Augmediated reality system based on 3D camera selfgesture sensing

Raymond Lo, Alexander Chen, Valmiki Rampersad, Jason Huang, Han Wu, Steve Mann
Department of Electrical & Computer Engineering, University of Toronto, Toronto, Ontario, Canada.
http://www.eyetap.org

***Abstract*—Three-Dimensional (3D) range cameras have recently appeared in the marketplace for use in surveillance (e.g. cameras affixed to inanimate objects) applications. We present FreeGlass™as a wearable hands-free 3D gesture-sensing Digital Eye Glass system. FreeGlass comprises a head-mounted display with an infrared range camera, both connected to a wearable computer. It is based on the MannGlas™computerized welding glass, which embodies HDR (High Dynamic Range) and AR (Augmented/Augmediated Reality). This recontextualizes the 3D range camera as a sousveillance (e.g. cameras attached to people) camera. In this sousveillance context, the range camera is worn by the user and shares the same point-of-view as the user. Computer vision algorithms therefore benefit from the use of the range camera to allow image segmentation by using both the infrared and depth information from the device for 3D hand gesture recognition system. The gesture recognition is then accomplished by using a neural network on the segmented hand. Recognized gestures are used to provide the user with interactions in an augmediated reality environment. Additionally, we present applications on serendipitous gesture recognition system in everyday life.**

I. Introduction

In recent years, gesture-based controls have been incorporated into various mobile devices such as smartphones and tablets [1], [2], [3]. Most of these devices rely on the multi-touch surface as their gesture interfaces. Other gesture recognition systems, such as the Microsoft Kinect, utilize an infrared range camera as the input device [4], which provides the user with "hands-free" input (not needing to hold any devices), via gestures. However, these devices, whether they require physical interaction or not, are usually external to the user. The user interacts with the device from a third person perspective. For example, consider the Microsoft Xbox Kinect. It functions as a surveillance camera, i.e. as part of the user's environment rather than as part of the user (sousveillance). Both the user and the Kinect can be considered separate entities in this interaction - once the user walks away from the Kinect, or because the Kinect is not always on and always with the user, there is no constancy of interaction. Essentially, these devices we use in some aspects of our everyday lives are not integrated with us for use in all aspects of our lives.

The principle of Humanistic Intelligence (HI) [5] and Natural User Interfaces [6] can be used to overcome this separation of user and device. That is, by using a wearable computer, there is no separation of device and user - the user is part of the feedback loop to the device. The wearable device is always ready to accept input of the user, regardless of time

Fig. 1: In one practical embodiment, FreeGlass comprises a 3D camera such as the ASUS Xtion (PrimeSense range camera), or a true time-of-flight 3D sensing camera, and a head-worn display such as the Epson Moverio BT-100 (head-mounted display), both connected to a wearable computer such as the ODROID-X2 "mobile computer". The resulting system provides for self gesture-sensing augmediated reality applications. The range camera is mounted onto the display and views the world from the user's point of view, aligning with the displayed subject matter.

or space [7], [8], [9], [10]. Therefore, by augmenting an HI wearable system with an infrared range camera, the user can gain 'hands-free' natural interaction with the system via gestures.

A. Augmediated Reality

FreeGlass can also help people see/sense our environment better, through Augmediated Reality. Augmediated is a portmanteau of augmented and mediated, referring to an ability not merely to add overlays (augment) but also to subtract (e.g. deliberately diminish) or modify. Examples include the MannGlass^{TM}welding helmet that diminishes the bright light of the arc and simultaneously augments the darker areas of the scene, in addition to providing computerized overlays to annotate a workpiece being welded [11], [12], [13]. Furthermore, the ability to sense and process depth information via the wearable Digital Eyeglass can even be beneficial to people who have problem seeing, for example, we can turn the eyeglasses into a navigation tool for the blinds or visual impairs [14].

II. 3D Hand Gesture Recognition

Hand gesture recognition consists of two main components:

1) Hand detection
2) Gesture recognition.

Hand detection concerns about how to robustly determine the contour of the hand in an environment with complex background; while gesture recognition is concerned about correctly interpreting the meaning of a gesture.

To achieve hand detection, many researchers take advantage of controlled environments, such as constant lighting and static background [15], [16]. However, these methods are not reliable in real world environments with complex lighting and background changes. Particularly, our proposed system utilizes a wearable camera that is always moving, and thus the assumption of having a static background is often not applicable. Other methods focus on tonal based features, such as skin color segmentation [17]. These features are not robust against dynamic lighting condition and non-static backgrounds, for example, similar colours between the background and human skin. In addition, some methods use specially coloured gloves or other sensing device such as the data glove to provide additional information for the segmentation [18]. Understanding the problems of the methods discussed above, we explore an alternative method based on the depth information provided by an infrared range camera, such as a PrimeSense camera, to perform close range hand detection.

The PrimeSense camera computes the depth map which contains information of an object's distance with respect to the camera. The depth map can be considered as an additional dimension of information for feature extraction and image segmentation [19], [20]. Most of the current approaches use only an infrared range camera from a third person perspective. The solution assumes there is no confusion between the hands depth information with other obstacles in the environment. Besides the infrared range camera, some approaches use a combination of a single color camera, a stereo color camera and a thermal camera to obtain additional information for image processing and image denoising [21]. These methods achieve promising results in the static indoor setting.

Other gesture recognition devices such as the Leap Motion controller are designed to capture hand gestures from the bottom up perspective [22]. Since this device is not equipped with a RGB camera, and thus not the ideal candidates for performing augmented or mediated reality applications when the gesture command is recognized. OpenNI consists of a set of open sourced API that allow us to capture natural interaction between human and computer via PrimeSense cameras [23]. Algorithms such as skeleton tracking can effectively track human body and its parts utilizing the depth map information [24]. However, in our application, gesture commands are performed in a very close range setting since the camera is mounted on the users head. For this reason, the gesture command are not recognizable by the depth map algorithm provided by the OpenNI framework in our case.

Lastly, other 3D cameras such as true time-of-flight camera from SoftKinetic[1] can be used to perform the segmentation and extraction of hand features with our proposed system (see Figure 8). However, these sensors are designed for short range depth extraction and thus lacking the ability to sense the environment for augmediated reality purposes. The current limitations and the future direction of a novel hybrid sensor approach will be further discussed in the future work section.

A. Proposed Method

For a mobile or a wearable platform, we attempt to minimize the number of devices in the system and instead of performing gesture recognition using PrimeSense camera from a third person view, where the camera observes the users gestures on a steady platform [25], we propose to use the camera from the first person perspective, where it is mounted on the user's eye glasses and observe the world from the user's point of view [14]. Therefore, a wearable construct based on the PrimeSense camera is of interest, which has appeared in the use of the navigation helmet proposed by Steve et. al [14].

Similar to methods [25], [17], [19], [20], we achieve the gesture recognition in two stages:

1) segmentation
2) classification

The purpose of the segmentation stage is to first locate the hands of the user in the image. We apply the classification algorithm to segmented image to identify the gestures.

B. Segmentation

In order for the system to classify the gesture, it needs to first identify the regions which contains user's hand(s). With our unique configuration, we can assume the hands appear as objects within close proximity to the camera. This information can be obtained from the range camera sensor, like

[1] http://www.softkinetic.com/fr-be/products/depthsensecameras.aspx

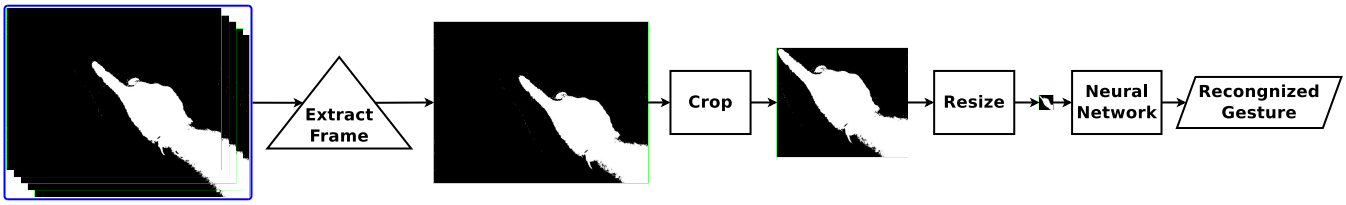


Fig. 2: The masked images are cropped, downsampled, and then processed through the neural net to determine the gesture.

the PrimeSense camera or the like. The PrimeSense camera provides two types of images:

1) Infrared image
2) Computed depth map

The infrared image is a gray scale image that shows the level of infrared light sensed by the camera. The depth map is provided by the camera which approximates the distance of the objects in the scene. The two images are filtered independently to remove pixels that do not meet the constraints / thresholds. The results are two binary images that intersect to produce the final image mask, a binary image for hand extraction as shown in Figure 3.

In one of our embodiment, due to device limitations, the depth map can only return a finite range of distance values. This is a known hardware specification in the long range sensors where the IR laser projector overpowers (i.e., overexpose) the subjects that are close range. A depth map pixel is set to zero if the viewing object is either too close or too far from the camera. Additionally, the distance of any light source or reflective material in the scene that corrupts the projected pattern is unknown and set to zero. With the camera worn on the user's head, we assume that the gestures appear within the distance range up to the fully stretched arm length away from the camera. This means that objects with depth values under certain threshold d_{th} are considered the candidates of the user's hand. However, this includes false candidates such as light sources, shadows, reflective objects, and distant objects. The resulting binary image sets the pixels under d_{th} to one and others set to zero.

Since the PrimeSense camera projects the patterns in the infrared spectrum, given the condition that no other infrared light source is present, the objects closer to the camera are relatively brighter than the objects from afar. We assume this property even with other light sources or highly reflective materials are present in the scene. With this assumption, a binary image based on the infrared image is created by applying a threshold to the pixel values. Denote p_{th} as the pixel threshold, we set the pixels below p_{th} to zero and others to one.

The intersection of the two binary images is performed to generate the mask. The binary image of the infrared image filters out the distant objects that would appear as candidates in the binary image of the depth map, while the binary image of the depth map filters out the pixel intensities greater than p_{th} that are too far from the camera, as shown in Figure 3.

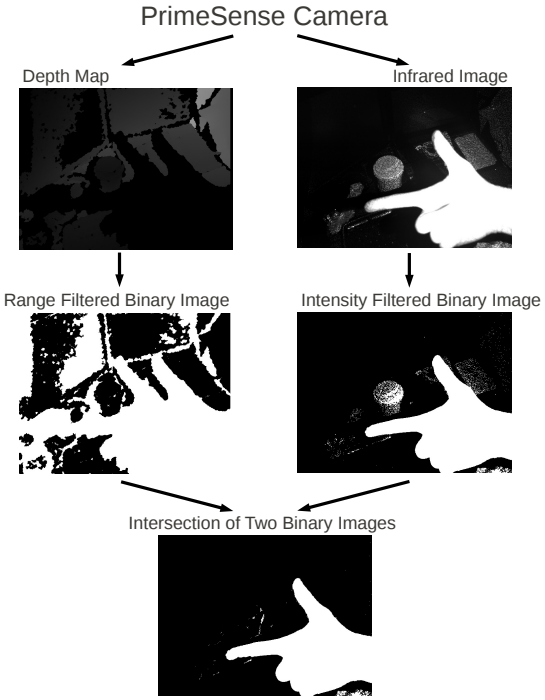


Fig. 3: Image segmentation steps. The binary image on the left sets pixels to one if the depth map is unable to identify the object's relative distance. The binary image on the right filters out the lower than threshold pixels by setting them to zero. The intersection of the two binary images becomes the image mask for gesture recognition. Notice that there are still noises present in the image mask. This happens when both binary images do not filter out the out-of-range pixels. For example, a close distance bright light source is both unidentified in the depth map and is high in pixel values in the infrared image.

To extract the hands from the image mask, we resort to fitting bounding boxes on the extracted contours. Typically, the two hands are the largest objects in the image mask. Therefore, we apply this heuristics of finding only the objects that are bounded by the two largest boxes. The two largest objects become the candidates for gesture recognition.

With the 3D true-time-of-flight camera, we can perform similar algorithm by removing extracting the user's hand based on the distance information. To reduce false-positive, the same heuristic could be applied as the user's hands are the closest object to the users and often emerged from the side of the frame (i.e., the hand contour creates a continuous curve that connect to the side of the frame).

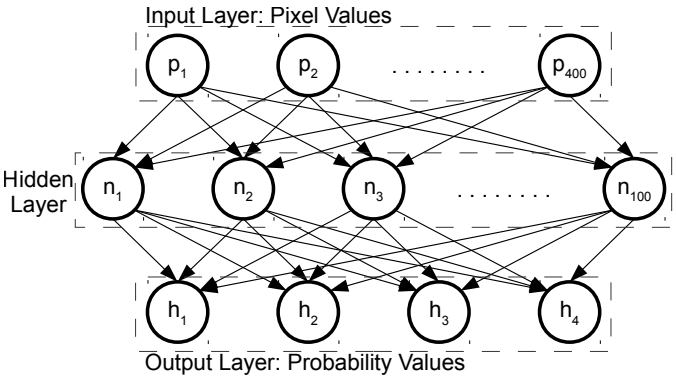


Fig. 4: The neural net implemented, takes 400 pixels at the input layer; has 100 nodes in the hidden layer; and 4 output nodes. Each node represents the confidence of the input being a specific gesture.

(a) Point Up (b) Lower Right (c) Point Angled (d) Upper Left

Fig. 5: Sample masked images of the 4-gestures trained into the neural net. During the classfication of the gesture, the system will recognize the two gestures: point-angled and point-up as finger pointing. This helps increasing the flexibility of gesture recognition for the users to post gestures that are natural to them.

C. Classification

We use a single layer neural network to achieve real time gesture recognition. The extracted image mask of the hands is downsampled to a 20×20 pixels image. This image is fed into the neural network, and the neural network outputs the probability of each gesture. Each pixel in this image patch is treated as an input unit as shown in Figure 4. Therefore, our input vectors to the neural network are always 400 to 1. For the hidden layer, we choose to only implement 100 hidden units. By choosing a small number for the hidden units, we are able to limit our total parameter size to 40400. We decided this number is an efficient use of computational resource for a real time recognition system on a wearable battery-powered system. Finally, we have 4 output units since there are 4 different possible gestures we are interested in, as shown in Figure 5. Each of these output units is the probability of an unique gesture.

To train our neural network, we first need to define the cost function. This function is the log likelihood of logistic regression. To find the best possible parameters for the model, we suppose to find find the parameter which will maximize this function. However, due to our gradient descent setting, we decided to add a minus sign in front of it and make it a minimization problem. Therefore, we are trying to maximize the log likelihood function using minimization techniques. To prevent over fitting to the training data, we added a regularization term by adding the square of each parameter at the end of the cost function. These regularization terms will punish the cost function as the parameters become too big, which can result in a floating point overflow. The training cost function $J(\theta)$:

$$J(\theta) = l(\theta) + R(\theta, \lambda) \tag{1}$$

The term $l(\theta)$ is the logistic regression for minimization:

$$l(\theta) = -\frac{1}{s}\sum_{i=1}^{s}\sum_{j=1}^{c}[y_j^{(i)} log(h_\theta(x^{(i)}))_j + (1 - y_j^{(i)}) log(1 - (h_\theta(x^{(i)}))_j)] \tag{2}$$

for which s denotes the total number of training cases and c denotes the total number of output gestures. Since our objective of this function is to add up the cost from each of our training cases. Thus, we use i to denote the current training cases that are being used to calculate the cost. $h_\theta(x^{(i)})$ denotes the estimation resulted from the forward propagation. After calculating the the estimate from forward propagation, we use a logistic function to rescale that number between 0 and 1.

The term $R(\theta, \lambda)$ is the regularization term:

$$R(\theta, \lambda) = \frac{\lambda}{2s}[\sum_{i=1}^{n}\sum_{j=1}^{p}(\theta_{i,j}^{(1)})^2 + \sum_{i=1}^{c}\sum_{j=1}^{n}(\theta_{i,j}^{(2)})^2] \tag{3}$$

for which n denotes the total number of nodes in the hidden layer and p denotes the total number of nodes in the input layer, which is also the number of pixel we have in each of our training image patch.

D. Training

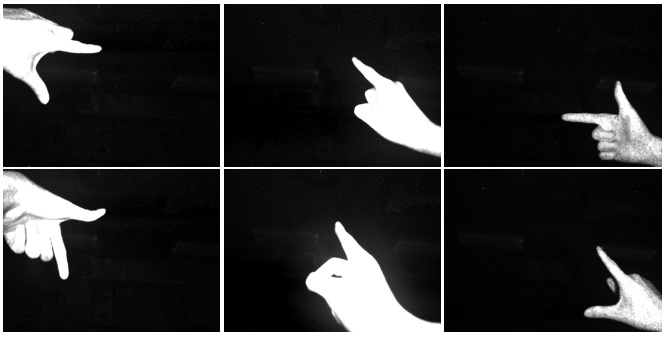
The training data were collected using the PrimeSense camera to record a sequence of the image masks of various hand gestures. In particular, we focus on the following gestures:

- the framing gestures (consists of both hands that form the corners in diagonal of each other)
- the finger pointing gesture.

1) Gesture Variation: One problem associated with gesture recognition is that the the orientation or form of a single gesture varies, with respect to the user and instance. Specifically, we consider two types of variations: the variations due to change in orientation [19], [20], [25] and variations due to different forms of gesture that represent the same action.

Figure 6 shows some gestures that have the same meanings. The differences of these forms of gestures are not mere geometric transformation from one to another. To adapt to the form variations, we first define a group of different gestures that mean the same action. Each gesture of the same group is trained separately.

In addition to the form variations, we also attempt to train for the variations in orientation. This allows recognition system to adapt to slight angle changes of the hand. The inclusion of the variations helps the training to account for the gesture differences, which avoids limited recognition of only a single instance of the gesture.



(a) Upper Left (b) Point (c) Lower Right

Fig. 6: Demonstration of the variations of gestures. The top row is one instance of three different gestures and the lower row is examples of alternative gestures of the top row.

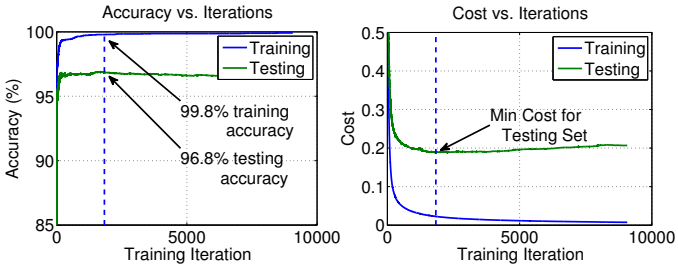


Fig. 7: A graph of the Cost function versus Training iteration. The graph shows the iteration at which to stop training the neural net - the minimum point of the testing cost. Beyond this iteration, more training causes an increase in the testing cost. At that iteration, the training set achieves a 99.8% accuracy and the testing set achieves 96.8% accuracy.

2) Data Collection: Collecting a large amount of training data is one of the most effective way to improve the performance of a learning algorithm. In our setting, we could collect sample data simply recording additional gesture samples in our daily use of the device. Although we are achieving high accuracy on our existing training data. We are constantly streaming our gestures and give label them with the correct label. This data collection approach will keep improving our learning algorithm in continuous usage.

3) Early Stopping: In order to avoid over fitting to our training data. We separated 80% of our data as our training data and 20% of our data as test data. On every iteration of neural net training, we run forward propagation to get our gesture prediction accuracy and cost on both training and test set. We plot the cost on both training and test sets versus the number of training iterations as shown in Figure 7. As you can see in the Figure 7, at around iteration 2000, the cost of the test data starts to increase while the cost of the training data is still decreasing. This implies that after approximately 2000 iterations, our neural net is being over trained to the training data, that is, if left to train forever, the neural network will only match items in the training data and reject everything else.

Fig. 8: Another example of FreeGlass which combines the 3D sensing camera from SoftKinetic and the transparent digital eye glass display from EPSON. The camera is mounted onto the headed mount display with our custom 3D printed parts.

III. Proposed Hardware and Implementation

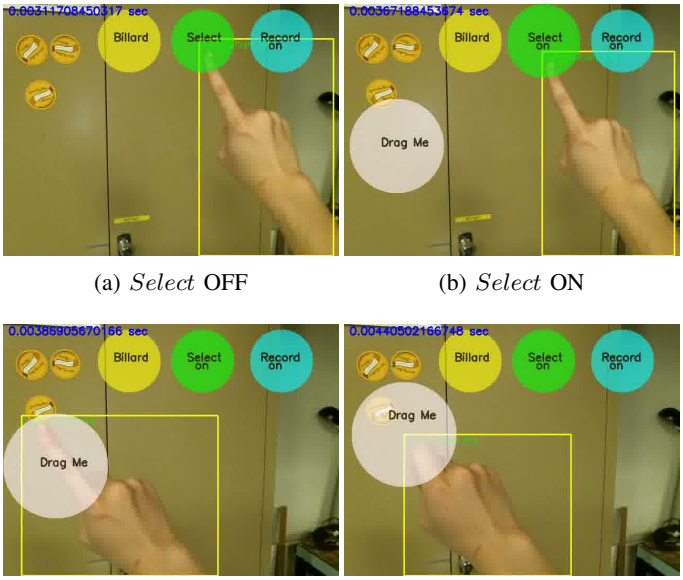
In this project, our goal is to create and implement an aug-mediated reality system using gesture recognition as a user interface. To achieve this, we utilize the 3D sensors (ASUS Xtion / SoftKinetic) to observe the world and gestures from a first person view (see Figure 8 and 11). The ASUS Xtion is a PrimeSense based range camera which provides depth maps and infrared images of the scene it is observing. This camera uses an infrared projector and infrared camera to determine the depth map. The images are processed in real time with an ODROID-X2, which is an ARM-based mobile development platform with a 1.7GHz ARM Cortex-A9 Quad Core processor. Finally we display the result using the Epson Moverio BT-100. The Epson Moverio BT-100 is a head-mounted display that uses transparent screen. Based on the principles discussed in [5], Epson's Moverio is a good candidate for mediated reality applications due to its special display allows users to interact with physical world with less eye straining issues. The Moverio is capable of streaming from an external video source and was therefore used as a display for the processed information from the range camera. In this project, we processed the range camera information with ODROID-X2 and added additional mediated reality information to the Moverio. The user will see a mediated reality, such as a mediated user gesture interface that will interact with real world object.

A. Performance

The training stage of our neural network achieved an accuracy of 99.8%. The cross-validation of the trained neural network achieved an accuracy of 96.8%. The performance in frames-per-second (FPS) on the ODROID-X2 is 25 FPS.

IV. Applications

Our proposed wearable eyeglasses system enables users to perform gesture-based control in everyday life. Unlike other



(a) *Select* OFF (b) *Select* ON

(c) Initiating $DragMe$ (d) Moving $DragMe$

Fig. 9: Some sample interactions using the wearable gesture recognition system. Fig 9a and Fig 9b shows a gesture toggling a virtual-button. Fig 9c and Fig 9d shows a gesture moving a virtual object in the scene.

mobile devices, our system is always-ready and it continuously processes information about the environment. Such feature enables novel applications, such as interactive QR-based infographic interface, which automatically acquires/mediates information about our environment and require minimum user intervention. Furthermore, the camera system also bring forth social implications that may arises as with other camera-enabled systems.

A. Augmediated Reality Interface with Real Physical Objects

The wearable 3D eyeglasses and 3D range sensors provide an novel interface for an immersive interaction with the physical world. Instead of solely based on augmented reality, which only add augmented elements in to the real world scene, our gesture recognition system along with the wearable 3D glasses can efficiently detect the surrounding environment and perform aug-mediated application, which involves adding and subtracting information in the real world scene and enable more interactivity between the user and computer. Users are able to see the real world environment along with the user interface. In Figures 9a and 9b, the user is able to toggle a $Select$ function by overlaying his/her 'finger pointing gesture' over the $Select$ button. Unlike traditional devices, where the users have to control the device with their hands, our approach gives the user a more intuitive and hands free control over the device.

B. Mediated Reality Window Management

Mediated reality window management is another important application for our gesture recognition system. Users will be able to drag around an augmented window as you can see in Figures 9c and 9d. This window can contain various information such as conference calling, web pages and GPS maps. This application allows user to do their daily desktop/laptop environment task directly in a first person mediated reality window.

C. Region of Interest Selection With Gesture

In most image based object recognition tasks, it is often difficult to determine which object of interest in a photo a person would like to classify on without any user inputs. To reduce the search area for object classification, methods such as eye tracking [27] and hand gestures are natural ways for the user to inform the system where the user's focus is. For example, if the scene contains a chair and a table and the person wants to see the price of that chair, without any additional indication from the user, the recognition system may only attempt to retrieve the price information of both items and display them. With the gesture recognition enabled in the system, the user is able to guide the system of a specific object of interest. This is done by first constraining the area of the view using the bounding box to bring forth the region of interest. The user may naturally select the region of interest by posting two-corner gesture, as seen in Figure 10.

In addition to utilizing our gesture recognition system as a preprocessing tool for object recognition, we integrate the use of QR (Quick Response) code to improve the accuracy of object recognition and to speed up recognition performance. QR code has been used extensively over the past few years in terms of advertising on mobile platforms due to its simplicity, robustness, and speed. People can look at a poster and scan the QR code and get redirect to the event website. In our wearable gesture recognition setting, it provide a natural experience when working with QR code technology. A user can be walking down the aisle in a grocery store and acquire product information by scanning QR codes. To activate the QR code and get additional information on the products such as ratings, the user can perform a gesture command to segment out the QR code of the interested product. Once the QR code are segmented and recognize, our wearable system will send a request to an URL and display the corresponding information.

V. Social Implications

Serendipitous gesture recognition on a mobile wearable device requires constant sensing of the environment. For example, the vision based wearable gesture recognition system (Figure 11) continuously processes every frame of incoming video so that the system is ready to recognize any hand gesture when it occurs. The system might remain dormant for an extended time, and then be "awakened" by a simple hand gesture that gives commands corresponding to the gesture the FreeGlass system recognizes. By definition, this particular setting will be classified as a sousveillance system [28], [29], [30], which is the opposite of surveillance, i.e. instead of a camera system mounted to a building, the camera is human-centric, and the world through this device is captured as a first person view. There has been recent debate over the appropriate

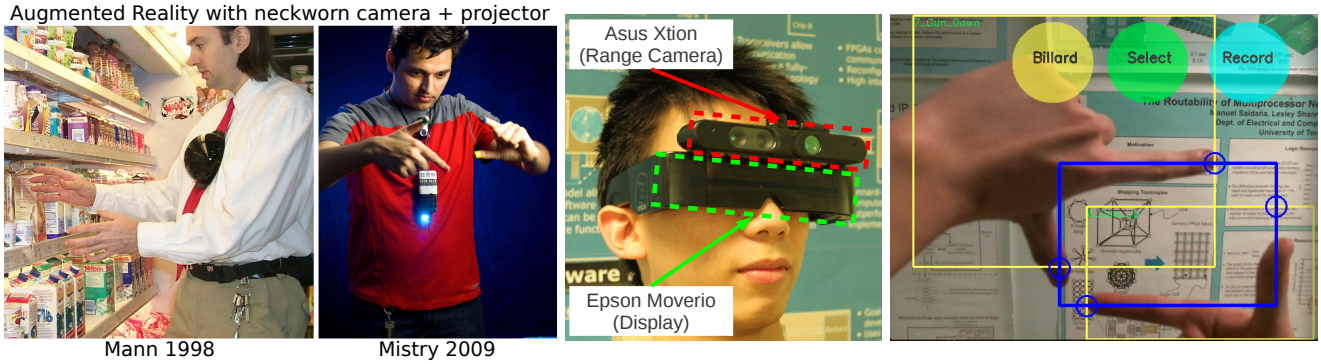


Fig. 10: Gesture-based wearable computing. Leftmost: Two examples of "Synthetic Synesthesia of the Sixth Sense"[26], commonly abbreviated as "Sixth Sense" or "SixthSense" which is a wearable computing system with various sensors. The system pictured here is capable of augmented reality without the need for eyewear. Other variations of SixthSense utilize a finger marker and RGB camera to capture the gestures in order to project augment reality frames on a surface [10]. Rightmost: In addition to Augmented Reality, FreeGlass also allows Augmediated Reality, i.e. a partially mediated reality that can, within the limits of the dark shade, selectively augment or diminish various aspects of the scene. Here for example, the system recognizes a two-corner gesture to overlay a bounding box (blue frame) on top of the wearer's view. The blue circles indicate the location of the finger tips. An application of such a gesture is to select a region of interest from the wearer's view.

(a) (b) (c)

Fig. 11: Applications of the collaborative gesture-based interface among multiple eyeglasses. Our proposed hardware system allows multiple users to interact with real physical objects, such as QR code on the building, sketchbook, or boxes. Users experience a shared of virtual and physical space through the eyeglasses, and our gestures interface provides an intuitive methods for real-time interaction.

use of sousveillance devices such as EyeTap, MannGlas, and Google Glass. On one hand, some people see sousveillance devices as a threat to personal privacy that should be forbidden. But many spaces already use surveillance, and as such, sousveillance is seen by many as creating a necessary balance in an otherwise one-sided "surveillance society".

Wearable Computing devices provide a revolutionary form of personal assistance, augmented reality/mediated reality and the like, [31], [32].

The interactive mediated reality building Figure 12 is a social experiment and artistic in(ter)vention conducted by author S. Mann to address the awareness of the use of sousveillance devices. Photography and cameras are examples of sousveillance. As common as such practices and devices are, there are still occasions when photographers are harassed by security staff or police officials. For example, persons with sousveillance devices are unwelcome in some commercial establishments where numerous surveillance cameras are installed. In this sense, surveillance is often the veillance of hypocrisy. To raise the awareness of problems with this one-sided veillance, the interactive mediated reality building was designed to take photos of the photographer as the photographer scans the QR (Quick Response) code installed on the building. As the photographer scans the QR code, he/she will see an augmented reality sign indicating there is no photography allowed of the building. However, in order to scan the QR code and see this NO PHOTOGRAPHY sign, the photographer must have already taken photos of the building. In return, the building will capture a photo of the photographer since he/she has violated the signs displayed by the building.

In the past, people have implemented wearable gesture recognition algorithms based on a processing RGB frames fed from the sousveillance devices. To perform gesture recognition, these devices has to process every RGB frame in order to detect if there is any gesture command given by the user. As an example, the SixSense project uses a RGB camera to capture user's gesture command by detecting the color finger marker taped on the user's fingers. Another good example will be the telepointer [33], which feeds RGB frames in order to provide a visual collaborative experience to the user. However, both projects requires constant polling on the RGB frame by the device, which has the potential to capture sensitive information of other surrounding people or environment. Thus, it has the

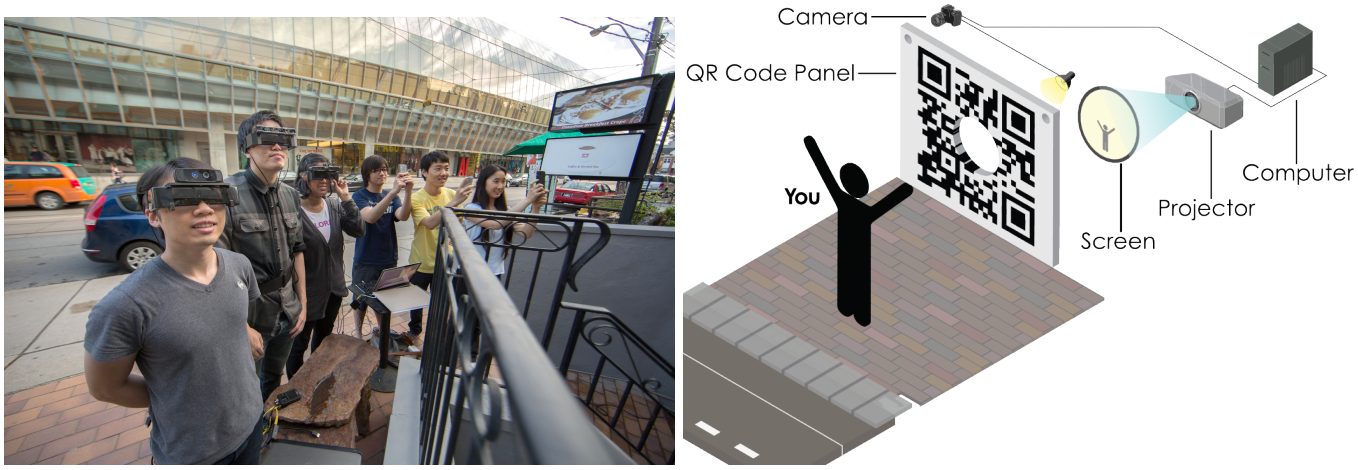


Fig. 12: Interactive QR code on a building. Users can interactive with the QR code sign by either capturing photos of the sign with a mobile phone or wearing the propose eyeglasses system. When an image of the sign is taken, the interactive QR code will automatically capture images of the audience/photographer with a flash (fire-back) and project the image onto the screen. With the wearable eyeglasses, the QR images will be scanned and information about the building will be retrieved automatically. Such interaction enables serendipitous gesture recognition. The sign also explored the notion of sousveillance and surveillance to allow photographers to understand the "No photography" and rises the awareness of sousveillance in our daily life (i.e., the contradiction of no photography is allowed when camera/vision system is often required to accomplish tasks in our everyday life).

potential to cause privacy intrusion.

To avoid these privacy concerns caused by RGB based wearable gesture recognition system, our system uses an infrared (only) camera for gesture recognition. Instead of polling for the RGB frames, we perform constant polling for the infrared frames. This approach not only achieves promising gesture recognition accuracy, but also avoids capturing sensitive information during its operation.

Hand gesture is a form of expression. The presence of gesture helps to signal the person's action and its intention. A hand gesture controlled sousveillance device forces user's expression to both the camera of the system as well and the people around the user who are aware of such device. This deliberate act of expression as a command is a way to notify the crowd of user's action, which may reduce people's suspicion on the misuse of the sousveillance system. This is because the gesture commands have to be visible to the camera, in line of sight with the other subject matters that exist in the same view. Thus, the user has to be conscious of the consequence on his or her action to the surrounding being while commanding the sousveillance device. This may lead to a healthier veillance community for all users by reducing the anxiety caused by suspicion of privacy intrusion.

Aside from the use of camera technology, the hand gestures also pose some social implications. In more general terms, hand gestures is a form of expression and communication in our world. As each culture/country has its own spoken language, so too does each have its own set of hand gestures with their own meanings - there is no "universal language" for gestures [34]. For example, the "thumbs-up" gesture which carries a meaning synonymous to "good-luck" in North America, in Iran, it carries a derogatory meaning - similar to the "middle-finger" in North American culture. Thus, if a wearable system is designed with a fixed set of gestures, it is possible that these gestures have different meanings in different cultures - some of which may even be derogatory. Therefore, having a fixed set of gestures can affect the global acceptance of a gesture-based wearable computer unless the gestures and their meanings become universally accepted. For example, the "upper-left" and "lower-right" gesture together (as shown in Figure 10), can be a global gesture for taking a picture in the selected area of view. Another solution is to design the gesture system to account for the cultural differences, that is, the gesture system is localized by country. However, whenever a user travels to a different country, the user will have to relearn the gestures required to interact with their wearable computer, and this can be inconvenient for the user.

VI. Future Work

In further development of the system, we are experimenting and expanding on our current base to incorporate more gestures to our system and create more ways for the user to interact with the environment in first person perspective. For example, we are currently developing a sport aid system that will help a pool player improve their skills. We are incorporating new gestures in this application to enable a user to segment the relevant ball and find the optimal path to hit the ball into the bag.

On the other hands, more advanced 3D sensors, namely an hybrid approach which takes the advantages of the short range SoftKinetic's time-of-flight 3D sensors and the PrimenSense's structural light IR laser 3D sensors, will provides a more robust and rich user experiences. The miniaturization eyeglasses and these sensors components also plays an important role in having such device to be used in consumer market, and allow further research to be explored in a larger scale, for example,

the 'No photography' experiment in a larger settings.

VII. Conclusion

We have proposed a 3D hand gesture recognition wearable system utilizing the ASUS Xtion PrimeSense range camera. We process information from the range camera, in real time, to recognize hand gestures. Once we get the user input through the range camera, we display the interaction, such as the corresponding action due to a gesture, back to the user via the Epson Moverio BT-100. We trained a neural network learning algorithm to learn 4 different gestures (5) that we are interested in to demonstrate the first prototype of our gesture interface and achieved 99.8% training accuracy, 96.8% testing accuracy, and we are able to run our recognition system at interactive frame rate (20-25fps) on an ODROID-X2 mobile computer.

Acknowledgment

The authors would like to thank Epson, ODROID and ASUS for their supports to this project.

References

[1] Google-Android. Gestures. [Online]. Available: http://developer.android.com/design/patterns/gestures.html

[2] B. D. Resource. Introduction to touch gestures. [Online]. Available: http://supportforums.blackberry.com/t5/Java-Development/Introduction-to-Touch-Gestures/ta-p/555363

[3] BlackBerry. Touch screen gestures - how to demo - blackberry z10. [Online]. Available: http://demos.blackberry.com/blackberry-z10/na/us/gen/how-to/your-blackberry-z10-smartphone/blackberry-z10-overview/touch-screen-gestures/index.html

[4] W. REDMOND and I. TEL AVIV. (2010) Primesense supplies 3-d-sensing technology to "project natal" for xbox 360. [Online]. Available: http://www.microsoft.com/en-us/news/press/2010/mar10/03-31primesensepr.aspx

[5] S. Mann, "Wearable computing: Toward humanistic intelligence," *Intelligent Systems, IEEE*, vol. 16, no. 3, pp. 10–15, 2001.

[6] ——, *Intelligent image processing*. IEEE-Wiley, 2001.

[7] ——, "'wearcam'(the wearable camera): personal imaging systems for long-term use in wearable tetherless computer-mediated reality and personal photo/videographic memory prosthesis," in *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*. IEEE, 1998, pp. 124–131.

[8] ——, "Wearable computing: A first step toward personal imaging," *Computer*, vol. 30, no. 2, pp. 25–32, 1997.

[9] ——, "Wearable, tetherless computer–mediated reality: Wearcam as a wearable face–recognizer, and other applications for the disabled," in *Presentation at the American Association of Artificial Intelligence, 1996 Symposium. Retrieved January*, vol. 21, 1996, p. 2002.

[10] P. Mistry and P. Maes, "Sixthsense: a wearable gestural interface," in *ACM SIGGRAPH ASIA 2009 Sketches*. ACM, 2009, p. 11.

[11] R. C. H. Lo, S. Mann, J. Huang, V. Rampersad, and T. Ai, "High dynamic range (hdr) video image processing for digital glass," in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 1477–1480.

[12] S. Mann, R. Lo, J. Huang, V. Rampersad, and R. Janzen, "Hdrchitecture: real-time stereoscopic hdr imaging for extreme dynamic range," in *ACM SIGGRAPH 2012 Emerging Technologies*. ACM, 2012, p. 11.

[13] S. Mann, R. C. H. Lo, K. Ovtcharov, S. Gu, D. Dai, C. Ngan, and T. Ai, "Realtime hdr (high dynamic range) video for eyetap wearable computers, fpga-based seeing aids, and glasseyes (eyetaps)," in *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*. IEEE, 2012, pp. 1–6.

[14] S. Mann, J. Huang, R. Janzen, R. Lo, V. Rampersad, A. Chen, and T. Doha, "Blind navigation with a wearable range camera and vibrotactile helmet," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 1325–1328.

[15] K. Imagawa, S. Lu, and S. Igi, "Color-based hands tracking system for sign language recognition," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. IEEE, 1998, pp. 462–467.

[16] P. Hong, M. Turk, and T. Huang, "Gesture modeling and recognition using finite state machines," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. IEEE, 2000, pp. 410–415.

[17] R. Kjeldsen and J. Kender, "Toward the use of gesture in traditional user interfaces," in *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*. IEEE, 1996, pp. 151–156.

[18] D. Sturman and D. Zeltzer, "A survey of glove-based input," *Computer Graphics and Applications, IEEE*, vol. 14, no. 1, pp. 30–39, 1994.

[19] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 1093–1096.

[20] D. Uebersax, J. Gall, M. Van den Bergh, and L. Van Gool, "Real-time sign language letter and word recognition from depth data," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 383–390.

[21] J. Appenrodt, A. Al-Hamadi, and B. Michaelis, "Data gathering for gesture recognition systems based on single color-, stereo color-and thermal cameras," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 3, no. 1, pp. 37–50, 2010.

[22] L. Motion. Leap motion. [Online]. Available: https://www.leapmotion.com/product

[23] OpenNI. About openni — openni. [Online]. Available: http://www.openni.org/about/

[24] PrimeSense. Nite middleware - primesense. [Online]. Available: http://www.primesense.com/solutions/nite-middleware/

[25] Z. Li and R. Jarvis, "Real time hand gesture recognition using a range camera," in *Australasian Conference on Robotics and Automation*, 2009, pp. 21–27.

[26] S. Mann and H. Niedzviecki, "Cyborg: Digital destiny and human possibility in the age of the wearable computer," 2001.

[27] M. Schiessl, S. Duda, A. Thölke, and R. Fischer, "Eye tracking and its application in usability and media research," *MMI-interaktiv Journal*, vol. 6, pp. 41–50, 2003.

[28] S. Mann, "Sousveillance: inverse surveillance in multimedia imaging," in *Proceedings of the 12th annual ACM international conference on Multimedia*. ACM, 2004, pp. 620–627.

[29] S. Mann, J. Fung, and R. Lo, "Cyborglogging with camera phones: Steps toward equiveillance," in *Proceedings of the 14th annual ACM international conference on Multimedia*. ACM, 2006, pp. 177–180.

[30] S. Mann, J. Nolan, and B. Wellman, "Sousveillance: Inventing and using wearable computing devices for data collection in surveillance environments." *Surveillance & Society*, vol. 1, no. 3, pp. 331–355, 2002.

[31] R. Hill, J. Fung, and S. Mann, "Reality window manager: A user interface for mediated reality," in *Proceedings of the 2004 IEEE International Conference on Image Processing (ICIP2004)*, 2004, pp. 24–27.

[32] C. Aimone, J. Fung, and S. Mann, "An eyetap video-based featureless projective motion estimation assisted by gyroscopic tracking for wearable computer mediated reality," *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 236–248, 2003.

[33] S. Mann, "Telepointer: Hands-free completely self-contained wearable visual augmented reality without headwear and without any infrastructural reliance," in *Wearable Computers, The Fourth International Symposium on*. IEEE, 2000, pp. 177–178.

[34] D. Archer, "Unspoken diversity: Cultural differences in gestures," *Qualitative Sociology*, vol. 20, no. 1, pp. 79–105, 1997.