clarify) human intent, in a variety of MR applications. In particular, gestures such as pointing, grabbing, or stretching have been shown to increase the immersiveness of these systems [16]. Contact- and vision-based devices are the two main technologies in gesture recognition systems [17]. A physical interfacing device captures the interaction of the user in the systems that have employed a contact-based device. The latter approach captures gestures with different kinds of cameras, e.g., depth camera, stereo camera, web-camera, etc.

Visual Modality: In MR realms, visual modality is especially important as it displays both digitally synthesized and real-world physical objects, permitting seamless interaction with both object types [18]. A key factor for an effective MR experience is modeling a seamless boundary between the real and virtual objects, so that interacting with either presents no discernible difference [19]. Technologies such as see-through screens and opaque HMDs make the visual interactions possible in various application domains, e.g., education [20] and training & simulation [21].

B. *Multimodal Sensor Fusion*

There is a significantly large body of academic work in the past on the fusion of different sensing modalities to extract a variety of insights in different domains [22]. In [23], Sargin et al. applied Hidden Markov Models (HMM) to interpret the correlation between gestures (for example, head gestures) and speech. Asano et al. presented a Bayesian network-based approach for detecting speech events, using both 1) sounds captured by an array of microphones and 2) human gestures captured by video sensors [24]. A similar use case was investigated in [25], to perform audio localization and thereby separate multiple speakers from one other and from surrounding noise. Hara et al. [26] also present a similar solution, using Bayesian Networks to isolate and extract speech utterances by specific individuals involved in interactions with robotic agents. Hol et. al. [27] apply Kalman filters to fuse trajectories estimated from inertial and vision sensors, to accurately identify camera dynamics (such as orientation and position) for AR applications. These techniques are leveraged in applications presented in [28], which primarily fuse visual and auditory cues for a natural human-agent interaction.

III. Proposed Approach for MMI-driven Sensor Fusion

Sensor fusion methodologies can be categorized into multiple types [29]: (a) data fusion combines data from several sources; (b) feature fusion extracts features from various modalities and embeds them into a composite feature map for final inference; while (c) decision fusion involves fusing decisions generated by multiple independent inferencing models. To support the fusion of visual, speech, and gestural modalities (the 3 input modes most commonly used in natural interactions), we propose an architecture that consists of multiple independent subsystems (engines). The Vision, Aural, and Gestural engines perform modality-specific inferencing tasks, including object detection, speech recognition and text classification, and gesture recognition, respectively. The final outcome of a user interaction is based on a *decision* fusion of the output of these individual inferencing pipelines.

Figure 1 illustrates the functional architecture of our proposed fusion framework. The outputs of each modality-specific Engine are exchanged via a common Communication Queue, which performs soft timestamp matching of token sequences output by each engine before feeding them into the Fusion Engine. As mentioned, state-of-the-art methods for each Engine often include complex DNN models, thus presenting significant challenges for low-latency execution on resource-constrained devices. To study the interplay between these different Engines, we shall test our framework with several different combinations of Engine-specific models and identify the resulting tradeoff between overall system accuracy and latency. We now further describe the individual components (Engines) of our overall system.

Vision engine: The purpose of this subsystem is to recognize the objects with which the user interacts. This component takes raw images as input and feeds them to an object detection algorithm. Afterward, its predictions are relayed to the Fusion engine. Once the object is identified by the object detection algorithm, the objects are tracked across successive frames by an object tracker, namely KCF (Kernelized Correlation Filter) [30], which helps to reduce the computational overhead.

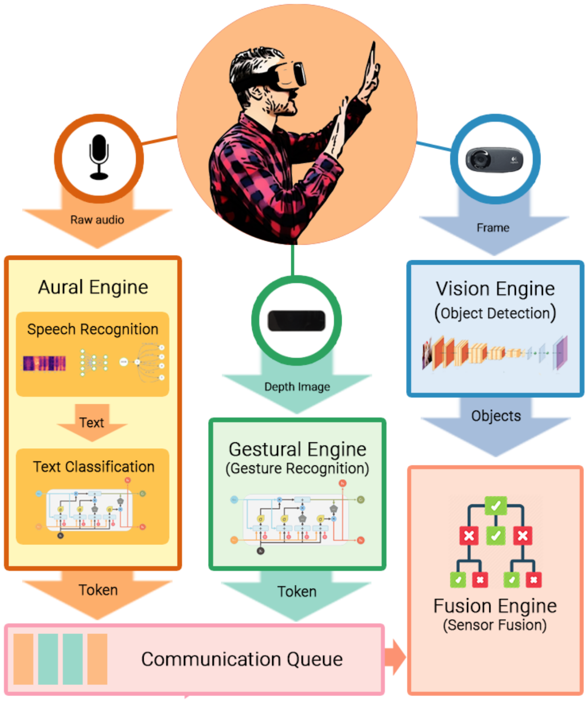


Fig. 1: High-level fusion architecture

Aural engine: This subsystem is responsible for transcribing users' utterances and classifying them to infer the user's action (or command). The speech modality is considered to be the primary input interaction modality of our system. An Automatic Speech Recognition (ASR) algorithm consumes the pre-processed captured audio signal before being fed into a text classification algorithm, which requires encoding text into a data structure appropriate for both training and inference. In order to extract the object with which user interacts, the utterance is passed through a hash table in which the keys are the names of objects and values are pre-defined identifiers. Multiple keys can have the same identifier since an object can have synonyms (e.g., mobile phone or smart phone). Afterward, a token consisting of the following three attributes is sent to Fusion engine: (1) *operation-id* which is an identifier for the intended operation, (2) *object-id* which is an identifier for the object, and (3) *multiplicity* which denotes whether or not the user mentioned multiple objects.

Gestural engine: Humans typically use hand gestures to supplement and disambiguate their vocal instructions. Accordingly, the gestural engine's goal is to classify gestures not only to identify interaction primitives, but to also support the refinement of issued voice commands. The vision-based devices are widely used in gesture recognition system as the contact-based methods can be detrimental to health [17].

Therefore, we choose to implement this engine with a vision-based device. This engine currently uses a Leap Motion Controller [1] to capture depth images of the hand, which are then processed by a classifier to identify the gesture performed by the user. Finally, the inferred gesture label is relayed to the Fusion engine via a token.

Fusion engine: This subsystem performs fusion of the three modalities mentioned above to identify the interaction performed. In our current framework, this fusion is triggered via the inputs of Aural and Gestural engines. Both of them will enqueue a token into a synchronized queue, with the tokens being subsequently consumed by the Fusion engine. The Fusion engine triggers the Object detector in the Vision engine to identify the relevant objects. The Fusion engine waits for 5 secs (this is configurable) for the output of the Vision engine. The objects returned by the Vision Engine (these objects have the same taxonomy as the objects identified by the Aural Engine) are compared with the object identifiers from the Aural and Gestural engines to identify a matching target. However, if the primary object detection algorithm fails to identify the corresponding object, the Fusion engine signals the Vision engine to switch to another higher-accuracy, high-latency alternative object detector. Once the corresponding (target) object has been determined, the Fusion engine signals the Vision engine to continue tracking this object.

IV. Test Environment and Implementation

To effectively demonstrate the capabilities of this architecture and study the tradeoffs between different instruction modalities, we decided to narrow down the implementation to fit a constrained scenario—one involving interactions in a *class/study room*. By carefully studying the operations and activities that would be possible in such a well-defined context, we would be able to first select a set of suitable modality-specific classification models and then study the individual and coupled accuracy-vs.-latency tradeoffs. In the selected scenario:

- The vision-based object detection pipeline is optimized to detect generic items typical of a classroom environment, including a laptop, keyboard, monitor, mobile phone, book, bottle and cups.
- The size of the vocabulary of the utterances, for the aural engine, is set to 188 (after filtering the stop words of a set of utterances that we gathered to describe and locate the aforementioned objects in different ways).
- The system focuses on recognizing 4 distinct gestures: (1) pointing, (2) zoom-in, (3) zoom-out, and (4) capture.

Our implementation supports 4 main operations as depicted in Table I.

Locate: "Locate" operation is performed when the user wants to search for an object in the real world, within the visual frame of the camera.

Zoom: The "Zoom" operation is a fusion of visual and gestural modalities, which together result in an enlargement of the video feed displayed on the wearable device. While most past work on MR applications involve human interactions with nearby objects, it is known that interactions with distant objects is more challenging [31]. To overcome instructional ambiguity for such distant objects, the Zoom gesture is used to enlarge an appropriate portion of the video frame.

Operation	**Interaction Modalities**	**Use-case**	**Output**
Locate	Visual and speech	Searching for an object in the real environment with an utterance.	Identified object highlighted in the video feed.
Describe	Visual, speech, and gestural	Learn more about an object with an utterance and/or a pointing gesture	Several properties of an object are superimposed on the display.
Zoom	Visual and gestural	Zoom into the frame to make interactions with smaller objects easier	The current video feed is enlarged.
Capture	Visual and gestural	Saving the current frame(s) for future reference	An instance of the video feed is saved as an image.

TABLE I: Fusion operations

Describe: In the "Describe" operation, a user can perform an action, using speech and/or gestural modalities, to elicit a descriptive feedback from the MR application. The execution branches depending on the semantics of the speech command (i.e. whether the command is explicit or ambiguous). Ambiguous *Describe* commands may come about with the use of the word *this*, in which case the system will automatically look for gestural inputs. For example, the user may utter *"What is this book"*, together with a corresponding pointing gesture. In this case, our system first identifies the inherent ambiguity in the referring expression "this", and then incorporates the sensed pointing gesture to identify the referred object. Another way in which a command can be deemed ambiguous depends on the speech command coupled with the environmental context. This entire fusion process is executed in three steps:

1) *Inferring speech command:* The token passed by Aural engine consists of the object identifier and its multiplicity. This operation can be implicitly ambiguous when the utterance includes *this* keyword, hence there are two types of *operation-ids* for the describe operations.
2) *Localizing the object:* The vision engine scans the current video feed to find the object(s) specified by the token output of Aural Engine. The detection of the hand is necessary in cases where the user has issued an ambiguous command, thus scanning for elements of two object classes, (a) the object and (b) the hand of the user, is needed. In such cases, the nearest (relevant) object is estimated by comparing its relative distance to the hand if a pointing gesture is detected (by examining the token passed by Gestural engine).
3) *Resolving the ambiguity:* The overall command can either be explicit operation (in which the user's intention is

[1] https://www.ultraleap.com/product/leap-motion-controller

executed as it is) or ambiguous (in which additional information is required to perform correct comprehension). Ambiguity in the speech command may occur due to mismatch between the multiplicity of an object mentioned in an utterance and the number of instances of the particular object detected by the Visual Engine. When the user mentions only one instance of an object but the Visual Engine detects multiple object instances, disambiguation of the utterance is performed with the aid of the pointing gesture.

Capture: The Capture command is a relatively trivial one, that causes the system to record and archive of the video feed for future reference.

A. Implementation Details

Our sensor fusion architecture is designed to be eventually deployed on wearable head-mounted display (HMD) smartglasses. For our current studies, which focus on exploring the tradeoff space across different modalities, we execute the developed framework on a desktop-class machine. All our experimental studies (results to be detailed in Section V-B) are conducted on a desktop machine with an Intel Core i5-8300H processor (a 4-core CPU which can operate upto 4 GHz), 16 GB of RAM, and one Nvidia GTX 1060 GPU with 6GB of VRAM.

Our code is implemented in Python. The various DNN components are implemented using Tensorflow[2], while the other non-DNN ML models are implemented with Scikit-learn[3]. Additionally, we use OpenCV[4] to execute the image processing tasks, NLTK[5] to preprocess the inputs for text classification, and PyKaldi[6] as a Python wrapper for Kaldi[7] to implement on-device speech recognition.

V. Experiments

Our overall system is designed to be modular and configurable–i.e., allow the specific instance of each individual component (e.g., speech recognition, object detection algorithms) to be modified, without affecting the overall fusion logic. To evaluate the interplay between the performance profiles of each component and the performance of the overall fusion engine, we adopt a "divide-and-conquer" approach [32], where we first evaluate each subsystem independently and then study the results of their coupled interaction.

A. Subsystem Evaluation

To understand the tradeoff between accuracy and computational overhead, we choose two representatives for each such sub-system/component: **H** represents a model instance with higher computational requirements and likely higher accuracy, while **L** denotes the model instance with low computation requirements.

Algorithm	**mAP**	**FPS**
SSD Inception v2	0.797	39
YOLO v3	0.963	12

TABLE II: Results obtained from object detection algorithms

1) Object detection: In general, two-stage detectors outperform one-stage detectors in terms of detection accuracy, but they are computationally intensive for wearable platforms [33]. Therefore, we chose two state-of-the-art one-stage object detection algorithms, namely, SSD [34] and YOLO v3 [35], and empirically evaluated their performance based on two parameters: *speed*, defined as the number of processed frames per second (FPS), and *accuracy*, defined by mean average precision (mAP) of object detection. We first employ transfer learning to re-train the detection models (which were initially trained for the *COCO* dataset [36]) with labeled data matching our class/study environment. We manually prepared a nearly balanced dataset comprising 2095 instances of 10 objects (laptop, keyboard, mouse, monitor, mobile phone, bottle, cup, pen, book, and hand); this dataset allows the object detector to be trained to recognize a sufficiently diverse, but finite number of object classes.

Table II depicts the performance of re-trained models. YOLO v3 is regarded as a high computational (**H**) model, as it has significantly higher latency, but outperforms the other algorithm in terms of accuracy. SSD Inception V2 model has been chosen as a low computational (**L**) model as it can achieve a higher throughput (higher number of frames per second), albeit at the expense of a lower accuracy.

2) Automatic Speech Recognition: The resources of a wearable device (the eventual target for our proposed MMI system) will be significantly constrained, especially for multimodal comprehension where multiple ML models will need to be executed concurrently. Accordingly, we explore various alternatives for ASR (automatic speech recognition) that exhibit moderate to high accuracy and relatively low computational overhead. The algorithms that were tested include Deepspeech[8], Kaldi[9], and IBM Watson Speech-to-text[10] on the cloud.

Table III tabulates our experimental results. Note that tools like IBM Watson offload the computation to the cloud and are thus suitable for resource-constrained devices, even though the additional round-trip propagation latency is likely to cause perceptible lag in an interactive system, especially under limited network bandwidth. Accordingly, we designate IBM Watson as the **L** model with low computational overhead, while Kaldi, which provides a relatively high accuracy (low word error rate (WER)) with relatively low transcription time but imposes higher local computation, is chosen as the **H** model. The use of Kaldi along with the extended ASpIRE model[11] provides good results with an acceptable latency of

[2] https://www.tensorflow.org

[3] https://scikit-learn.org/stable/

[4] https://opencv.org

[5] https://www.nltk.org

[6] https://pykaldi.github.io/

[7] https://kaldi-asr.org/

[8] https://github.com/mozilla/DeepSpeech

[9] https://kaldi-asr.org/

[10] https://www.ibm.com/cloud/watson-speech-to-text

[11] http://kaldi-asr.org/models/m1

	WER	Transcription Time
DeepSpeech	13.59%	2.9s
Kaldi	15.60%	1.5s
Modified Kaldi	12.50%	1.5s
IBM Watson Speech-to-Text	5.50%	2.5s

TABLE III: Performance of speech detection algorithms

Algorithm	Precision	Recall	F1-Score
SVM (Linear Kernel)	0.927	0.915	0.920
NN	0.910	0.486	0.647
LSTM	0.978	0.973	0.977

TABLE IV: Accuracies of text-classification models

processing. This model was modified by creating our own dictionary and language model using context specific grammar and merging them with the original dictionary and language models of ASpIRE. The resulting enhanced model increased the likelihood of recognizing the context-specific words for our scenario.

3) Text Classification: To support high-accuracy text classification, we developed a modified Long-Short Term Memory (LSTM) neural model as LSTM-based models have proven to be extremely popular for text analysis [37]. We adopted a word embedding layer which learns the multidimensional mapping of the input vector. Each test utterance is tokenized using word counts learned in the training process (the training process uses the data from the speech engine to learn a dictionary where the counts of words are stored) which are then fed into trained DNN. Confirming results reported previously, we observed that the sequential approach is superior to a simpler bag-of-words (BoW) approach. For example, for an utterance such as ``Show me the details of the book'', the LSTM model correctly classifies the instruction as Describe, whereas a BoW model would incorrectly label this as a Locate instruction. While our LSTM model is not perfect, we find that the accuracy is sufficiently high for our MMI scenarios, where utterances will be further *disambiguated* by additional complementary modalities. The model was trained with a dataset consisting of 700+ utterances belonging to 3 distinct operations: (1) locate, (2) describe and (3) no_op, with no_op representing a 'null' class to capture non-command utterances (which arise as the system is continuously listening to the user). As an alternative to the LSTM model, we also tested two alternative modes: (a) a vanilla neural network (NN) and (b) a Support Vector Machine (SVM) model as well. Table V tabulates the results. As the precision and recall of NN is much lower compared to the other two models, we then settle on the LSTM and SVM models as the **H** and **L** models, respectively.

4) Gesture recognition: We collected data from 20 users performing a total of 260 samples of 4 distinct gestures: pointing, zooming in and out, and capture. Using this dataset, we formulated two features, namely *single-finger* (euclidean distance between the palm center and the fingertip of each finger) and *double-finger* features (euclidean distances between adjacent fingertips) using an approach similar to what is presented in [38]. Based on the observed time taken to perform a gesture (mean=487ms, std=197ms) and the sampling frequency (145 Hz) of Leap Motion Controller, we aggregate features from 30 samples and feed this aggregated feature set into an LSTM-based classifier. As before, the LSTM model is compared with two other models: (a) a Support Vector Machine (SVM) and (b) a vanilla Neural Network (NN). Although the accuracy of LSTM and NN models (detailed in Table V) are roughly comparable, LSTM was chosen over NN because of its higher precision. LSTM and SVM models were similarly chosen as **H** and **L** model, respectively.

Algorithm	Precision	Recall	F1-Score	Latency
LSTM NN	0.78	0.77	0.77	2.2ms
NN	0.77	0.77	0.77	1.5ms
SVM (Linear kernel)	0.74	0.72	0.72	0.1ms

TABLE V: Performance of gesture classification models

B. Fusion-based System Evaluation

The performance of the sensor fusion architecture can be modeled as a function of 4 algorithms, namely, object detection (OD), automatic speech recognition (ASR), text classification (TC), and gesture recognition (GR). We chose two models (**H** and **L**) for each algorithm and evaluated the latency of the system (time taken to display the output for 5 frames from the start of an action). If the system utilizes a cloud-based service, the latency figures include the additional overhead of network communication. To compute latency, two fusion operations, Locate (O_1) and Describe (O_2), are considered as they require multimodal interactions, whereas Zoom and Capture operations are determined solely via gesture recognition. Each operation was repeated 30 times– i.e, the study involved 30 different utterances with different objects, for both the Describe and Locate operations).

Because the comprehension accuracy is a function of not just the choice of algorithms but also the *environmental context*, we evaluated the performance of different fusion strategies under two distinct contexts: (a) **Context A** in which 5 objects (4 large objects and 1 small objects) are placed at a distance of 1 meter from the observer; and (b) **Context B** in which 5 different objects (3 large and 2 small objects) are placed at a distance of 2 meters. To understand the impact of different algorithm choices (*H* vs. *L*) in detail, we also measured the resource consumption in terms of usage of RAM, VRAM, CPU and GPU resources.

Table VI tabulates the observed latency and accuracy values, for $2^4 = 16$ different combinations of the individual model pipelines, as well as the corresponding consumption of computing resources. Note that the Locate primitive ($O1$) does not involve the use of a gestural input; accordingly, the latency and accuracy numbers for $O1$ are evaluated only for $2^3 = 8$ distinct combinations. We make the following key initial observations:

- The overall latency of the fusion process is roughly constant across different contexts and operator primitives, given a specific combination of (OD, ASR, TC, GC) models. However, the latency exhibits significant variation across

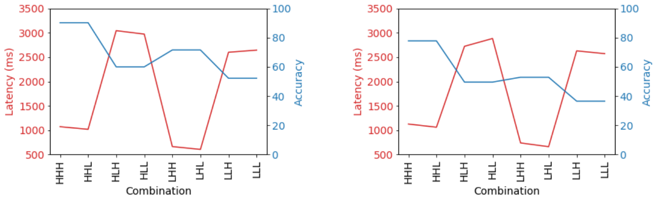
Algorithm				Context A				Context B				Resource Consumption			
				Latency (msecs)		Accuracy (%)		Latency (msecs)		Accuracy (%)					
OD	**ASR**	**TC**	**GR**	**O1 (Loc.)**	**O2 (Desc.)**	**O1 (Loc.)**	**O2 (Desc.)**	**O1 (Loc.)**	**O2 (Desc.)**	**O1 (Loc.)**	**O2 (Desc.)**	**RAM**	**CPU**	**GPU**	**VRAM**
H	H	H	H	1209	1276	90.2	87.0	1255	1290	77.8	72.0	6217	38	32	5657
H	H	H	L		1265		77.3		1282		64.0	6155	31	31	5597
H	H	L	H	1152	1187	90.2	87.0	1187	1194	77.8	72.0	5949	31	31	5597
H	H	L	L		1180		77.3		1185		64.0	5890	31	31	5753
H	L	H	H	3171	3296	60.0	75.0	2864	3089	49.6	60.0	5195	35	23	5658
H	L	H	L		3253		66.7		3140		53.3	5110	34	24	5597
H	L	L	H	3102	3190	60.0	75.0	3048	3149	49.6	60.0	4904	35	24	5597
H	L	L	L		3211		66.7		3007		53.3	4854	34	19	5755
L	H	H	H	846	919	71.5	69.0	860	877	52.9	51.0	3479	53	17	3479
L	H	H	L		916		61.3		859		45.3	3411	53	17	3617
L	H	L	H	789	844	71.5	69.0	782	784	52.9	51.0	3197	53	18	3617
L	H	L	L		834		61.3		780		45.3.	3170	51	17	3556
L	L	H	H	2727	3001	52.2	60.0	2656	2584	36.5	42.0	2444	37	15	3678
L	L	H	L		3065		53.3		2665		37.3	2383	37	15	3617
L	L	L	H	2774	3007	52.2	60.0	2664	2602	36.5	42.0	2172	39	18	3617
L	L	L	L		2745		53.3		2552		37.3	2116	39	18	3554

TABLE VI: Performance comparison of different model combinations (OD[L] - SSD, OD[H] - YOLO, ASR[L] - IBM Watson, ASR[H] - Kaldi, TC[L] - SVM, TC[H] - LSTM, GR[L] - SVM, GR[H] - LSTM)

different model choices–e.g., for $O1$, the latency can vary from approx. 850 msecs (for the (L,H,H) tuple) to over 2700 msecs (for the (L,L,L)) combination. Interestingly, the choice of $ASR = L$ results in a significant increase in computation latency, due to the added network latency of interacting with the cloud-based ASR engine.

- While there is still an overall trend of a tradeoff between overall complexity and accuracy, the tradeoff is not linear in the performance of individual components, but varies due to the coupling across the different modalities. For example, for the case of the $O2$(=Describe) primitive, the accuracy numbers for (H,H,L,H) and (H,H,H,H) are both approximate equal, reaching a value of 87% for context A and 72% for context B, respectively. The choice of a lower complexity TC model does, however, result in significantly lower resource overhead–e.g., the CPU utilization reduces by $\sim$7%– and a nearly 100msec reduction in latency. Clearly, in this case, $TC = L$ is a preferred choice, as it results in lower processing overhead and latency without any concomitant negative impact on system accuracy.

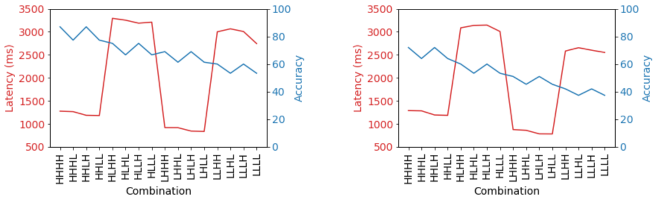
Latency vs. Accuracy for different Contexts: We next study the impact of different environmental contexts on this overall latency-vs.-accuracy tradeoff. Figures 2 and 3 plot the variations in accuracy and latency for both contexts A and B, for the cases of the Locate & Describe operators, respectively. We can see that the relative performance is *context dependent*. For example, context B (which has a more distant view of objects) observes a steeper drop in accuracy as we progressively select less complex models. In particular, with the increased distance to the observed objects (context B), the accuracy of the SSD (OD=L) algorithm becomes significantly inferior to that obtained by the use of YOLO (OD=H). On the other hand, both $TC = L$ and $TC = H$ models show similar performance in terms of accuracy across different contexts, implying that the SVM-based classifier (TC=L) model is preferable for both contexts. The overall latency variation also differs across contexts–e.g., for the Locate operator, the (H,L,L) combination incurs higher latency for Context B but lower latency for Context A, compared to the (H,L,H) combination.



(a) Context A

(b) Context B

Fig. 2: Tradeoff between accuracy and latency in **locate** operation



(a) Context A

(b) Context B

Fig. 3: Tradeoff between accuracy and latency in **describe** operation

Latency vs. Accuracy for Different Operators: Figures 2 and 3 also help us study and evaluate the impact of different model choices, for different interaction operations. We see that the Describe operation experiences a higher latency (almost 100 msecs higher for the (H,L,H,H) combination), compared to the latency experienced by an identical model applied to the Locate operation. Moreover, there are significant differences in the resulting accuracy. In general, for an identical selection of model choices, the Describe operation has approx. 10-15% lower accuracy. However, most interestingly, this degradation in accuracy is not universal across all model choices. Note, for example, that (H,L,L,H) has approx. 15% higher accuracy for the Describe operation, compared to the corresponding accuracy for the Locate operator. In this case, the additional information provided by the recognition of the pointing gesture helps to isolate the target object much more precisely, without a significant increase in the overall processing latency.

Effect of resource utilization on latency: From the results obtained and plotted in Table VI, it is clear that the choice of different models can result in very different profiles of resource consumption as well. For example, we observe that for the object recognition pipeline, the YOLO model (OD=H) utilize significantly more resources (49% in RAM, 35% in GPU, and 36% in VRAM) compared to SSD (OD=L), but with lower CPU usage (35%). Consequently, the choice of SSD seems to reduce the latency by 17 - 22%. Similarly, the use of a cloud offloading for ASR results in a significantly higher processing latency (approx. 180-200% higher), due to the network latency. However, the resource consumption in terms of RAM (24%) and CPU (10%) are slightly lower compared to the alternate on-device implementation.

Key Takeaways: Our detailed performance results demonstrates the absence of *an universally superior* combination of models–the choice of models and the consequent accuracy vs. latency tradeoff depends on the complexity of the environmental context and the interaction primitive being performed. For the specific *Study Room* setting that we analyzed, 3 combinations emerge as suitable candidates under different contexts: (1) `HHLH`, a combination that imposes high CPU and GPU overhead, but is suitable for scenarios where high accuracy and low latency are essential, (2) `LHLH`, which is a CPU-intensive (high RAM and CPU usage and low GPU usage) combination and is suitable for contexts where latency is prioritized over the accuracy, and (3) `LLLH` which is especially appropriate for situations where the accuracy and latency constraints are less stringent.

VI. Discussion

While our work establishes the non-obvious coupling between the computation pipelines of different modalities (and the resulting latency vs. accuracy tradeoffs), there are, however several issues that need additional exploration.

Automatic Inference of Context (Scene Complexity): Our work shows that the right choice of model combinations is *context dependent*. However, for practical application of context-based model selection, the context itself must first be determined. Context determination itself (e.g., whether the current physical scene is cluttered or only has a few objects) requires additional computation, and the benefit of context-dependent model adaptation may be negated if the context determination process itself has high complexity and latency. Accordingly, developing low-complexity, lightweight complexity estimators is an important prerequisite for our proposed operational model.

Incorporating History in the Interaction Comprehension Pipeline: Our current experimentation settings and results utilize a *memoryless* interaction model, where each interaction primitive is analyzed and estimated in isolation. In real environments, interaction primitives are obviously temporally correlated–e.g., a Zoom operation is likely to be followed by a Describe operation. The incorporation of such priors (likelihoods) is likely to further improve the process of dynamic model selection.

Incorporating Additional Metrics in the Overall Tradeoff: Our current evaluation has concentrated primarily on the accuracy vs. latency tradeoff. While these are important system parameters, the choice of models may need to consider additional metrics as well. For example, the energy overhead is likely to be another important constraint, especially when battery-constrained wearable devices are used continuously (e.g., in smart factories or warehouses). As our results demonstrate, latency itself may be distinct from resource consumption–e.g., the cloud-based ASR model has higher latency due to network overheads, but consumes lower CPU and RAM resources.

VII. Conclusion

In this paper, we focused on the problem of developing a suitable multimodal sensing and fusion framework to support natural interactivity for mixed reality (MR) applications. Our work is motivated by the rapid growth in use of DNN-based complex models for sensing-based comprehension and perception tasks, and the challenge of executing them on resource-limited pervasive devices. We demonstrate that multimodal fusion may offer a way to reduce the reliance on such complex DNN models—high accuracy estimation on one or more sensing modalities may allow the overall accuracy of fusion to remain unaffected, even if the other sensing modalities utilize less computationally-complex processing pipelines.

To support such a model, we first present a configurable and extensible multimodal sensor fusion framework, where the models for individual sensing modalities can be dynamically modified without affecting the the overall fusion process. We then study a specific instantiation of this framework, applied to a study room setting, which utilizes vision, aural and gestural input to support four different interaction primitives. By experimental studies, we quantify the performance of two different choices (low and high complexity) for each model, first in isolation and then jointly (after sensor fusion). Our studies reveal the choice of an appropriate *combination* of modality-specific models is non-trivial: depending on the models chosen, the instruction comprehension latency can vary from $\sim$700-3200 msecs and the accuracy between $\sim$26-90%. Moreover, the optimal model combination depends both on the scene complexity and the specific interaction primitive. On finer inspection, we see that (a) most optimal combinations involve the use of a higher complexity Gesture Recognizer (GR) and a low-complexity Text Classifier (TC), (b) the choice of a high vs. low complexity Object Detector (OD) is dependent on the scene complexity and (c) a higher-complexity automatic speech recognizer (ASR) is usually preferred over a lower-complexity cloud-based model, except for extremely resource-poor devices. We hope that our work motivates more careful focus on the problem of adaptive model selection for low-latency, interactive instruction comprehension.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative, and Ministry of Education

(MoE), Singapore under AcRF Tier-1 grant 19-C220-SMU-008. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

[1] G. Lampropoulos, E. Keramopoulos, and K. Diamantaras, "Enhancing the functionality of augmented reality using deep learning, semantic web and knowledge graphs: A review," *Visual Informatics*, vol. 4, no. 1, pp. 32 – 42, 2020.

[2] L. Meegahapola and I. Perera, "Enhanced in-store shopping experience through smart phone based mixed reality application," in *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 2017, pp. 1–8.

[3] K. Saroha, S. Sharma, and G. Bhatia, "Human computer interaction: An intellectual approach," *International Journal of Computer Science and Management Studies*, vol. 11, 08 2011.

[4] S. Jamali, M. F. Shiratuddin, and K. W. Wong, "An overview of mobile-augmented reality in higher education," 2014.

[5] M. Lee, M. Billinghurst, W. Baek, R. Green, and W. Woo, "A usability study of multimodal input in an augmented reality environment," *Virtual Reality*, vol. 17, no. 4, pp. 293–305, Nov. 2013.

[6] S. S. M. Nizam, R. Z. Abidin, N. C. Hashim, M. C. Lam, H. Arshad, and N. A. A. Majid, "A Review of Multimodal Interaction Technique in Augmented Reality Environment," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 4-2, pp. 1460–1469–1469, Sep. 2018.

[7] R. Scalise, S. Li, H. Admoni, S. Rosenthal, and S. S. Srinivasa, "Natural language instructions for human–robot collaborative manipulation," *The International Journal of Robotics Research*, vol. 37, no. 6, pp. 558–565, 2018.

[8] T. Williams, "A framework for robot-generated mixed-reality deixis," in *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 2018.

[9] K. Nagao and J. Rekimoto, "Ubiquitous talker: Spoken language interaction with real world objects," 1995.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," *arXiv e-prints*, p. arXiv:1412.5567, Dec 2014.

[12] C. E. Hughes, C. B. Stapleton, D. E. Hughes, and E. M. Smith, "Mixed reality in education, entertainment, and training," *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 24–30, Nov 2005.

[13] M. Cavazza, F. Charles, S. J. Mead, O. Martin, X. Marichal, and A. Nandi, "Multimodal acting in mixed reality interactive storytelling," *IEEE MultiMedia*, vol. 11, no. 3, pp. 30–39, 2004.

[14] Zhiyun Li, R. Duraiswami, and L. S. Davis, "Recording and reproducing high order surround auditory scenes for mixed and augmented reality," in *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nov 2004, pp. 240–249.

[15] S. Yao, "Headphone-based immersive audio for virtual reality headsets," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 3, pp. 300–308, August 2017.

[16] J. Y. Lee, G. W. Rhee, and D. W. Seo, "Hand gesture-based tangible interactions for manipulating virtual objects in a mixed reality environment," *The International Journal of Advanced Manufacturing Technology*, vol. 51, no. 9-12, pp. 1069–1082, 2010.

[17] H. S. Hasan and S. A. Kareem, "Human computer interaction for vision based hand gesture recognition: A survey," in *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, Nov 2012, p. 55–60.

[18] C. Coutrix and L. Nigay, "Mixed reality: A model of mixed interaction," in *Proceedings of the Working Conference on Advanced Visual Interfaces*, ser. AVI '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 43–50.

[19] M. Billinghurst, H. Kato, and I. Poupyrev, "The magicbook—moving seamlessly between reality and virtuality," *IEEE Comput. Graph. Appl.*, vol. 21, no. 3, p. 6–8, May 2001.

[20] F. Liarokapis, P. Petridis, P. F. Lister, M. White *et al.*, "Multimedia augmented reality interface for e-learning (marie)," *World Transactions on Engineering and Technology Education*, vol. 1, no. 2, pp. 173–176, 2002.

[21] J. Stevens, P. Kincaid, and R. Sottilare, "Visual modality research in virtual and mixed reality simulation," *The Journal of Defense Modeling and Simulation*, vol. 12, no. 4, pp. 519–537, 2015.

[22] K. Jayarajah, V. Subbaraju, N. Athaide, L. Meeghapola, A. Tan, and A. Misra, "Can multimodal sensing detect and localize transient events?" in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX*, M. A. Kolodny, D. M. Wiegmann, and T. Pham, Eds., vol. 10635, International Society for Optics and Photonics. SPIE, 2018, pp. 331 – 343. [Online]. Available: https://doi.org/10.1117/12.2322858

[23] M. Sargin, O. Aran, A. Karpov, F. Ofli, Y. Yasinnik, S. Wilson, E. Erzin, Y. Yemez, and A. Tekalp, "Combined gesture-speech analysis and speech driven gesture synthesis," *2012 IEEE International Conference on Multimedia and Expo*, vol. 0, pp. 893–896, 07 2006.

[24] F. Asano, Y. Motomura, and S. Nakamura, "Fusion of audio and video information for detecting speech events," in *Sixth International Conference of Information Fusion, 2003. Proceedings of the*, vol. 1, July 2003, pp. 386–393.

[25] V. M. Trifa, A. Koene, J. Moren, and G. Cheng, "Real-time acoustic source localization in noisy environments for human-robot multimodal interaction," in *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, Aug 2007, pp. 393–398.

[26] I. Hara, F. Asano, H. Asoh, J. Ogata, N. Ichimura, Y. Kawai, F. Kanehiro, H. Hirukawa, and K. Yamamoto, "Robust speech interface based on audio and video information fusion for humanoid hrp-2," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2404–2410 vol.3.

[27] J. D. Hol, T. B. Schon, F. Gustafsson, and P. J. Slycke, "Sensor fusion for augmented reality," in *2006 9th International Conference on Information Fusion*, 2006, pp. 1–6.

[28] S. Shafer, A. Stentz, and C. Thorpe, "An architecture for sensor fusion in a mobile robot," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, April 1986, pp. 2002–2011.

[29] R. Sharma, V. I. Pavlovic, and T. S. Huang, "Toward multimodal human-computer interface," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 853–869, 1998.

[30] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, p. 583–596, Mar 2015.

[31] M. Whitlock, E. Harnner, J. R. Brubaker, S. Kane, and D. A. Szafir, "Interacting with Distant Objects in Augmented Reality," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, Mar. 2018, pp. 41–48.

[32] B. Dumas, R. Ingold, and D. Lalanne, "Benchmarking fusion engines of multimodal interactive systems," in *Proceedings of the 2009 international conference on Multimodal interfaces - ICMI-MLMI '09*. ACM Press, 2009, p. 169.

[33] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, p. 261–318, Feb 2020.

[34] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[35] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[36] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.

[37] C. Li, G. Zhan, and Z. Li, "News text classification based on improved bi-lstm-cnn," in *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*. IEEE, 2018, pp. 890–893.

[38] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," *IEEE Signal Processing Letters*, vol. 23, no. 9, pp. 1188–1192, Sep. 2016.