Hand Gesture based Region Marking for Tele-support using Wearables

Archie Gupta, Shreyash Mohatta, Jitender Maurya,
Ramakrishna Perla, Ramya Hebbalaguppe, Ehtesham Hassan
Smart Machines R&D Group, TCS Research, New Delhi, India

Abstract

Wearable Augmented Reality (AR) devices[1] *are being explored in many applications for visualizing real-time contextual information. More importantly, these devices can also be used in tele-assistance from remote sites when on-field operators require off-field expert's guidance for trouble-shooting. For an effective communication, touchless hand gestures are the most intuitive to select a Region Of Interest (ROI) like defective parts in a machine, through a wearable. This paper presents a hand gestural interaction method to localize the ROI in First Person View (FPV). The region selected using freehand sketching gestures is highlighted to the remote server setup for expert's advice. Novelty of the proposed method include (a) touchless finger-based gesture recognition algorithm that runs on smartphones, which can be used with wearable frugal modality like Google Cardboard/Wearality, (b)reducing the network latency and achieving real-time performance by on-board implementation of recognition module. We conducted user studies that suggest the ease and usefulness of the proposed method. Further, we evaluated the the effectiveness of the ROI gesture using the PASCAL Visual Object Classes(VOC) criteria.*

1. Introduction

As mobile devices are equipped with increasingly powerful processors and high-speed network access, it is possible to explore mobile devices as a platform for AR applications. In this paper, we present an AR tele-assistance framework for remote assistance and remote scene exploration into a natural collaborative interface. This framework helps in reducing the high costs of having the expert on the site and also long downtimes due to travel. Applications of tele-presence include customer calling a call center to help troubleshoot a printer, video-conferencing, repair, maintenance and inspection in industries, wiring in aircraft and remote control of the machinery amongst others [28, 9]. Our tele-assistance framework involves an expert personnel present at server location, assisting the novice inspector at a remote location, by observing the video stream sent from user's head mount to the server via the network.

Touchless hand gestures are more effective to highlight the ROI in industrial outdoor setting, where audio communication is hard to comprehend owing to ambient noise. This helps the expert in understanding the problem and assist the on-site inspector via audio/text instructions, so as to solve the desired task accurately and quickly. Despite the availability of high-end sophisticated AR gadgets such as Microsoft Hololens, Daqri smart helmet[2] etc., our work focuses on frugal smartphone based head mounts such as Google Cardboard/Wearality due to its economically viable nature, portability and scalability to the mass market. Using the stereo-rendering of camera feed and overlaying the related information on the smartphone screen, these devices can also be extended to AR applications.

To achieve this, an efficient algorithm for hand gesture recognition in FPV, which focuses on highlighting the ROI in the Field-of-View (FoV) for smartphones, through a single monocular camera is proposed. This poses further challenges such as (i) lack of additional sensors like the depth sensing and IR sensors on smartphones, (ii) the hand portions may fall under different illuminations, the background might be static or noisy, and (iii) robust tracking of ROI approximated via a bounding box is required to point desired location correcting the motion due to ego centric view.

The summary of contributions are as follows:

- A novel hand gesture for highlighting ROI in FPV for frugal AR devices has been explored. This can enable the wider reach of frugal devices such as *Google Cardboard* and *Wearality* in AR.
- Our gesture recognition algorithm is explored for (i) the suitability to work with RGB channel stream or the

[1] Head Mounted Devices such as Google Glass, BT Moverio, and Vuzix

[2] https://daqri.com/,https://www.microsoft.com/microsoft-hololens/en-us

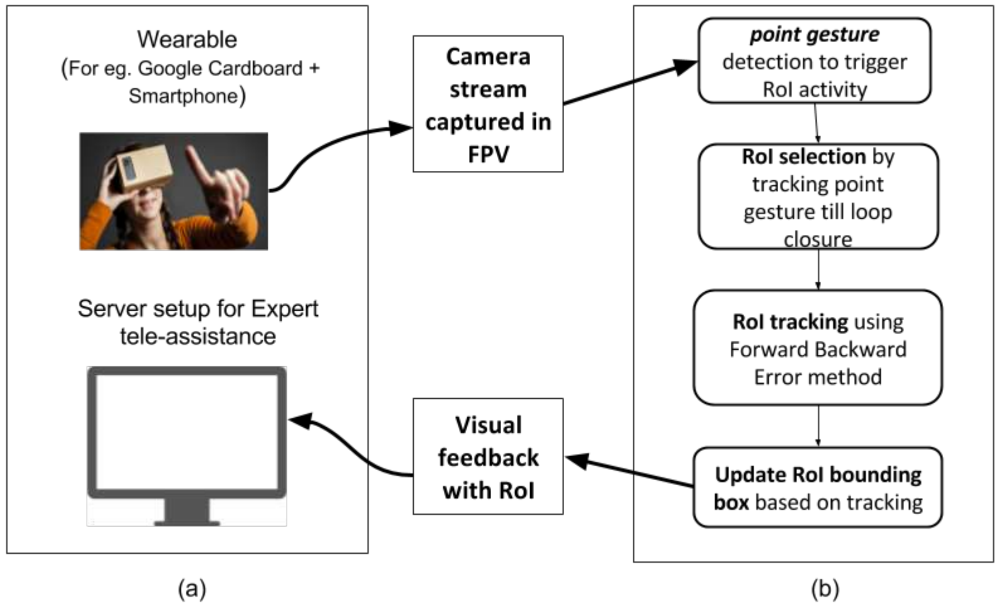

Figure 1. Our proposed tele-assistance setup: (a) showing user with wearable (for eg. a smartphone with Google Cardboard) performing ROI marking and server setup for expert feedback, and (b) high-level blocks of ROI recognition algorithm from ego centric view.

pixel data from a single smartphone monocular rear-camera without built-in depth sensors, (ii) the large scale deployments in real-time implementations without network dependency, and (iii) reliability and accuracy in dynamic outdoor settings.

2. Related Works

Several systems have proposed the use of tele-assistance in tasks involving manipulation of physical objects which shows that it is important to provide the remote expert with the user's FoV and maintain a shared context of the desired task to be performed [9, 8, 21]. Szalavri et al.[27] proposed a client-server architecture of multi-user AR to support operators and enhance the assistance of visualisation experts. Participants can see the same spatially aligned model while independently controlling virtual content in their FoV. *TeleAdvisor* [9] and *Shared Space* [4] AR systems allowed remote expert to work with user's traditional tools and overlay virtual objects in user's FoV through a computer interface.

Recent studies have also revealed that in addition to sharing the user's FoV with the remote expert, it is also necessary to provide the expert and worker with right interface and functionality to collaborate and work efficiently. To address this problem, Bauer et al. [2] conducted an empirical study aiming at expert using *telepointer* to highlight region in wearable video conference system. *GestureCam* [14] explains a system in which expert hand gestures on touch screen device are captured and sent to user's wearable. The results have shown that the experts preferred *telepointer* or gesture based interaction over verbal communication in assisting tasks. But, there are no existing formal studies on highlighting the ROI from a wearable.

In this paper, we address the problem of providing the user an interface to highlight region in his FoV to communicate efficiently via intuitive free-form gestures. Since user performs a task wearing Head Mounted Device(HMD), providing an additional hardware would amount to increased cost and complexity. Therefore, we propose a novel touch-less/air hand gestures in user FoV to highlight the ROI on frugal smartphone based head mounts such as Google Cardboard/Wearality. The recent work by Perla et al.[20] has discussed an industrial inspection framework where extension of *Google Cardboard*, which was initially envisioned for VR, was extended to AR. Further, [11] and [18] discussed simple hand gestures for wearables with egocentric view. This work motivated the idea of highlighting the ROI for frugal AR headsets.

Recognizing hand gestures from single monocular RGB data captured from FPV has been a challenging task in computer vision as smartphones are not equipped with depth sensors. Serra et al. [23] proposed random forest super pixel classification for hand segmentation in egocentric videos. In [12], effective skin pixel extraction using Gaussian mixture model has been proposed. Betancourt et al. [3] presents a four stage approach for hand gesture recognition which does hand-presence detection followed

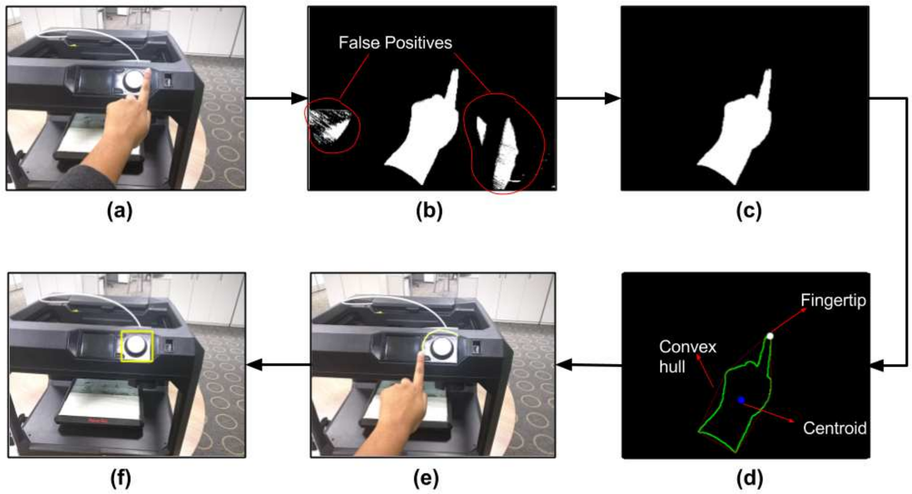

Figure 2. ROI Selection: (a) RGB Image frame acquired from a smartphone, (b) detection of skin-like pixels, (c) Largest boundary/contour segmentation for reduction of false positives, (d) *point gesture* detection using convexity defects, (e) Intermediate frame showing *point gesture* detection and ROI highlighting, (f) Bounding box overlaid on the object of interest.

by segmentation using a Bayesian approach. The trajectories of hand shape using centroids are further analysed for high level motion inferencing. Kalal et al. [13] proposed a novel object tracker called Median Flow Tracker that detects tracking failures with the help of Forward-Backward error method and selects consistent trajectories for tracking in video sequences. While these methods propose sophisticated detection and tracking; they are computationally heavy and difficult to port on a smartphone. In this work, we consider the factors such as real-time performance, accuracy, usability and latency of algorithm on smartphone. We also conducted a feasibility study evaluating ease and usefulness of the application in tele-assistance applications.

3. Proposed Method

We propose a novel marker-less and real-time two stage sequential gesture recognition method to highlight the ROI in the user's FOV. First, we detect a *dynamic gesture* which involves detecting the presence of a stable hand, followed by raising the index finger while rest of the fist closed (termed as *point gesture*, as shown in Figure 1(a)) to trigger *ROI Selection*. This is followed by another dynamic gesture involving moving *point gesture* around the object of interest. The main blocks of the algorithm as shown in Figure 1(b) are: (i) *point gesture* detection, (ii) ROI selection, (iii) ROI tracking, and (iv) subsequent updating of bounding box around the ROI. From Figure 2, the image frames obtained from the smartphone camera are first down-scaled to a resolution of 640×480, to reduce the processing time, without compromising much on image quality. Figure 2(b), (c), and (d) show the detection of skin pixels, largest contour segmentation which correspond to the hand (assumed as hand occupies prominent region while performing gesture from wearable), and *point gesture* detection respectively.

The *point gesture* recognition (Refer Section 3.1) will trigger ROI selection module 3.2 which performs fingertip detection on subsequent frames and draws an approximated bounding box around the object of interest following the locus of detected fingertips. The resultant bounding box is then tracked (3.3) as shown in Figure 3. Robust tracking of the marked ROI is an important challenge for a comfortable user experience at the remote site[3]. We utilize Shi-Tomasi feature points[24] for representing the marked ROI which is tracked in the subsequent frames using forward-backward(FB)[13] error method. The application scenario discussed in the paper will not have abrupt motion unless the user's object of interest is changed. FB error is an efficient method to deal with small motion and requires less computing resource. The blocks in Figure 1 (b) are discussed in the following sections:

3.1. Point Gesture Detection

Morerio et al. [19] observed that YC_bC_r color space shows better clustering of skin pixels data; the histogram of chroma channels (C_b and C_r) exhibit unimodal distribution while changing luminosity results in multimodal Y channel histogram. Reference [6] exploits the spatial characteristics of human skin color using chroma channel values. Thus, we use the Chroma channel information for skin pixel detection

[3]Tracking is incorporated for two reasons (a) as the relative distance between the object and user can vary and bounding box needs to be registered to object of interest (b) to address egocentric motion of user and object motion.

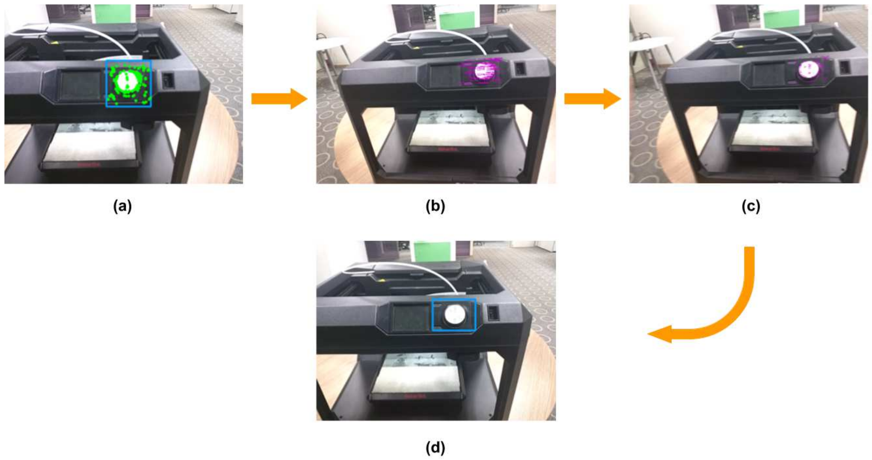

Figure 3. ROI Tracking: (a) Highlighted ROI and key Shi-Tomasi features[24] within the bounding box region, (b) Forward optical flow trajectories on the subsequent frames during the egocentric motion, (c) Backward optical flow trajectories on the previous frame[13], and (d) Updated bounding box approximation over the ROI.

making the hand detection process illumination invariant. Equation 1 describes the Chroma range used for segmenting the hand region from the background scene.

$$\begin{aligned} 77 &< C_b < 127 \\ 133 &< C_r < 173 \end{aligned} \tag{1}$$

Since the objective is for gesture recognition from FPV, it is safe to assume that the hand region would be the most prominent object present in user's FoV. We retain only the largest blob which covers a significant part of hand region by contour segmentation, using topological structural analysis of digitized binary images by border following algorithm, discussed in reference [26]. This step effectively removes all the skin-like background objects segmented in the previous step as shown in Figure 2(c). The binary mask from contour extraction is combined with the original image to produce the segmented hand region, which can be further used to recognize the *point gesture*. *Point Gesture* for ROI highlighting is initialized after the following conditions are satisfied:

- The hand region should occupy atleast 12% of the FOV which is empirically determined on the basis of the distance of the user hand from the wearable (Google Cardboard in our case). This helps in avoiding false detection of skin-like blobs.
- The steady hand is detected by observing centroid of the blob within certain radius for short duration. This is followed by user raising his index finger to highlight the ROI. The distance of the farthest point(fingertip) from the centroid is tracked and the gradual increase in this distance is verified to qualify foreground contour to be a *point gesture*.

3.2. ROI Selection

The *point gesture* recognition will trigger ROI selection which performs fingertip detection on subsequent frames and draws an approximated bounding box around the object of interest following the locus of detected finger tips. Fingertip detection is performed by computing convex hull of foreground hand contour (as shown in Figure 2 (c) and (d)) using the Sklansky's algorithm [25] and convexity defects (comprises of *start*, *end* and *defect points*). A *start* or *end point* which (i) is farthest from centroid of the convex hull, and (ii) lies above the centroid (avoids false positives), qualifies to be fingertip. The same conditions are verified for subsequent frames and fingertip locus is stored.

Since free-form drawing might look cluttered, we approximate it by superimposing the bounding box over the ROI. The procedure described in Section 3.1 is followed for subsequent frames to compute the fingertip location. The false positive (or outlier) fingertip detections which can distort ROI are eliminated by thresholding the distance between consecutive frames detections. The distance is empirically determined and set to 100 pixels. This distance is observed over subsequent frames and when it decreases gradually, ROI is assumed to be near to completion and an up-right approximated bounding box is fitted over fingertip locus.

3.3. ROI Tracking

Figure 3 shows the steps involved in the tracking of the ROI. The block Figure 3(a) shows the highlighted region and key Shi-Tomasi feature points [24] determined on it.

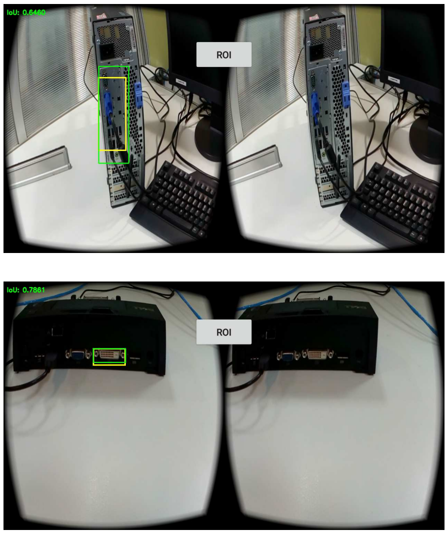

Figure 4. ROI highlighting through Google Cardboard: A yellow box indicating output from the proposed algorithm while the green box is the ground truth. We note the IoU between the two is $>$ 50%

These feature points are tracked every 3^{rd} frame to reduce the processing time using Lucas-Kanade optical flow [5] with pyramidal approach. Figure 3 (b) and (c) shows the optical flow trajectories of the feature points on the subsequent frames.

In order to improve the accuracy of tracking, we employ the Forward-Backward error method. In this method, the feature points are tracked twice, firstly from the previous image to the current image (forward in time) which yields a point set $\mathrm{P_f}$, and then from the current image to the previous image, using $\mathrm{P_f}$ set to yeild a point set $\mathrm{P_b}$. The points from the previous frame are used as validation points. The FB error is defined as the distance between these two tracking trajectories. We have used the Euclidean distance to determine the error, ($\mathrm{D}(\mathrm{P_f}, \mathrm{P_b})$ = $\|x_f(i) - x_b(i)\|$). The displacements that are within 5 pixels are considered as the reliable point trajectories and the corresponding forward tracked point is marked as inlier. Bounding box is then drawn onto the reliable point set thus obtained. This filters out the noisy feature points.

4. Results

We conducted an initial user evaluation to examine the use of ROI highlighting module in our framework, using the touchless gestures. We report measures both objectively and subjectively.

4.1. Objective Metrics

Quantitatively, we measure the ROI accuracy using PASCAL object detection criteria/ Intersection over Union (IOU)[7] to measure the bounding box overlap of ground truth image with our algorithm detected ground truth. We tried to highlight 30 different parts of 3D printer, Lathe Milling machine and desktop computer parts. We achieved IOU of minimum 50%, which is sufficient for an expert to understand the object of user's interest.

We carried out experiments to assess the application qualitatively and determine its usefulness. The participants in workplace (our research lab) consisted of 24 subjects of which 9 were females and 15 were males with average age of 26 years. The Lathe Milling workshop comprised of 24 participants, of which all were males with average age of 30 years. None of them had prior experience with respect to the usage of Google Cardboard(Nexus 6 and Nexus 5X mobiles are used for experiments) and touchless interface (hand gestures). All participants were asked to test the application 4 times each. Figure 4 shows some sample true detections of the object. In over 90% of the situations, the IoU was over 50% which we find adequate to spot the defect quickly via a telepresence set-up.

4.2. Subjective Metrics

A set of subjective metrics were obtained that measure both usability and user experience. These indicators measure human performance and user satisfaction. Users ratings were collected using a five-point Likert scale (Refer studies [10][16]) ranging from 1 to 5 (where 1 - Very Poor, 2 - Poor, 3 - Fair, 4 - Good, 5 - Very Good). The list below indicate 5 subjective metrics.

1 **Ease:** How easy was it to mark the ROI using air gestures?

2 **Responsiveness:** Did the application perform near real time while highlighting and tracking ROI? (We assume decent network connectivity to ensure server client communication happening smoothly)

3 **Accuracy:** How many times did the method detect the ROI correctly?

4 **Usefulness:** Rate the usefulness of the application and hands-free gestures in AR related tasks.

5 **Comfort:** Was there any strain in using the app/device?

Figure 5 shows that in ambient lighting environment, which was present in the workshop, the ratings for accuracy are relatively higher. This is because the algorithm performs fairly well in backgrounds which does not have much skin like colors. This would mean a much more robust skin model that works on a phone need to be developed. The subjective metric - *ease* and *usefulness* of the application were rated high as users found that it is more intuitive and comfortable to get assistance through tele-assistance setup using hand gestures than audio communication solely. Since the entire gesture recognition module is performed on-board, *responsiveness* of the gesture is quick thereby reduces the turn-around-time by 60% in completing the desired task efficiently. *Comfort* metric received lesser rating as Google Cardboard was found to induce simulation sickness in few users.

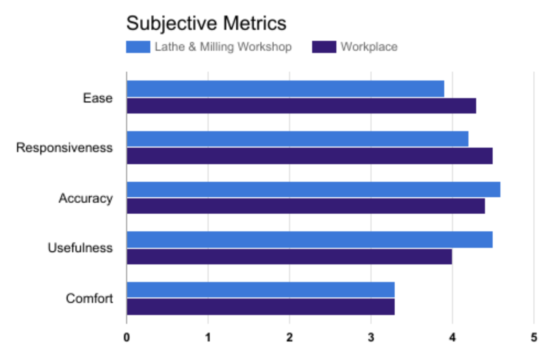

Figure 5. Mean Likert ratings of the subjective metrics, namely, (1) Ease, (2) Responsiveness, (3) Accuracy, (4) Usefulness and (5) Comfort.

4.3. Demo Video

The whole set-up of how the user performs the ROI highlighting is shown in the demo video at the URL: `https://arc4224.github.io/AirGestures-ROI/`

4.4. Hardware Setup

Motorola Nexus 6 and LG Nexus 5X, along with Google Cardboard were used to conduct experiments. Google Glass Explorer Edition 1 running Android 4.4 (Texas Instrument OMAP 4430 SoC, 1.2 GHz Dual ARM 7 processor, 1GB RAM, 5MP Camera) was used to test the application on Google Glass. Since Google Glass has limited computing power, the following backend server was used for computation of the ROI using hand gestures. The server-hardware configuration: Tesla C2075, CUDA Driver Version : 7.0, Computing Capability : 2.0, Total amount of Global Memory : 5375 Mbytes 14 Multiprocessors, 32 CUDA Cores per MP, Max no. of threads per MP : 1536, and Max no. of threads per block : 1024.

5. Discussion

A major problem in developing systems and interaction through speech and gesture is to determine the object(s) to which a user is referring, since irrelevant objects may likely fall into the user's gaze and pointing direction. To this end, we address the problem through introduction of ROI selection of the desired object using a natural user interface. We have developed an AR system in which the user, via free-form air gestures, can select an object onto which a bounding box is then superimposed. As a result, it is particularly suitable for applications with dense targets and rich visual elements.

Comparison with similar algorithms: Despite the availability of sophisticated algorithms for accurate hand detection [17], our method differs in methodology and uses simpler skin based model to achieve real-time performance. The primary drawback of the existing methods is (i) the necessity of a large training set that covers multiple lighting conditions, multiple skin tones and complex environments for the initial classifier, and (ii) computational time taken while testing. Several existing tele-presence frameworks [22, 1] uses touch based interaction on a tablet/smartphone for marking the ROI unlike our work which uses free-hand gestures through wearable for marking the same giving hands-free user experience while seeking assistance from the expert. We observed that Hui et. al [29] uses Kinect for depth detection as opposed to using just frugal devices such as smartphone and Google Cardboard for detection of hand.

Limitations of our work: We acknowledge the problems that may arise if the object of interest has relatively smooth surface leading to detection of less feature points; in such cases tracking might fail. Accuracy of the algorithm will suffer while performing gesture recognition in conditions such as (i) poor illumination, (ii) background scene blending with skin-like colors, and (iii) unexpected occlusions. Despite the entire proposed gesture recognition module has been implemented on smartphone for the use with Google Cardboard, smartphones need to be sufficiently powerful with atleast 2GB RAM and 1GHz processor to process enough number of frames for the ROI selection and tracking. Devices/wearables with low processing power such as Google Glass can't be used to run gesture module on-board; server side implementation had to be done to avoid battery drain and heating issues on Google Glass.

6. Conclusion

An Augmented Reality tele-presence framework for highlighting the region of interest from the wearable has been presented. We have demonstrated a touch-less gesture recognition algorithm on a smartphone with Google Cardboard in a dynamic background setting. This can enable the wider reach of frugal devices such as *Google Cardboard* for AR. Real-time performance is achieved by implementing gesture recognition module on-board. The evaluation of the framework has been done through set of subjective metrics and the accuracy of bounding box approximation over the object of interest using PASCAL VOC criteria.

In future, we would work on a much more robust skin detection algorithm as the current algorithm still mis-classifies skin-like background colors in some cases. We would also work on improving pointing gesture detection.

References

[1] L. Alem, F. Tecchia, and W. Huang. Remote tele-assistance system for maintenance operators in mines. 2011.

[2] M. Bauer, G. Kortuem, and Z. Segall. " where are you pointing at?" a study of remote collaboration in a wearable videoconference system. In *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on*, pages 151–158. IEEE, 1999.

[3] A. Betancourt, P. Morerio, L. Marcenaro, E. I. Barakova, M. Rauterberg, and C. S. Regazzoni. Towards a unified framework for hand-based methods in first person vision. In *ICME Workshops*, pages 1–6. IEEE Computer Society, 2015.

[4] M. Billinghurst, S. Weghorst, and T. Furness. Shared space: An augmented reality approach for computer supported collaborative work. *Virtual Reality*, 3(1):25–36, 1998.

[5] J.-Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001.

[6] D. Chai and K. N. Ngan. Face segmentation using skin-color map in videophone applications. *IEEE Transactions on circuits and systems for video technology*, 9(4):551–564, 1999.

[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[8] D. Gergle, R. E. Kraut, and S. R. Fussell. Action as language in a shared visual space. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 487–496. ACM, 2004.

[9] P. Gurevich, J. Lanir, B. Cohen, and R. Stone. Teleadvisor: a versatile augmented reality tool for remote assistance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 619–622. ACM, 2012.

[10] R. Hebbalaguppe, K. McGuinness, J. Kuklyte, G. Healy, N. O'Connor, and A. Smeaton. How interaction methods affect image segmentation: user experience in the task. In *User-Centered Computer Vision (UCCV), 2013 1st IEEE Workshop on*, pages 19–24. IEEE, 2013.

[11] S. Hegde, R. Perla, R. Hebbalaguppe, and E. Hassan. Gestar: Real time gesture interaction for ar with egocentric view. In *International Symposium on Mixed and Augmented Reality*. IEEE, 2016.

[12] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *Pattern recognition (ICPR), 2010 20th international conference on*, pages 2756–2759. IEEE, 2010.

[14] H. Kuzuoka, T. Kosuge, and M. Tanaka. Gesturecam: A video communication system for sympathetic remote collaboration. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 35–43. ACM, 1994.

[15] H. Liang, J. Yuan, D. Thalmann, and N. M. Thalmann. Ar in hand: Egocentric palm pose tracking and gesture recognition for augmented reality applications. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 743–744, New York, NY, USA, 2015. ACM.

[16] T. J. Maurer and H. R. Pierce. A comparison of likert scale and traditional measures of self-efficacy. *Journal of applied psychology*, 83(2):324, 1998.

[17] A. Mittal, A. Zisserman, and P. H. Torr. Hand detection using multiple proposals. In *BMVC*, pages 1–11. Citeseer, 2011.

[18] S. Mohatta, R. Perla, G. Gupta, E. Hassan, and R. Hebbalaguppe. Robust hand gestural interaction for smartphone based ar/vr applications.

[19] P. Morerio, L. Marcenaro, and C. S. Regazzoni. Hand detection in first person vision. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 1502–1507. IEEE, 2013.

[20] R. Perla, R. Hebbalaguppe, G. Gupta, G. Sharma, E. Hassan, M. Sharma, L. Vig, and G. Shroff. An ar inspection framework: Feasibility study with multiple ar devices. In *International Symposium on Mixed and Augmented Reality*. IEEE, 2016.

[21] A. Ranjan, J. P. Birnholtz, and R. Balakrishnan. Dynamic shared visual spaces: experimenting with automatic camera control in a remote repair task. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1177–1186. ACM, 2007.

[22] A. Roudaut, S. Huot, and E. Lecolinet. Taptap and magstick: improving one-handed target acquisition on small touchscreens. In *Proceedings of the working conference on Advanced visual interfaces*, pages 146–153. ACM, 2008.

[23] G. Serra, M. Camurri, L. Baraldi, M. Benedetti, and R. Cucchiara. Hand segmentation for gesture recognition in egovision. In *Proceedings of the 3rd ACM international workshop on Interactive multimedia on mobile & portable devices*, pages 31–36. ACM, 2013.

[24] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[25] J. Sklansky. Finding the convex hull of a simple polygon. *Pattern Recognition Letters*, 1(2):79–83, 1982.

[26] S. Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[27] Z. Szalavári, D. Schmalstieg, A. Fuhrmann, and M. Gervautz. studierstube: An environment for collaboration in augmented reality. *Virtual Reality*, 3(1):37–48, 1998.

[28] J. Trevelyan, W. R. Hamel, and S.-C. Kang. Robotics in hazardous applications. In *Springer handbook of robotics*, pages 1521–1548. Springer, 2016.

[29] H.-S. Yeo, B.-G. Lee, and H. Lim. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. *Multimedia Tools and Applications*, 74(8):2687–2715, 2015.