Hyperion: A Wearable Augmented Reality System for Text Extraction and Manipulation in the Air

Dimitris Chatzopoulos
Systems and Media lab
The Hong Kong University of Science and Technology

Carlos Bermejo
Systems and Media lab
The Hong Kong University of Science and Technology

Zhanpeng Huang
Systems and Media lab
The Hong Kong University of Science and Technology

Arailym Butabayeva
Systems and Media lab
The Hong Kong University of Science and Technology

Rui Zheng
Systems and Media lab
The Hong Kong University of Science and Technology

Morteza Golkarifard
Sharif University of Technology

Pan Hui
Systems and Media lab
The Hong Kong University of Science and Technology

ABSTRACT

We develop *Hyperion* a Wearable Augmented Reality (WAR) system based on Google Glass to access text information in the ambient environment. *Hyperion* is able to retrieve text content from users' current view and deliver the content to them in different ways according to their context. We design four work modalities for different situations that mobile users encounter in their daily activities. In addition, user interaction interfaces are provided to adapt to different application scenarios. Although Google Glass may be constrained by its poor computational capabilities and its limited battery capacity, we utilize code-level offloading to companion mobile devices to improve the runtime performance and the sustainability of WAR applications. System experiments show that *Hyperion* improves users ability to be aware of text information around them. Our prototype indicates promising potential of converging WAR technology and wearable devices such as Google Glass to improve people's daily activities.

KEYWORDS

AR, smart-glass, Google Glass, wearable, WAR, offloading

ACM Reference format:
Dimitris Chatzopoulos, Carlos Bermejo, Zhanpeng Huang, Arailym Butabayeva, Rui Zheng, Morteza Golkarifard, and Pan Hui. 2017. Hyperion: A Wearable Augmented Reality System for Text Extraction and Manipulation in the Air. In *Proceedings of ACM Multimedia Systems, Taipei, Taiwan, June 2017 (MMSys'17),* 12 pages.
https://doi.org/http://dx.doi.org/10.1145/3083187.3084017

MMSys'17, June 2017, Taipei, Taiwan

ACM ISBN 978-1-4503-5002-0/17/06…$15.00
https://doi.org/http://dx.doi.org/10.1145/3083187.3084017

1 INTRODUCTION

Augmented Reality (AR) is the research area that deals with the class of problems of supplementing the reality with virtual objects. Mobile Augmented Reality (MAR) is the research area that deals with the class of problems of supplement the reality of mobile users with virtual objects [20, 23, 26]. The amount of virtual objects can be huge and the selection of the most suitable ones [17, 45] is not trivial due to the limited screen sizes of the smartphones, or the wearable mobile devices like Google Glass. In the case of a mobile and dynamic environment where the goal is to augment the reality of mobile users who are experiencing the world through wearable devices, the research field is called Wearable Augmented Reality (WAR). The most popular approaches in the reality augmentation are vision based since over 80% of information received by human comes from eyes [29]. However, this augmentation has to be triggered by existing information of many types in the ambient environment. One of the most important problems in WAR systems is the information extraction from ambient environment using the sensors of the mobile user. Popular applications of vision based WAR lie in the area of tourism [28], gaming [31] and health. In the case of tourism, WAR applications can assist mobile users on browsing on countries that are not using the same language or provide information about attractions. While in the case of gaming, WAR can provide more realistic environments. According to a report from the WHO [43], 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 have low vision. Compared to the optical methods, most mobile apps are based on digital amplification. Some apps zoom in images large enough so that the visually impaired can see it. Magnifying Glass [32] is a such type app that magnifies what a user is looking at through the camera. Indifferent from these application specific proposals, authors of [8] proposed a WAR recommendation system that selects which virtual objects to present on the limited-size screens of the wearable devices. However, the authors assumed the existence of a cloud database that has stored all the potential virtual objects and they didn't explain how the information is extracted for their system.

The main challenge of information extraction, and more specifically Optical Character Recognition (OCR), is the computational needs of the existing computer vision methods. But the capabilities of modern mobile devices have become enough to fulfill the resource requirements. However, it is important to fulfill the users' quality of experience during the use of such applications. In several situations, where hand free movements are necessary (i.e., driving, walking, carrying shopping bags), mobile users are not willing to see through the screen of their mobile phone the Augmented Reality because this can be inconvenient. Wearable devices such as smart-glasses are now filling this gap but, unfortunately, their computing resources are inferior to mobile phones (i.e., dual-core CPU on Google Glass versus Octa-core on modern smartphones). On the other hand, they have multiple network interfaces and are able to connect in parallel with more than one network devices. In addition, the mature computation offloading solutions at the edge of the network (i.e., cloudlets *data center in a box* [5] [39] [34]), named *fog computing* ([4]) offers a way to execute a demanding task without compromising the Quality of Experience (QoE) that is cased by the network delay.

The most popular wearable AR device is Google Glass [44], although there exist other alternatives such us Microsoft HoloLens [1], and MAD Gaze [2]. The Microsoft device provides a wide range of sensors to track the users environment and its changes, overlaying complex AR graphics that merge with the real objects. It offers great capabilities for visualizing holograms in the real environment, and offers gaze tracking based on the users head movements. However, due to device size and weight, it is targeted for indoor environments. MAD Gaze is based on the Android Platform, and it is targeted as a cheaper version of Google Glass, with similar features, design and hardware. Therefore, any system developed for Google Glass devices can be used in MAD Gaze due to its platform implementation, this cheap smart-glass alternative can run software previously developed for Google Glass. Compared with other mobile platforms, Google Glass is designed to cooperate with human eyes. Previous mobile applications were implemented for mobile devices and they require from the users to hold the devices to aim at scene, while Google Glass based applications detect whatever users are looking in a hands-free user experience. By converging Google Glass with WAR technology, mobile users are able to access additional information on demand without leaving from their current activities. WAR applications have been used to assist human operations in many directions. One example is the work of [46] where authors present a technology for machine maintenance assistance and another one is the work of Ugulino and Fuks [38] who support indoor landmark identification in the context of blind pedestrians mobility.

As a motivational scenario, we consider a Japanese tourist, name Eiko, in London. Since most of the signs in London are in English Eiko has to try to translate most of them in order to navigate properly and see most of the monuments. However, it is not convenient for Eiko to use her smartphone all the time and not enjoy the scenery. A WAR application implemented in Google Glasses is able to help Eiko, and any tourist in general, to navigate in a country with different spoken language.

[1] https://www.microsoft.com/microsoft-hololens/en-us
[2] http://www.madgaze.com

In this paper, we develop *Hyperion* (Section 3), a WAR system based on Google Glass to extract text information from the ambient environment. *Hyperion* overlays text information directly on the current view so that users see both the information and real world in real time. *Hyperion*, in Greek mythology, was one of the twelve titans and was the first to understand, by diligent attention and observation, the movement of both the sun and the moon and the other stars, and the seasons as well. We gave this name to our system because we want it to understand the current view of the user in a sense that it is able extract the surrounding text.

In comparison with previous works (Section 2), our proposal features an efficient, deployed and adaptive system to extract text information from the user surroundings view. By using Google Glass, the system supplies on-demand services with hand-free user experience. *Hyperion* targets to help the mobile users by answering "what's on the screen?" in classroom or "which road do I take?" in the drive. Work modalities and user interface are dedicatedly designed to guarantee friendly user experience. The systems does not only adapts to the environment it also includes some predefined by user situations that provides different functionality for each one. We designed four different modes that we think can be useful for the user: *(i)* Explore, *(ii)* Reading, *(iii)* Meeting and *(iv)* Driving. Each mode offers a set of inputs and outputs that maximize the users' convenience. The possible inputs are *(i)* the touchpad available on Google Glass, *(ii)* the voice command and *(iii)* hand gesture while the available outputs are via *(i)* Google Glass' display and *(ii)* speech synthesis. Experiment results (Section 4 are discussed to gain insight into advantages and limitations of the system, which gives directions for further improvements. In more detail, after we evaluated the technical performance of *Hyperion* in terms of execution time and energy needs, we recruit a few students, with visual impairment, from our universities to use our prototype implementation and evaluate its practicality and we also survey them via an extended technology acceptance model (TAM) [16]. It is worth mentioning that we do not focus on devising new wearable device or designing new algorithms, but try to integrate existing algorithms with our Google Glass based *Hyperion* to provide to mobile users access text information that were not accessible before. Then, a short discussion about the advantages and disadvantages of our proposed system in real world situations are described in Section . Furthermore, we depict the future target users of *Hyperion*, the visually impaired. We use the participatory design guideline with an initial survey (Section 5 and future improvements such as more accurate gesture recognition. Finally, in Section 6 we conclude the work of our system.

2 RELATED WORK

OCR based applications are developed to extract text from images and deliver it to users. Mobile OCR [35] converts text information in an image into regular text and is compatible with multiple languages while SayText [30] uses the built-in camera to turn text image into plain text and read it out using speech synthesis. In order to help visually impaired persons to recognize text in natural scene environment, Ezaki et al. [13] proposed to first locate text section and then zoom in the selected section for further recognition. Multiple extraction algorithms such as Sobel edge and Otsubinarization are integrated to improve accuracy. Chen and Yuille [10] developed an algorithm

for text detection in natural images to assist visually impaired people walking through city. A major drawback of the aforementioned applications is separating plain text displaying from current view. Users have to switch between result view and current view from time to time, interrupting their major task. As a new interface striding between the virtual and real, WAR supplies a new way to access assistant information without distracting users from their major tasks [20]. WAR can be used to solve the problem by overlaying the result directly on current view, so that users are able to watch the result and current view simultaneously and without distractions.

Ma et al. [25] used embedded mobile camera to detect text and then blended the text information with live video from camera. They further implemented translation function, which is very similar to WordLens [41], an app to translate any word when targeted by a mobile camera currently integrated with the Google Translate app. Wang et al. [42] developed a lexicon-based word recognition method. Recognized text is compared against words in the lexicon, which are then re-scored and non-maximal suppressed using features based on their global layout. Resultant words are aligned with their locations in images. Brian et al. [22] proposed NAVIG, a WAR guidance application to help the visually impaired. The system integrates a bio-inspired vision method for location and recognition. In order to facilitate totally blind users, many applications adopt speech synthesis technology to read out the text or use spatial audio overlays for direction awareness.

Other than augmenting text information, several works focus on extracting other information such as color and expression. Color Picker [7] and Color Identifier [21] are two available apps developed to assist the color blind people. The apps fetch color of texture at the center of view and then show the color with a circle or text form to users. Tanuwidjaja et al. [36] developed Chroma, a wearable MAR system to help the color blind users to see a filtered image of the current scene. It automatically adapts to various color blindness. Several work modalities such as highlighting, contrast, daltonization, and outlining are designed to meet users' diverse requirements. According to their experiments, color blind participants appreciate the system helping them identify color.

In face-to-face communication and social activities, facial expression is an important cue of participants' response, which is, however, not available for the visually impaired. Anam et al. [2] developed Expression, a dyadic conversation aid system to extract interlocutor's social signals such as facial features and emotions. There are a few vision based applications on Google Glass. OpenGlass [40] helps mobile users identify objects and environments via established crowd-sourcing technologies. It leverages social networks to search answers and delivers results to users through speech synthesis. Tanveer et al. [37] presented a Google Glass based prototype to help people to get affective cues including numbers, genders and ages of participants in a party scenario. Chroma [36] is a Google Glass application that extracts the colors in the current view and then display it to the users. Ha et al. [15] developed an assisted system on Google Glass to help users in cognitive decline.

The commented methods such as text recognition are computational intensive, Dumitras et al. [12] designed a client-server architecture to offload computational tasks on a server, which is able to guarantee run-time performance on mobile devices. The mentioned system leverages cloudlet based offloading to improve runtime performance. However, it requires at least two hop wireless transmission to complete cloudlets offloading, increasing time delay by at least 100 milliseconds due to the high round trip time to the cloudlet.

Hyperion differs from the related work in many ways:

(1) First of all, it is a working prototype of a WAR application that supports multiple types of interaction with mobile users.

(2) Secondly, from a systems point of view, it utilizes a companion smartphone to improve the overall system performance.

(3) Thirdly, it's, activity-based, modalities decrease the resources and energy needs and prolong the battery lifetime.

3 HYPERION

We develop *Hyperion*, a real time text extraction framework from the ambient environment. *Hyperion* is implemented in Google Glasses and part of it is offloaded to more resourceful devices. The design is based on three design principles that are presented in the next paragraph. Mobile devices are not the most appropriate platforms for text extraction while Google Glasses are designed to work with human eyes and provide first person view of the scene, so it is a better choice for systems related to human vision. Compared with mobile devices, the Glass works in a free-hand way to guarantee friendlier user experience [9, 18] in scenarios such as driving, walking while carrying bags in our hands. In addition, the development advice of Google Glass [14] on providing complementary unique services is consistent to our design.

In this section we depict the design principles that our system follows (Subsection 3.1 to achieve a good user design, interaction system-user and therefore, empower the user experience. Then, we describe the system architecture (Subsection 3.3) such as hardware, software, and frameworks. In Subsection 3.2, we illustrate the already implemented work modalities to explore, drive, and meeting as user case scenarios. Finally, we enumerate the main features implemented in order to interact with the system (Subsection 3.4): voice commands, speech synthesis, hand gesture and text recognition, and computation offloading.

3.1 Design Principles

Hyperion is based on the following three design principles:

Simplicity: a system for daily use should be as simple as possible to reduce intrusion and make an easy adaptation for non-experienced users. The simplicity is twofold in terms of hardware and functional aspects. Hardware components should not be bulky nor weighty to increase device intrusion. Similarly, system functions should be simple enough to meet the most urgent demands. Too many functions require additional manipulations and distract the user from the system intentions. In addition, functions should be invoked within a few steps so that users can access them easily and quickly.

Relevance: system should focus on what users are interested in. Many systems provide additional functions such as translation, text editing, and image storage. These functions are useful but not the most relevant. Hence, we focus on how to deliver text information to the users. The main focus of the system is to provide text-information around the user in a fast, non-intrusive and simple manner.

Adaptivity: we found from the formative work that vision disability causes inconvenience in various situations. In some cases such as

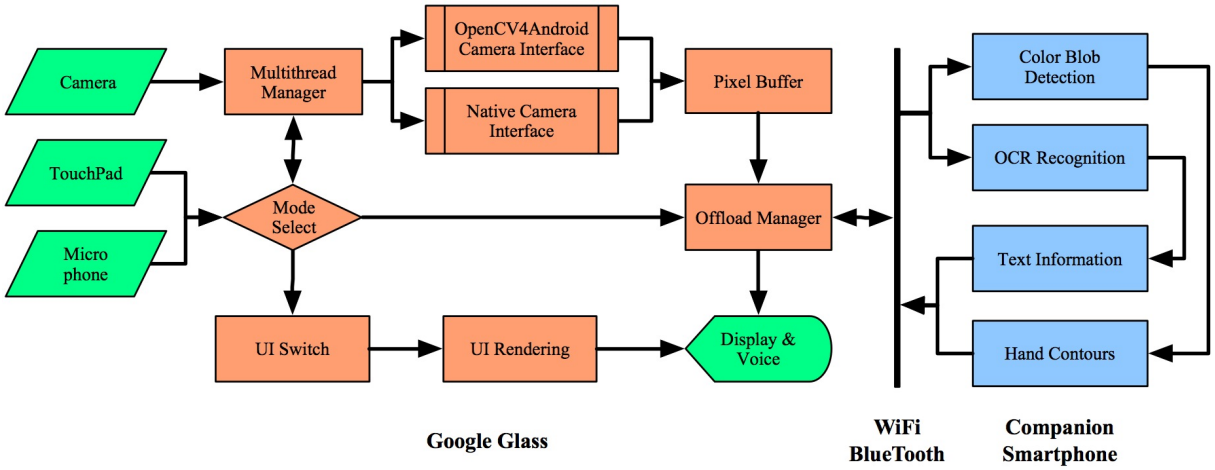


Figure 1: System flow of *Hyperion*. There exist three inputs (camera, touchpad and microphone) on the Google Glass and two outputs (display and voice). The Google glass is connected to a companion device that is responsible for computational expensive tasks.

looking for direction during shopping and driving, they occasionally require to access text information. A fire-and-forget usage model is suitable so that users are able to accept service quickly and continue with their main activities. In other situations such as attending class and meeting, they need continuous assistance, so an always-on model is preferable. User interaction should also adapt to scenarios. We implemented speech synthesis as an alternative as it does not distract visual attention in driving.

3.2 Work Modalities

In this Subsection we describe four different modes that we have defined in order to simplify the interactions available for some situations. For example, while driving the voice commands should be preferred than hand gesture recognition, as it might lead to dangerous scenarios while driving. Besides, while the user is in a meeting voice commands should be avoided. Therefore, we designed our system according to these environmental, and risk requirements so we do not only keep the user's privacy and security, but to facilitate the user interaction with *Hyperion*, see Table reffig:modalities.

These working modes can be preselected by the user so it limits the inputs/outputs in particular scenarios. *Hyperion*, supports (in the current stage) the following modes: explore, reading, meeting, and driving, used in different scenarios. The system will adapt to users' command, so we illustrate some modes according to the situation and the users' commands. The set of the available inputs are: Touchpad, Voice command and Hand gesture (see more in Subsection 3.4) and depending on the mode, a subset of them is used. For example, in the case of explore mode, all of them are activated while in the case of driving only the voice command is activated in order to not put the driver in any distraction. The possible outputs are the visual display, where the output is written in the Google Glass screen and the speech synthesis and not always both of them are available. For instance, in the case of exploration both of them are dispensable while in the case of reading only the visual display is working.

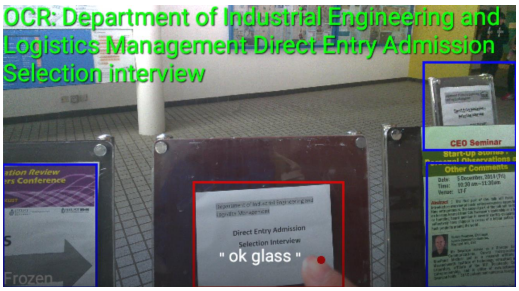
Explore mode This mode is designed to help users access text in non-critical situations, such as shopping in a mall and walking around campus. In these scenarios, users require additional information to occasionally make non-urgent decisions . The information is useful for prompt decisions and can be discarded immediately after use. For instance, a user may need the information from the direction board in a shopping mall to decide which direction to take. After one acquires the necessary information to choose the direction, it is not required any longer. Explore mode is designed as a fire-and-forget usage mode. As multiple text sections may exist in the current view, all candidate text sections are highlighted with virtual blue rectangles and become selectable as illustrated in Figure 2a. Users use their fingers to select a text section of their interest, which is highlighted with a red virtual rectangle. The system recognizes the text in the selected section and displays or reads it out to the users.

Reading mode This mode is designed for the users to conveniently access text printed on objects around them, such as books and restaurant menus in their hands, or product price tags on the shelves in front of them. For such cases, *Hyperion* is used as an amplifier. As illustrated in Figure 2b the app renders a virtual magnifying glass, which works as a viewfinder to amplify the spot that the user's finger is pointing at. The virtual hand lens can be moved to magnify any interesting content by simply placing the fingers on it.

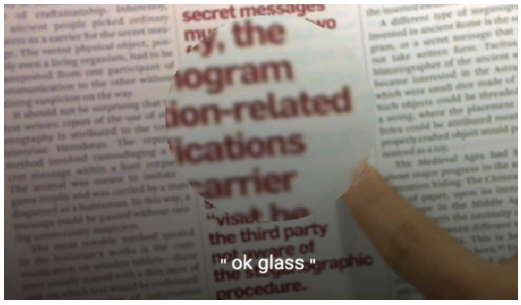
Meeting mode This mode is used when users have to access text from a presentation, such as attending a class or a meeting. It is inconvenient or not possible for users to get close enough when they need a better view to the presented information. Considering that text sections and users location are relatively still in the meeting mode, the system tries to recognize all texts located in a fixed recognition region outlined by four white corners as shown in the right panel of Figure 2c. The recognition region is automatically adapted according to the position and size of the display screen. The meeting mode does not include hand gesture manipulation as it could be perceived as impolite or even cause misunderstanding in the aforementioned scenarios. Since the meeting mode is usually used in quiet environments, only hand touch manipulations and display output are allowed.

Table 1: A list of work modalities that the system supports.

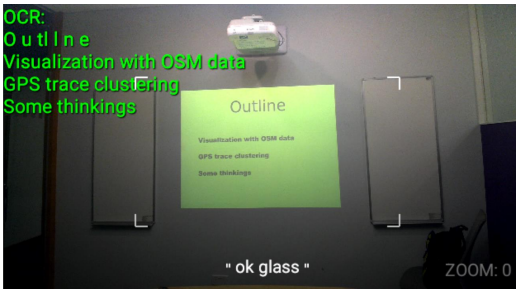
Work mode	User input			Output		Application scenarios and text recognition
	Touchpad	voice command	Hand gesture	Visual display	Speech synthesis	
Explore	✓	✓	✓	✓	✓	Text boxes (e.g., billboards, nameplates, direction boards) at a certain distance
Reading	✓	✗	✓	✓	✗	Printed text around or on nearby objects (e.g., books, price tags, restaurant menus, and exhibits)
Meeting	✓	✗	✗	✓	✗	Text shown in presentations (e.g., in class, meeting, report, and screening room)
Driving	✗	✓	✗	✓	✓	Text with certain patterns in critical situations (e.g., guiding boards by roads)



(a) Explore mode.



(b) Reading mode.



(c) Meeting mode.

(d) Driving mode.

Figure 2: The four implemented modalities. Each modality is suitable for different types of use, requires different input and implements different output. Examples of such uses are listed in Table 1.

Driving mode This mode is specially designed for drivers to acquire brief text while driving. Drivers have limited time to look for the text information on the road signs. The driving mode assists drivers to fast access the traffic information without much distraction. As most road signs share similar patterns with a few characters and specific foreground and background colors, the system automatically detects and recognizes text sections with certain patterns as illustrated in Figure 2d. Recognized texts are delivered to the drivers in order to reduce their distraction during the driving. In the driving mode, the system supports voice command rather than hand manipulations, while both display and speech synthesis are used to deliver recognized texts to the drivers.

3.3 System Architecture

Hyperion is developed based on Google Glass device with the Glass Development Kit (GDK), which is an add-on to the Android SDK specifically for the development of Google Glass applications. Figure 1 depicts the system flow of *Hyperion*. The computationally intensive parts of it are executed remotely to a companion smartphone while the light ones are executed locally. The embedded front camera on Google Glass, with resolution 1920*1080, continuously records the current scene. Limited by the physical size of the display screen, we use OpenCV4Android to provide low-resolution images when we use the camera's preview interface. However, the resolution is not sufficient for text recognition but it is satisfactory for hand gesture detection.

In order to be able to perform text recognition we employ the native camera interface to fetch raw data from camera buffer. We have designed two subroutines to separate the threads of the parallel execution of text recognition and hand gesture detection. These two processes are executed at different frequencies and are scheduled by the thread manager as illustrated in the system flow of *Hyperion* in Figure 1. As the hand gestures are not fast, the dedicated process can be executed in relatively low frequency. However, in order to capture any text change, we have to detect text at higher frequency. The main thread is responsible for display, UI rendering, and user input response and should run at full frequency in order to provide smooth view to the users. Both hand gesture detection and text recognition are computationally intensive and considering the limited computational capabilities and the battery life of Google Glasses such applications are not feasible. We leverage the offloading technology proposed in [19] to improve runtime performance and long-time operation. However, cloud services may be unavailable due to failures of network and distant architectural tiers or may impose monetary costs for their use. One the other hand, smartphones have been evolved and become powerful and ubiquitous and they have multiple network interfaces. *Hyperion* used WiFi-direct and Bluetooth to offload the computationally expensive parts of it to the user's companion smartphone. The smartphone executes the offloaded tasks and sends back the results to the Google Glasses. The results may include text strings, location and hand contours. The offload subroutine is managed by the offload manager in Figure 1. Since user interface does not require heavy computation, it is rendered locally on the Google Glasses. In order to keep the overhead and the requirements of *Hyperion* as low as possible, the usability and user experience of the system in different scenarios (Subsection 3.2) we have identified four major modes that cover all the possible use cases but do not require as many resources as a generic version. Each work mode has its unique interaction for specific application scenarios. When the system switches to another work mode by pad touching or voice command, the related user interface is selected and rendered for interaction, as shown in the Figure 1.

3.4 System Implementation

Our work modalities require several major technologies including voice command, speech synthesis, hand gesture detection and text recognition, which have been implemented.

Voice command: We predefined several voice commands to switch work modalities and trigger functions. All voice commands start with "ok glass" while this command by itself moves to the upper menu. The keywords are related to the functions and limited to less than three words. For instance, "ok glass, start *Hyperion*" will launch our system. "ok glass, switch to explore" will switch to explore mode. Whenever the user selects a mode that supports hand gesture, she has 5 seconds to put her hand in the middle of the screen in order to initialize the hand gesture detection. All voice commands are defined as meta data in an extensive markup language (XML) file. We added voice trigger intent filter to detect all defined voice commands. In order to make it work, we are required to explicitly declare development permission in the manifest file.

Speech synthesis: The system uses Text To Speech (TTS) to synthesize speech with given texts. *Hyperion* implements the initial listener so that the system automatically initializes the module with specific settings when it is launched. Whenever texts are recognized from current view, they will be sent to TTS for speech synthesis if the current work mode supports speech output. As TTS requires intensive system resource, the system stops it to release occupied resource when it is not used in certain work modes (explore and meeting) or system conditions (e.g., pausing).

Hand gesture detection: There exist several approaches for hand gesture detection. Given that the system does not require accurate positioning and that Google Glass has poor computational capabilities, we employed a lightweight color-based detection algorithm. The method uses reference color for detection and requires to sample the color of the hand skin and then utilizes the skin color to extract pixel sections with similar color in current view. As in most cases user's hand takes indeed a large part of current view, it is reasonable to assume that the largest pixel section with similar color in images is the hand section. The largest pixel section is further extracted to get contours of user's hand, which is used for positioning in the image space (Figure 3). The color-based method is sufficient for our system in most cases. However, it occasionally fails when ambient light or background objects have similar color of hand skin.

Text recognition: We integrate the OCR-based tess-two library[3] with consideration of recognition accuracy and computational efficiency. Fast head motions cause image blur, which severely degrades recognition accuracy. As fast head motions indicate that the user may be not gazing at any text section, we determine whether the user is looking at any text by using the squared distances between consecutive frames [33]. When the system detects that the user is gazing at any text, a separate thread is notified to analyze the frame captured from the Glass camera. Result accuracy fluctuates due to image blur, ambient lighting, and capture angle. A confidence factor is used to evaluate whether to accept or discard the recognized result. Considering that text information in captured image sequence is highly relevant, results with continuous lower confidence factors will be discarded. All accepted results are rendered to display or spoken out. As the tess-two is resource-intensive, the system releases any native memory whenever text recognition is not used in other work modes. However, in order to guarantee efficient memory access, occupied resource remains in memory during continuous text recognition.

Computation offloading: We speed up *Hyperion* using a computation offloading approach with the help of a companion device. The computational intensive methods are offloaded to nearby mobile device to improve run-time performance. To facilitate system development, we proposed an annotation-based offloading method. System developers only need to annotate "@Offloadable" at the definition point of each computational intensive method. No additional development workload is required. The compiler resolves the annotation and automatically submits the method to the task queue for remote execution, which is implemented using the Java reflection mechanism. *Hyperion* employs two separate lightweight threads for the offloading service: one thread is responsible for socket listening, and the other is used to receive data.

[3] https://github.com/rmtheis/tess-two

(a) (b) (c)

Figure 3: Once the user's says the command "ok glass", the system will start a color-based hand gesture recognition method: (a) sampling skin color; (b) detecting sections with specified color; (c) hand section extraction. After the system detects the nearest text that the hand points to, it starts the OCR features to extract the information.

4 EVALUATION

We conducted several experiments in order to evaluate the performance of *Hyperion* from aspects of both technical specification and user experience. In this Section we want to study not only the system performance but the user experience with our proposed system. Experiments are carried out with and without offloading to evaluate the performance and sustainability improvements. For the experiments in the user experience case, we invited a set of 6 participants, 4 males and 2 females (ages from 20 to 24), from our university campus. All participants are from different departments of the university with an average age of 24. We measure the **realtime performance and sustainability** of *Hyperion* by running multiple experiments. Distance and view angle are related to the accuracy of the recognition result (Figure 7). We follow a similar experiment setup [12] to evaluate the **system accuracy** (Figure 8). Rather than take images of text from various distances and angles and then recognize the text from the images, the participants stand at different distances at different angles to recognize text by using our system. In addition, the text size is also considered and tested int the experiment. **Snellen chart**[27] is widely used to measure visual acuity. In the experiment, participants are asked to state the letter in the chart at the distance of 6 meters with no vision correction, eyeglasses, and *Hyperion*. The experiment is designed to verify whether and how much *Hyperion* can compensate for the vision loss of the participants. Moreover, we set up another set of experiments for each of the four modalities and measure the user satisfaction:

Walking around campus: The experiment is used to verify system functions in explore mode. Participants walk around our campus and try to collect text information without standing close to it. Text sections such as bulletins and posters in their current view are highlighted with virtual rectangles. The participants use their fingers to select a text section. Our system recognizes the texts in the selected section and displays them to the participants.

Reading a book: The experiment is used to verify system functions in reading mode. Participants are required to read a book page full of text. The book is placed at a certain distance (e.g., 40 to 50 cm). The participants manipulate the virtual magnifying glass with their fingers to amplify characters in the book.

Having a meeting: The experiment is used to verify system functions in meeting mode. While participants are sitting in a meeting room, slides with text information are projected on a screen placed in front of them. Participants are thus asked to recognize the text information using our system, which adjusts the virtual recognition region to precisely fit the projection screen.

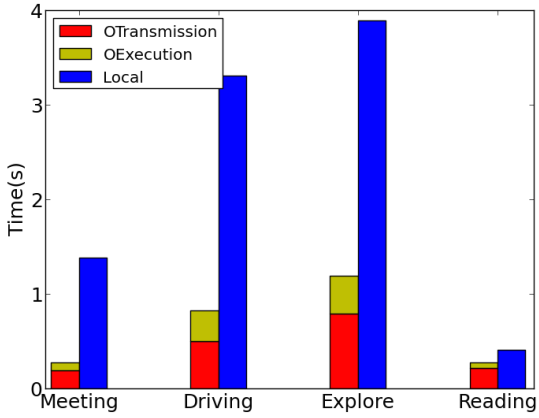


Figure 4: The time cost per frame of system in different modes with and without offloading. The blue bar shows time per frame without offloading. The combination of the yellow bar (execution time on local device) and the red bar (transmission time) represents time per frame using offloading.

Walking along a roadway: The experiment is used to verify system functions in driving mode. We replace driving a car by walking along a road for safety consideration. Participants are walking along the road and wearing the Glass device. Our system tries to automatically recognize front directional boards (e.g., white text with red background) in current view and reads it out to the participants.

4.1 System performance evaluation

The major challenges of developing applications for Google Glass are the poor computational capabilities and the limited battery capacity, which constrain the run-time performance and sustainability of the applications. We adopted the device to device offloading between the Google Glass and a Google Nexus 5 smartphone (with 2.26GHz CPU and 32 GB memory) to deal with these issues. Figure 4 shows the time cost of a single frame by running our system with and without computation offloading in different modes. Although the data transmission between the Glass device and the smartphone

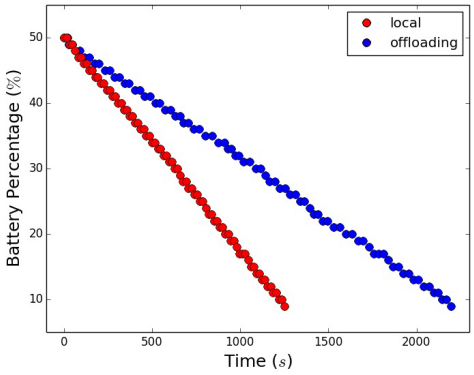


Figure 5: The energy consumption test by running the system in explore mode with and without offloading.

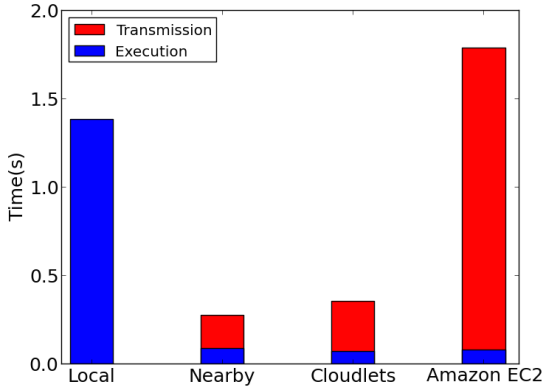


Figure 6: The performance comparison between different types of offloading.

produces some latency (the red bar), the powerful computational capabilities of the smartphone dramatically speed up the execution of computational intensive tasks, which results in a great decrease of the total time cost. The performance gain varies per modes. The explore mode achieves the highest acceleration ratio as it includes particularly intensive computational tasks such as hand gesture detection, multiple text sections detection, and text recognition. On the other hand, the reading mode only requires lightweight image magnification, so the margin for improvement is limited. Explore and driving modes require processing of large amount of data because are dealing with recorded video. The use of computation offloading requires this data to be send from the glasses to the companion device and this requires substantial time. This fact makes *Hyperion* unable to achieve real-time performance in scenarios where network latency is considerable [6]. A side benefit of offloading is the reduction in energy consumption to prolong the execution time of the system on the Google Glass. As computational intensive tasks are executed on offload devices, the CPUs on local device, where *Hyperion* runs, is able to work at lower frequency, which helps reducing the energy consumption. In our system the explore mode is the most computational intensive, and therefore it achieves beneficial in terms of energy savings. Figure 5 shows the time required to consume the same amount of battery energy with and without computation offloading in explore mode. Without offloading strategy the system drains battery almost twice faster, which indicates that computation offloading significantly improves system sustainability.

Apart from device–to–device offloading, there are also other types of offloading such as cloudlet based [15], and private and commercial cloud based offloading [24]. Apart from the cost consideration, we choose companion mobile devices for offloading because it is less intrusive than the cloudlet offloading which may require cumbersome laptops. Additionally, connecting the Glass device with users' own smartphones is more secure and more reliable. To verify the performance gain between different types of offloading, we tested the system in the meeting mode by offloading to nearby device, cloudlets, and an Amazon EC2 instance . As illustrated in Figure 6, the nearby device offloading is also the best among other offloading approaches, which is attributed to the significantly less data transmission time between local device and offload devices. It is worth mentioning that the connection to the companion device is single hop while the one to the cloudlet is two-hop via a WiFi router, which is the reason why the transmission time to cloudlet is higher.

4.2 User study evaluation

In this Subsection we focus on the visual acuity using a Snellen chart and the overall system experience with the previous mentioned four scenarios. In the first experiments we use the Snellen chart to evaluate the visual acuity. The Snellen chart test quantitatively evaluates how *Hyperion* is able to compensate for the vision loss. The setup consists in test points that are placed every one meter along the distance direction and every 30 degrees along the angle direction. The 6 participants were invited to recognize text of different sizes at each point using *Hyperion*. The average recognition accuracy is evaluated at each test point and normalized between 0 and 100% as shown in Figure 7. The accuracy declines along both the distance and angle directions for a given text size. It also decreases as text becomes smaller at the same test point. However, the plot shows that the recognition accuracy decreases faster along the distance direction than angle direction, which indicates the recognition accuracy is much more sensitive to the distance than the angle. It is partly due to the reason that text resolution drops dramatically when the distance increases, but it stays the same from different angles with only certain shape distortion. As people normally turn and target the heads at what they are looking at, angle offset is not a serious problem for practical use. On the contrary, the distance and text size are directly related to the ability to see the text. However, the farthest distance to recognize a text is determined by the Google Glass camera lens. We expect future updates of the Google Glass camera hardware to improve "visual acuity" of the system.

We invited six participants with different level of visual impairment to perform the test. Participants stood at 6 meters away from the Snellen chart and were asked to state the line number of the smallest characters that they are able to distinguish. Without taking any device, a half can see the largest letter in the first line, while the other half are not able to see any letter. Figure 8 illustrates their visual acuity when they wear eyeglasses that enables magnification to compare with *Hyperion* performance. Four participants found

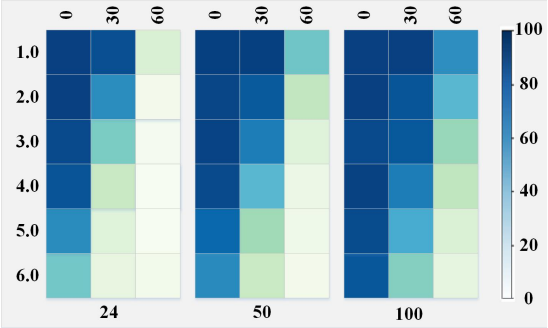


Figure 7: Recognition accuracy (color) of the Snellen chart from 1 to 6 meters (Y-Axis) and different angles (X-top-axis) at each test point with different text sizes.

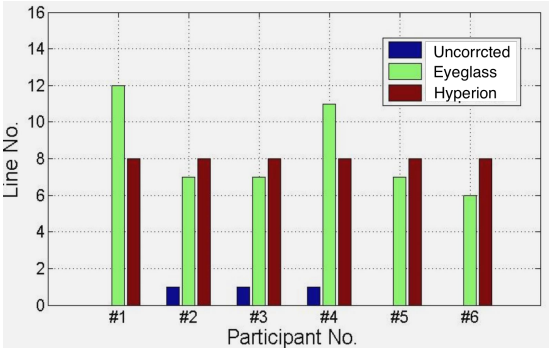


Figure 8: Vision acuity of participants wearing no device, eyeglass, and Hyperion. Letters in higher line numbers are much smaller.

that *Hyperion* improved their vision conditions. The results show that our system improved their visual acuity to see more than one to two lines of letters compared to the eyeglasses. However, other two participants said it did not help them. Constrained to the Glass's camera lens and resolution, the smallest letters that *Hyperion* is able to recognize are in the 8th line. The other two participants, who found our system did not work, have high visual acuity to see letters in the 11th and 12th lines, which contain smaller letters than the 8th line. From the experiments we discover that our system improves the vision of people who can not recover their vision acuity high enough (e.g., not able to see the letters in the 8th from 6 meters far away) even wearing eyeglasses or contact lenses. *Hyperion* fills the gap of users' desired vision and the corrected vision using eyeglasses and/or contact lenses, which is consistent to the system goal of supplementing rather than replacing existing tools such as eyeglasses and contact lenses for the visually impaired.

The remaining four experiments designed to verify the different modes in terms of user experience. To qualitatively measure user's experience of *Hyperion*, we adopted an extended technology acceptance model including perceived enjoyment (PE), perceived usefulness (PU), perceived ease of use (PEOU), and intention to use (IOU), which are widely employed as predictors to applicability and acceptance of a system. We then used the Likert scale[1] score to evaluate all four terms of each work mode. Each participant was

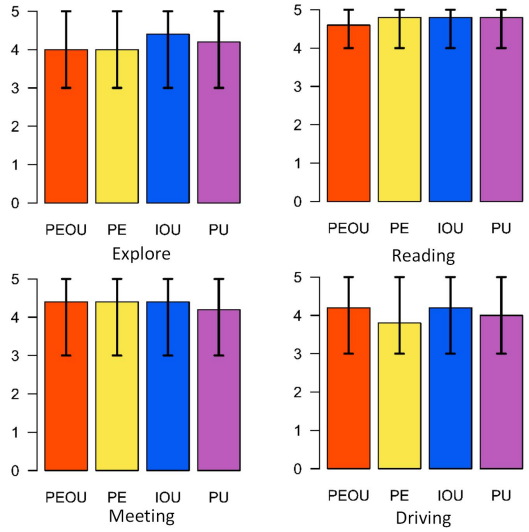


Figure 9: The Likert scale score of user experience in explore, reading, meeting, and driving modes.

asked to test *Hyperion* in the four different modes and evaluate our system based on his/her experience. The score in the Likert scale ranges from 1 to 5. We decided to use these four criteria to better identify the weaknesses of our system, but preferred not to use a higher number of them because we thought that the unexperienced user would not be able to well discern between finer criteria. The results of our user based subjective evaluation are summarized in Figure 9. Overall, the evaluation of the users is positive with the lowest scores of 3 points and many features scoring between 4 and 5 points. The highest scores and small biases to the average values of all criteria indicate that reading mode is the most popular. Many participants said the user interaction of reading is convenient and similar to a real magnifier glass. It is believed that the reading mode will be very useful for senior people. Each criteria of the meeting mode also gets a high average score, but the biases are much larger as the participants have diverse opinions. Explore and driving modes have relative low average scores with the driving mode being the lowest. Some participants thought that it may be dangerous to use the system during the driving. Other participants doubted whether the system is able to detect road signs when they drive very fast along the high way. As most shopping streets are full of different billboards and they may partially overlay each other from different view angles, some participants also suspected the explore mode may not recognize and find the necessary information.

Notably, many participants prefer using *Hyperion* in indoor rather than outdoor environments, which is also observed from the figure that reading and meeting modes gain higher scores than explore and driving modes. We explained the design purpose and functions to the participants and asked them to score the IOU term of each mode before they took the experiments, so the IOU has the highest scores in the all four modes. However, the PE and PEOU are relative low as the system does not run smoothly and long enough due to the poor computational capability and quick overheating of the Glass device.

5 DISCUSSION

In this Section we discuss the possible disadvantages and issues with the current design, the future paths to follow in our work, and future guidelines for our future system for the visually impaired feasibility using participatory design.

By leveraging computation offloading, we alleviate the problem of poor computational power and limited battery life of Google Glasses as explained in [36]. However, *Hyperion* still has several limitations that are explained in this section. Our current prototype is designed to recognize text information with a few words. It is not suitable for text regions full of characters. Although it may be technically feasible, we do not integrate it into our system for several considerations. Limited to Glass camera's field of view, too many characters will make individual character difficult to recognize. Moreover, it requires heavy computational cost as the number of characters increases. Furthermore, designing a user-friendly interface to deliver too much information is not recommended on a tiny prism screen as the Google Glass [3]. We tried scrolling and multi-page display, but neither is satisfactory in user interaction and display efficiency.

Although a large portion of computation has been outsourced, the battery still drains fast (about 1.2% per minute) even with the basic video recording through OpenCV4Android interface. Without efficient cooling system, Google Glass becomes quickly hot and its performance dramatically drops. Ha et al. [15] wrapped the Google Glass with an ice pack to sustain CPU computational efficiency. However, it is only for prototype evaluation. The ice pack seriously induces device intrusion to impede friendly user experience.

The system does not support other information patterns except for text. Figures and geometry designs such as road signs are also important cues to understand surrounding environments. The system implements image-based zooming function for amplification (partial used in reading mode), which is able to amplify non-text patterns. However, the image-based amplification is limited due to Google Glass camera and display screen resolutions. OpenCV4Android has the ability to extract non-text features, but it requires significant effort from other fields (e.g., machine learning) to understand non-text content.

The optical axis of built-in camera is not consistent with eyesight. The participants have to target the Glass camera and not only their eyes at objects. Many participants of our study feel dizzy after using the Glass for a while. The deficiencies seriously impede usability and acceptance of the system. Another major system limitation is induced by the vision-based manipulations. Even with rapid advances in camera devices, the vision-based amplification and recognition are constrained to the limit of optical imaging.

Other aspect we have to mention regarding *Hyperion* wearable device is that Google has discontinued the Google Glass project, although it is still one of the main platform for AR research purposes. The development of similar AR devices continues in the market and with the hardware improvements on power and efficiency it will be on going for the next years until the virtual world converges with the real one. These devices such as MAD Gaze, and other Android powered smart-glasses offer analogous characteristics and features to Google Glass. Furthermore, Microsoft supports the AR advancements with its new wearable device HoloLens, which offers similar features that Google Glass, and future developments on contact lenses focus in AR/VR capabilities. Therefore, future developments of Hyperion will be feasible, enabling the increment of devices coverage.

Hyperion is based on hand gesture as the major interaction. It may not be the most natural interaction approach but it is preferable due to the current Glass hardware. We considered using an additional camera mounted on the Glass frame to track user's eyeball movements, in order to detect what the user is gazing at. An additional camera, however, would increase devise intrusion. In our future work, we will use stickytape sensors at the inner side of Google Glass frame to detect eye muscles, which is then used to evaluate eyeball movements. Other alternative to track users' gaze is to follow similar techniques used by HoloLens to track users head movement and therefore, their gaze. The addition of better hand/finger gesture recognition can be advantageous to move around the menus and have more accurate object selection in the real world.

As par of our future work, we will implement our current system in other Android powered smart-glasses (i.e., MAD Gaze), and the new Microsoft HoloLens platform; and evaluate the performance with different scenarios (indoor/outdoors). With more powerful hardware, and new developments of more efficient computer vision techniques, AR systems will become part of our daily life. Moreover, Microsoft HoloLens does not provide any native AR feature to extract text form the ambient environment, and our system can take advantage of its multiple sensors to provide for example better scanned environment for AR layouts over real objects or OCR text applications, and improve *Hyperion* user experience.

5.1 Rethinking Visual Impairment

As part of future work, *Hyperion* could be helpful for visually impaired in their daily life. In order to design a system that is both useful in function and flexible in user interaction, we require wide and in-depth understanding of issues related to the visual impairment; therefore we have created a survey to contribute our system initial participatory design.

We invited 21 participants for a paper-based survey questionnaire, which consists of 10 questions regarding to their difficulties associated with vision loss, current solutions for their vision correction, and their specific needs. All participants are from different departments of our campus with an average age of 24. Participants have different levels of the visual impairment, with most (12) of severe level, 6 with moderate level, and 3 with mild vision loss.

Participants are visually impaired due to various reasons. The most common reason is a continuous staring at the screen and book, which is consistent to the report [11]. Other reasons are attributed to genetic acquisition and long-time activities in low light conditions. 81% of participants adopt eyeglasses to compensate for their visual loss, with the rest alternating contact lenses. Seven participants explain their preference to eyeglasses over contact lenses due to allergic reaction. Over a third of participants are not satisfied with their current solutions, while reasons vary from inconvenience during sports to allergic reactions and heavy weight of eyeglasses. It is interesting that most of the complaints come from those suffering from a more severe condition. This is partly due to the reason that they require thicker glass lenses, which makes eyeglasses heavier. In

addition, severe visually impaired participants deeply depend on aid tools in their daily activities.

We also asked participants to describe their most-in-need features for visual correction system. Over half participants (57%) show interest in capability of automatic adaptation to different levels of the visual impairment, indicating that they prefer solutions that can change along with shifts of their vision acuity. It is majority due to the fact that they has to buy new eyeglasses and/or contact lenses when their vision acuity changes. The frustrations caused by the visually impairment are diverse.

We list several typical activities that the visually impaired people find inconvenient to deal with in their daily life.

Shopping: Many participants find it is inconvenient during shopping, especially in large shopping mall where they rely on signs to find the stores. Many signs hang on wall and ceiling to save space, which makes it difficult for them to see clearly. In addition, many mini stores (e.g. coffee shop or drinking bar) list supply items on billboard on the other side of counter. They have to inquire again and again before confirming their orders.

Sports: It is also inconvenient and occasionally dangerous for them when participating in many sports such as swimming, snorkeling, and scuba diving. Many participants said they would still wear eyeglasses or contact lenses when playing basketball and football in order to get clear visions, but from time to time their eyeglasses may be knocked away or even broken by fierce body collision. Sweat accidentally flows into their eyes, making them afflictive especially when they are wearing contact lenses. They also have to clear their contact lenses to avoid irritation.

Taking class/Meeting: As most participants are college students, the first activity that they find affected by the visual loss is taking class and attending a group meeting. Many of them can not see clearly what's on blackboard or projector screen without vision aid, which makes them unable to catch up with the progress. Sometimes it is embarrassing when they are asked to answer questions displayed on it. They have to sit near the blackboard and screen, or ask neighboring classmates for help. It is even worse when speakers and display screen are separated by the podium and far from audiences.

Driving: Driving is another situation that many participants mentioned. They state that it is difficult to get traffic information such as how close is front car and what's color of signal lights. They have to carefully drive their cars to avoid any traffic accident. The first author had a very dangerous experience caused by the visual impairment of his friend during driving. His friend could not see information on traffic board at the highway fork. In order to confirm the direction, he braked the car suddenly, almost colliding with a following car.

5.2 Inspirations

We have to mention that even though all participants are suffering from the visual impairment and they are not satisfied with current solutions, no participant chooses any medical method such as laser surgery. Eyeglasses and contact lenses remain the choice of most participants. It may be partially due to cost and health considerations of medical surgery. Another reason lies in the widely social acceptance of eyeglasses. Just as a participant noted, eyeglasses as a primitive technology is attributed to the fact that it was invented 800 years ago, which indicates that wearable system similar as eyeglasses may be preferable for vision correction. From their answers to our questionnaire, we discover that all situations that the visually impaired suffer from are caused by their uncertainty of what they want to see or what they are looking at. If we can help them reduce or even avoid such kind of uncertainty, they are able to alleviate or even avert the situations as normal people.

Another discovery of our survey is that due to the diverse nature of the situations that users can be (i.e., nature of their surroundings), similar approaches to interact with the device might not be the best solution to handle all cases. As we have included in our current system, in some situations voice commands and text to speech information is more suitable for drivers, so as not to distract their attentions; while in other scenario such as class or meeting room, user gesture control and displaying information are preferable to keep quiet and not disturb the others.

In the future design, we have to take into into account the insights from our previous study regarding visually impaired for our future development. The addition of other features that might be useful to visually impaired users such as face and object recognition. In the former, the users can get sound information about the faces recognized, for example who is in front of the users if she is known by the user. The latter, we can help users to recognized similar physical objects in a supermarket or at home so the users does not confuse them, for example orange juice and milk packages.

6 CONCLUSION

In this paper, we proposed *Hyperion* a wearable Augmented Reality system to extract text information from the ambient environment. *Hyperion* supports Google Glass and in the future other commercialized smart-glasses, is able to provide offloading methods to the smartphone, cloudlet, cloud to be energy efficient and offers a user real-time experience. This paper describe the design principles of *Hyperion* such as simplicity, relevance and adaptivity that lead towards an efficient, interactive and responsive wearable system. In order to provide some situations limitations due to privacy, security concerns (i.e., driving, meeting, lectures) we design four work modalities that offer different input/output interactions. The implemented main features: voice, hand gesture, text recognition and computation offloading enable a seamlessly user experience. Furthermore, this paper evaluates the system performance, system precision (i.e., users' visual acuity) and the mentioned user experience with several experiments. We can claim that offloading techniques are still a useful method to improve system responsiveness and energy efficiency, due to the lack of processing capabilities of wearable devices, energy consumption oriented. In spite of the current deficiencies of Google Glass, the experiment results show that the participants are interested in *Hyperion*. Current trends on wearable mobile devices imply that there is a great potential and interest from investors so we expect more powerful and resourceful devices to be available in the near future. Our next step include the participation of more wearable devices (i.e. smart watches) in our system in order to make the use of our system more transparent to the mobile user. We envision the use of smart watches in the control of the Google Glass as one more complementary way together with asynchronous information downloading from cloud databases.

7 ACKNOWLEDGEMENTS

This research has been supported, in part, by General Research Fund 26211515 from the Research Grants Council of Hong Kong, the Innovation and Technology Fund ITS/369/14FP from the Hong Kong Innovation and Technology Commission, and the MRA15EG01 research grant from Microsoft Research Asia.

REFERENCES

[1] I Elaine Allen and Christopher A Seaman. 2007. Likert scales and data analyses. *Quality progress* 40, 7 (2007), 64.

[2] ASM Iftekhar Anam, Shahinur Alam, and Mohammed Yeasin. 2014. Expression: A Dyadic Conversation Aid using Google Glass for People with Visual Impairments. In *The ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 211–214.

[3] Mark Billinghurst and Hayes Raffle. 2014. *The Glass Class: Designing Wearable Interfaces*. Technical Report. CHI courses.

[4] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 13–16.

[5] Samia Bouzefrane, Amira F Benkara Mostefa, Fatiha Houacine, and Herve Cagnon. 2014. Cloudlets authentication in NFC-based mobile computing. In *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2014 2nd IEEE International Conference on*. IEEE, 267–272.

[6] Tristan Braud, Farshid Hassani Bijarbooneh, Dimitris Chatzopoulos, and Pan Hui. 2017. Future Networking Challenges: the Case of Mobile Augmented Reality. In *Proceedings of the Annual IEEE International Conference on Distributed Computing Systems (ICDCS 2017), Atlanta, GA*. IEEE.

[7] Martin Brizio. 2013. Color Picker. (2013). http://www.youtube.com/watch?v=Cpcwah-xnKA. http://www.youtube.com/watch?v=Cpcwah-xnKA.

[8] Dimitris Chatzopoulos and Pan Hui. 2016. ReadMe: A Real-Time Recommendation System for Mobile Augmented Reality Ecosystems. In *Proceedings of the 2016 ACM on Multimedia Conference (MM '16)*. ACM, New York, NY, USA, 312–316. https://doi.org/10.1145/2964284.2967233

[9] Ke-Yu Chen, Daniel Ashbrook, Mayank Goel, Sung-Hyuck Lee, and Shwetak Patel. 2014. AirLink: Sharing Files Between Multiple Devices Using In-air Gestures. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*. ACM, New York, NY, USA, 565–569. https://doi.org/10.1145/2632048.2632090

[10] Xiangrong Chen and Alan L. Yuille. 2004. Detecting and reading text in natural scenes. In *Proceedings of the IEEE conference on Computer vision and pattern recognition*. 366–373.

[11] DailyMail. 2014. Nearly 70% of adults suffer from blurred vision caused by staring at their computer screens. (2014). http://www.dailymail.co.uk/news/article-2536785

[12] T. Dumitras, M. Lee, P. Quinones, A. Mailagic, Dan Siewiorek, and P. Narasimhan. 2006. Eye of the Beholder: Phone-Based Text-Recognition for the Visually-Impaired. In *IEEE International Symposium on Wearable Computers*. 145–146.

[13] N. Ezaki, M. Bulacu, and L. Schomaker. 2004. Text detection from natural scene images: towards a system for visually impaired persons. In *Proceedings of the International Conference on Pattern Recongition*. 683–686.

[14] Google Glass. 2014. Google Glass Design Principles. (2014). https://developers.google.com/glass/design/principles https://developers.google.com/glass/design/principles.

[15] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. 2014. Towards wearable cognitive assistance. In *Proceeding of the MobiSys*. 68–81.

[16] A. C. Haugstvedt and J. Krogstie. 2012. Mobile Augmented Reality for Cultural Heritage: A technology Acceptance Study. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 247–255.

[17] Gang Hu, Jie Shao, Lianli Gao, and Yang Yang. 2015. Exploring Viewable Angle Information in Georeferenced Video Search. In *Proceedings of the 23rd ACM International Conference on Multimedia (MM '15)*. ACM, New York, NY, USA, 839–842. https://doi.org/10.1145/2733373.2806344

[18] Zhanpeng Huang, Weikai Li, and Pan Hui. 2015. Ubii: Towards Seamless Interaction Between Digital and Physical Worlds. In *Proceedings of the 23rd ACM International Conference on Multimedia (MM '15)*. ACM, New York, NY, USA, 341–350. https://doi.org/10.1145/2733373.2806266

[19] Zhanpeng Huang, Weikai Li, Pan Hui, and Christoph Peylo. 2014. CloudRidAR: a cloud-based architecture for mobile augmented reality. In *Proceeding of the MobiSys 2014 Workshop for Mobile Augmented Reality and Robotics-based technology systems*. 29–34.

[20] Z. P. Huang, P. Hui, C. Peylo, and D. Chatzopoulos. 2013. *Mobile Augmented Reality Survey: A Bottom-up Approach*. Technical Report. HKUST Technical Report (cited as arXiv:1309.4413).

[21] Alberto Iglesias. 2013. Color Identifier. (2013). http://play.google.com/store/apps/details?id=com.visualnet.glasscolor http://play.google.com/store/apps/details?id=glasscolor.

[22] Brian F. G. Katz, Slim Kammoun, Gaetan Parseihian, Olivier Gutierrez, Adrien Brilhault, Malika Auvray, Philippe Truillet, Michel Denis, Simon Thorpe, and Christophe Jouffrais. 2012. NAVIG: augmented reality guidance system for the visually impaired. *Virtual Reality* 16, 4 (2012), 253–269.

[23] Hiroaki Kimura, Eiji Tokunaga, and Tatsuo Nakajima. 2007. System Support for Mobile Augmented Reality Services. In *Proceedings of the 2007 ACM Symposium on Applied Computing (SAC '07)*. ACM, New York, NY, USA, 1616–1623. https://doi.org/10.1145/1244002.1244346

[24] Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, and Xinwen Zhang. 2012. ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *Proceedings IEEE INFOCOM*. 945–953.

[25] Derek Ma, Qiuhau Lin, and Tong Zhang. *Mobile Camera Based Text Detection and Translationd*. Technical Report. Stanford University.

[26] Justin Manweiler. 2014. Cloud-based Mobile Augmented Reality, Pitfalls and Strategies in a Realtime Deployment. In *Proceedings of the 2014 Workshop on Mobile Augmented Reality and Robotic Technology-based Systems (MARS '14)*. ACM, New York, NY, USA, 21–21. https://doi.org/10.1145/2609829.2609892

[27] Paul McGraw, Barry Winn, and David Whitaker. 1995. Reliability of the Snellen chart. *BMJ: British Medical Journal* 310, 6993 (1995), 1481.

[28] Stefano Messelodi, Carla Maria Modena, Lorenzo Porzi, and Paul Chippendale. 2015. i-Street: Detection, Identification, Augmentation of Street Plates in a Touristic Mobile Application. In *Image Analysis and Processing?ICIAP 2015*. Springer, 194–204.

[29] Opivision. 2014. EYES 101: Basic Facts and Anatomy. (2014). http://www.opivision.com/index.php?option=com_content&view=article&id=97&Itemid=100

[30] Haave Oy. 2014. SayText. (2014). https://itunes.apple.com/cn/app/saytext/id376337999?mt=8

[31] Alexandros Panayiotou. 2016. RGB Slemmings: An augmented reality game in your room. (2016).

[32] David Parry. 2014. Magnifying Glass. (2014). https://play.google.com/store/apps/details?id=com.davidparry.magnifying

[33] Swati Rallapalli, Aishwarya Ganesan, Krishna Chintalapudi, Venkat N. Padmanabhan, Lili Qiu, and Mahadev Satyanarayanan. 2014. Enabling physical analytics in retail stores using smart glasses. In *Proceeding of the MobiCom*. 115–126.

[34] Mahadev Satyanarayanan, Zhuo Chen, Kiryong Ha, Wenlu Hu, Wolfgang Richter, and Padmanabhan Pillai. 2014. Cloudlets: at the leading edge of mobile-cloud convergence. In *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*. IEEE, 1–9.

[35] Smart Mobile Software. 2014. Mobile OCR. (2014). https://play.google.com/store/apps/details?id=mobileocrfree

[36] Enrico Tanuwidjaja, Derek Huynh, Kirsten Koa, Calvin Nguyen, Churen Shao, Patrick Torbett, Colleen Emmenegger, and Nadir Weibel. 2014. Chroma: A Wearable Augmented-Reality Solution for Color Blindness. In *The ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 799–810.

[37] M. Iftekhar Tanveer and Mohammed E. Hoque. 2014. A Google Glass App to Help the Blind in Small Talk. In *Proceeding of the ASSETS*. 297–298.

[38] Wallace Ugulino and Hugo Fuks. 2015. Landmark Identification with Wearables for Supporting Spatial Awareness by Blind Persons. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 63–74. https://doi.org/10.1145/2750858.2807541

[39] Tim Verbelen, Pieter Simoens, Filip De Turck, and Bart Dhoedt. 2012. Cloudlets: Bringing the cloud to the mobile user. In *Proceedings of the third ACM workshop on Mobile cloud computing and services*. ACM, 29–36.

[40] Dapper Vision. 2014. OpenGlass. (2014). http://www.openshades.com

[41] Quest Visual. 2014. Word Lens. (2014). https://itunes.apple.com/us/app/word-lens/id383463868?mt=8 https://itunes.apple.com/us/app/word-lens.

[42] Kai Wang, Boris Babenko, and Serge Belongie. 2011. End-to-End Scene Text Recognition. In *IEEE International Conference on Computer Vision*. 1457–1464.

[43] WHO. 2014. Visual impairment and blindness. (2014). http://www.who.int/mediacentre/factsheets/fs282/en/

[44] Wikipedia. 2016. Google Glass. (2016). https://en.wikipedia.org/wiki/Google_Glass

[45] Hongzhi Yin, Bin Cui, Zi Huang, Weiqing Wang, Xian Wu, and Xiaofang Zhou. 2015. Joint Modeling of Users' Interests and Mobility Patterns for Point-of-Interest Recommendation. In *Proceedings of the 23rd ACM International Conference on Multimedia (MM '15)*. ACM, New York, NY, USA, 819–822. https://doi.org/10.1145/2733373.2806339

[46] Xianjun Sam Zheng, Cedric Foucault, Patrik Matos da Silva, Siddharth Dasari, Tao Yang, and Stuart Goose. 2015. Eye-Wearable Technology for Machine Maintenance: Effects of Display Position and Hands-free Operation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2125–2134.