Robust Hand Gestural Interaction for Smartphone based AR/VR Applications

Shreyash Mohatta, Ramakrishna Perla, Gaurav Gupta, Ehtesham Hassan, and Ramya Hebbalaguppe
Smart Machines R&D Group, TCS Research, India
shreyashm12@gmail.com, {r.perla,g.gupta7,ehtesham.hassan,ramya.hebbalaguppe}@tcs.com

Abstract

The future of user interfaces will be dominated by hand gestures. In this paper, we explore an intuitive hand gesture based interaction for smartphones having a limited computational capability. To this end, we present an efficient algorithm for gesture recognition with First Person View (FPV), which focuses on recognizing a four swipe model (Left, Right, Up and Down) for smartphones through single monocular camera vision. This can be used with frugal AR/VR devices such as Google Cardboard[1] and Wearality[2] in building AR/VR based automation systems for large scale deployments, by providing a touch-less interface and real-time performance. We take into account multiple cues including palm color, hand contour segmentation, and motion tracking, which effectively deals with FPV constraints put forward by a wearable. We also provide comparisons of swipe detection with the existing methods under the same limitations. We demonstrate that our method outperforms both in terms of gesture recognition accuracy and computational time.

1. Introduction

With the evolution of touch screens, the keyboard and mouse based interface is gradually disappearing in smart digital devices. However, instinctive communication between human and machines is still a challenging task and in that direction, hand gesture based user interfaces is an active area of on-going research in computer vision [12]. Hand gestures form a popular means of interaction between computers, wearables, robots and humans which have been discussed in many recent works exemplified with the Augmented Reality(AR) devices such as *Microsoft Kinect*, *Hololens*, *Google Glass* [8, 7, 3]. AR involves overlaying computer simulated imagery on top of the real world, as seen through the optical lens of the device or a smartphone/tablet camera. The device side advances in smartphone technology have introduced several low-cost alternatives such as *Google Cardboard* and *Wearality* which are video-see-through devices for providing immersion experiences with a VR enabled smartphone [1]. 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.

The motivation to use these frugal headsets with a smartphone for AR applications was primarily due to its economically viable nature, portability and scalability to the mass market. Nevertheless, these frugal devices have limited user-input capability such as *magnetic trigger*, *conductive lever* or *head-tilt* interaction being used for *Google Cardboard* based applications. The recent work by Perla et al.[11] has discussed an inspection framework in an industrial use-case with multiple AR devices. Here, the extension of *Google Cardboard*, which was initially envisioned for VR, was extended to AR. Their framework uses *Google* speech engine for speech based user interaction, which requires constant network connectivity for communicating with the servers. But, in an industrial or an outdoor setting, the speech recognition is found to be inaccurate due to ambient noise.

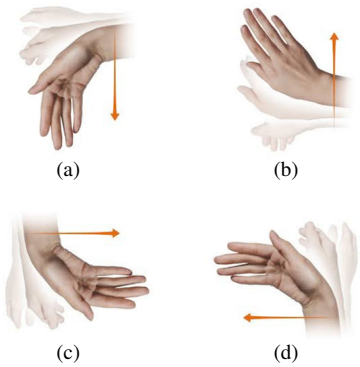


Figure 1. Simple hand gesture support for *Google Cardboard*: It involves a simple (a) down-swipe, (b) up-swipe, (c) right swipe, and (d) left swipe motion. This support enables user friendly interaction for AR based FPV applications with egocentric motion.

Hedge et al. [6] discussed gestural support which provides intuitive ways of interaction based on different number of hand swipe gestures included in the lexicon beyond simple event based interaction provided by magnetic trigger and conductive lever. It also conducted a feasibility study

[1] https://vr.google.com/cardboard/
[2] http://www.wearality.com/

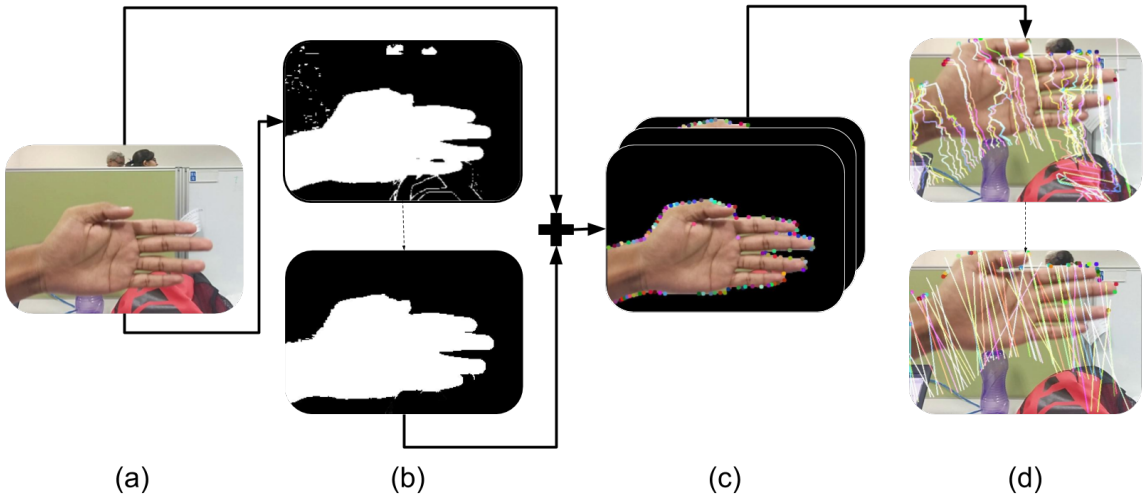


Figure 2. Block diagram of the proposed hand swipe gesture recognition algorithm: (a) Shows the palm frame grabbed during the hand swipe from a wearable device, (b) Hand detection module, that utilizes the Chroma channel information followed by the largest contour extraction to eliminate the noisy blobs, (c) Keypoint extraction (Shi-Tomasi descriptors) on the hand segment, (d) Depicts the overlaid optical flow trajectories on the sequence of frames

through both subjective and objective metrics against the other modes of available interactions such as conductive lever, head tracking, and, the speech interface. Hand gestures based interaction is found to be the most user-friendly interaction achieved through reduced effort and any physical strain. In this paper, we present a robust hand gesture recognition technique, which can detect and recognize 4 types of hand swipes (such as Up, Down, Left and Right, as shown in Figure 1) by on-board processing of camera feed in real-time, with a low computational load. We circumvent the shortcomings of the probabilistic models for hand detection by using a simpler palm detection model that works purely on C_b and C_r values using a statistical model. The approach is computationally efficient and can be run on an ordinary smartphone. Hand swipe based interaction has several applications such as traversing a list on a wearable or flip pages on a wearable/tablet. The summary of contributions are as follows:

1 A novel hand swipe-gesture based user interaction method 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.

2 Our gesture recognition algorithm is explored for (i) their suitability to work with RGB channel stream or the 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.

3 We have the also created a Hand-swipe dataset that comprises of 4 different swipes, and published the same online [3] for the benefit of the community. Subsequently, demonstrated the performance of this approach on 100 sample hand swipe videos and the results are compared with Hedge et al. [6] and Nanogest gestures SDK[4].

2. Related Work

Recognition of hand gestures using wearables is a challenging vision problem because of the non-static reference of the camera where (a) the hand may be visible in part or full, (b) the swipes tend to be close to head mounted AR device and may often be outside the depth of field of the camera making it blurry, and (c) the hand portions may fall under different illuminations, the background might be static or noisy. Betancourt et al. [3] present a four stage approach for hand gesture recognition which does hand-presence detection followed by segmentation using a Bayesian approach. The trajectories of hand shape using centroids are further analysed for high level motion inferencing. Liang et al. [7] proposed tracking and recognition of hand poses in six DOF space for manipulating virtual objects using Random forest based regressor and Kalman filter based tracker.

In a similar approach by Song et al. [14], the random forest based regressor models the RGB data capturing the hand motion for 3D gesture recognition. In [2], RANSAC based feature point sampling is applied for addressing the ego-motion because of camera movements. The trajectories

[3]Website: *https://sam16222.github.io/GestureRecognition/*
[4]http://www.nanocritical.com/nanogest/

of these feature points are subsequently described by appearance based description for hand gesture recognition. While these methods propose sophisticated detection and tracking schemes for hand gesture analysis; these algorithms are computationally heavy and are difficult to port on a smartphone with ordinary configuration (1GB RAM, 1 GHz processor).

Hedge et al. [6] throw light on simple gestures using GMM based hand modeling for wearable with egocentric view. While GMM [13] is a highly viable option for hand detection, tracking and subsequent classification, the computational load of having a GMM based swipe recognition model on phone leads to compatibility issues with most mobile phones and thus hinders real-time and accurate classification. They also take into account only two swipes for hand gesture classification, while our work extends it two 4 types of gestures (up, down, left, and right swipes). In this work, we take into account multiple cues including palm color, hand contour segmentation, and motion tracking which effectively deals with FPV constraints caused due to wearable device. We also provide comparisons of swipe detection with the existing methods which lack accuracy and real-time performance. We demonstrate that our method outperforms the gesture recognition accuracy and computational time.

3. Proposed Method

In this section, we propose an approach for real-time hand swipe detection from FPV in varied background settings. We focus on hand swipe recognition using only RGB image data without any depth information as current smartphones do not have in-built depth sensors. Figure 2 shows various blocks in the hand gesture recognition from a wearable device. The blocks are (a) shows the image acquisition step - images are down-scaled to 640 x 480 resolution to reduce the processing time without compromising much on image quality, (b) hand detection module consisting of skin detection and segmenting out the largest contoured object which corresponds to palm/hand , (c) the resultant image after foreground hand extraction and key Shi-Tomasi features detection on the foreground to track the hand motion, and (d) tracking module consisting of optical flow trajectories of key feature points on the upcoming sequence of frames and resulting displacement vectors for swipe classification.

3.1. Hand Detection

Morerio et al. [9] 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. The luminance property merely characterizes the brightness of a particular chrominance value [10]. Thus, we used the chroma channel information for skin pixel detection. This makes the hand detection process illumination invariant. Chai and Ngan [5] have developed an algorithm that exploits the spatial characteristics of human skin color using chroma channel values. Equation 1 describes the threshold values of the filters used for segmenting the hand region from the background scene.

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

The filtering of skin pixel data by this approach might also generate noisy blobs for some cases, when there exists skin-like colors in the background. This makes the hand motion tracking difficult. To avoid this problem, 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 [15]. 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 the user field of view.

This step effectively removes all the skin-like background objects segmented in the previous step as shown in Figure 2(b). The binary mask from contour extraction is combined with the original image to produce the segmented hand region which can be further used for key point detection and tracking to recognize the gesture.

3.2. Hand Tracking

The obtained foreground hand region is then used to obtain Shi-Tomasi [13] feature points, employed with Lukas-Kanade optical flow [4] with pyramidal approach for tracking. Tracking will be initialized for the detected key feature points after two following conditions are satisfied:

(i) Once hand region occupies the certain area in the user field of view (FoV). The area threshold can be heuristically determined based on the device being used and typical distance of user hand from the wearable (26% of area has been chosen as the threshold for *Google Cardboard* based application discussed in the results section). This helps in avoiding tracking of false alarm blobs.

(ii) The minimum number of feature points attained on the foreground hand region (40 points used as threshold in experiments). This helps in achieving better accuracy, by tracking a good number of points for the cases of blurry vision from smartphone camera when the hand is held very close. Feature points will be re-spawned on the subsequent frame as the number of points on blurry image will be very less and also swipe gestures typically involves user palm region in the FoV which is smooth thereby reducing the number of points.

After initialization, these points are tracked frame by frame and the resultant displacement vectors are calculated.

3.3. Egocentric Correction

Another important concern while considering FPV applications is egocentric correction. Small irregularities due to the head motion can give rise to anomalous results. By using final displacement vectors to compute the overall flow of the feature points, we eliminate small irregularities (as shown in Figure 3). Skin-like objects that are part of the largest contour won't show any effect on the performance as their features flow is similar to the global motion, and our algorithm takes care of features that are tracked in local neighborhood as explained in Section 3.4. Since there isn't any significant movement for these points, their respective magnitudes are considerably small and thus their impact is significantly reduced when normalized as given in Equation 2.

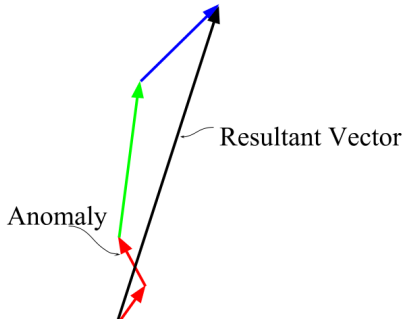


Figure 3. Flow vectors representing feature point movement across subsequent frames and resultant displacement vector (black color) correcting errors caused due to user movement in FPV applications.

3.4. Hand Swipe Classification

We propose a more robust model for classification unlike Hegde et al. [6]. The slope based classification based on consecutive frames can be quite erroneous, due to the similarity and proximity of detected feature points, which can undergo significant jumps across frames. So, casting swipe direction decisions based on consecutive frames is avoided for better accuracy. Our classification algorithm works on the idea of flow vectors displacement computed over the entire duration of gesture. The accuracy has been improved in situations as described below:

(i) For the cases when we lose the initialized feature points correspondences by a significant amount in subsequent frames, we calculate direction of swipe θ_i for that particular interval i (as shown in Figure 4)using the following equation:

$$\theta_i = \frac{1}{M}\sum_{k=1}^{N} m_k \theta_k \tag{2}$$

where

$$M = \sum_{k=1}^{N} m_k \tag{3}$$

Where m_k and θ_k are the magnitude and direction of displacement vector for each feature point respectively. N is the number of feature points retained over interval i.

Since a typical swipe gesture involves user hand moving from one end of the frame to the other end, feature vectors are expected to be of higher magnitude. For this reason, feature vectors are given weights in proportion to their magnitudes, as shown in Equation 2, thereby reducing weights to the false positive features tracked in local neighborhood. This process can be repeated by re-spawning the feature points and tracking for next interval $i+1$ till the foreground hand region goes out of FoV. This step helps in improving robustness of algorithm by avoiding false alarm recognition for the cases when camera goes out of focus for very short duration of gesture.

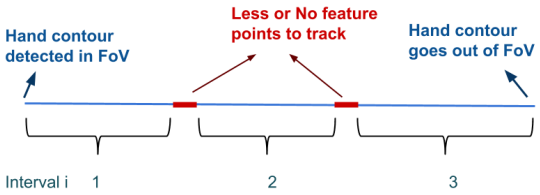


Figure 4. Time intervals(i) are shown for which feature points are obtained on the foreground hand region and marked red regions representing the feature points less than threshold of 40% due to camera defocus or smooth foreground hand region.

(ii) Resultant direction θ_f of user swipe can be obtained from θ_i values calculated for all the intervals over the entire duration of hand gesture using the following equation.

$$\theta_f = \frac{1}{T}\sum_{i=1}^{n} t_i \theta_i \tag{4}$$

where t_i is time taken for interval i, T is the sum of all t_i's, and n is the total number of intervals. In equation 4, feature points sustained for longer duration are given higher priority thus improving the accuracy. This resultant direction θ_f is used for recognizing the swipe gesture with a tolerance band of $\pm 35^\circ$ (example, $-35^\circ \leq \theta_f \leq 35^\circ$ for Right swipe).

Predicted Gesture / True Gesture	Up	Down	Left	Right	Unclassified
Up	21	0	1	1	2
Down	0	24	0	0	1
Left	1	0	22	0	2
Right	1	1	0	21	2

Table 1. Confusion matrix of our proposed algorithm yeilding an accuracy of 88%

4. Results

We perform hand swipe detection experiments as an application on a frugal VR/AR device *Google Cardboard*. Hence, an application has been developed to work on smartphones running the Android OS. The stereo camera view on the smartphone helps the application to be used with *Google Cardboard*. The detected gesture result is overlaid after single swipe as shown in Figure 5.

4.1. Dataset and Experimental Set-up

The hand swipe dataset used to test gesture recognition is composed of 100 sample videos recorded using Android smartphones, when used from *Google Cardboard*. There were 18 subjects involved in the experiments, with ages spanning from 21 to 53. The ground truth information regarding the direction of swipe was labelled manually. The videos were recorded in different places (living room, indoor office setting, cafetaria, among others) in order to gather variations in lighting conditions, color compositions and dynamic background scene. The videos were recorded with Nexus 6 and Moto G3 smartphones with 1280 x 720 px resolution and 30 fps. Our dataset comprises of 4 different swipes(Left, Right, Up, Down), and we have published the same online [5] for the benefit of the community.

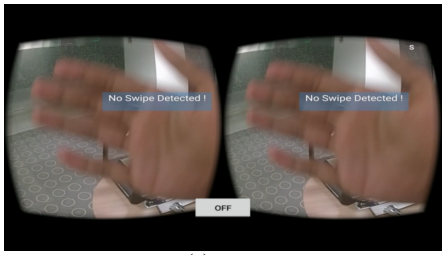
The Google Cardboard VR SDK for Android is used for developing smartphone application for Gesture recognition from wearable *Google Cardboard*. OpenCV SDK for Android has been used for implementing the proposed approach to run on Android smartphone devices.

4.2. Evaluation and Discussion

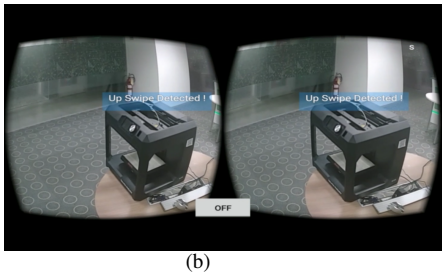
In order to compare the performance of proposed approach, we evaluated our algorithm on the hand swipe dataset of 100 sample videos and compared the same with Hegde et al. [6]. Nanogest SDK[6] which is commercially available for swipe gesture classification on iOS and Android platforms is also evaluated for its accuracy. The obtained results were expressed using multi-class confusion matrices, a statistical tool used to analyze the efficiency of a classification model.

[5]Website: *https://sam16222.github.io/GestureRecognition/*

[6]http://www.nanocritical.com/nanogest/



(a)



(b)

Figure 5. Hand gesture recognition from a *Google Cardboard* based AR application for smartphones. (a) User performing a hand-swipe gesture wearing the *Google Cardboard* head-mount. (b) Recognition result "Up Swipe" being displayed on the smartphone screen.

The rows are the actual gestures, with the columns containing the predicted results. Higher the diagonal values, better the accuracy of the classification task. The confusion matrix is used to analyze each class separately using precision an recall values.

The experimental results of our proposed approach are summarized in Table 1, and show that our algorithm yields 88 correct classifications with 12 mis-classifications. The mis-classifications were analyzed and root cause for the same was determined: 3 misclassifications were observed due to the ambiguity of the direction of the swipe. The other 9 mis-classifications resulted due to a high proximity of the hand to the camera, making the obtained feed blurry beyond a comprehensive threshold. The confusion matrices of the other two algorithms, Nanogest SDK[7] (shown in Table 2) and Hegde et al. (shown in Table 3) respectively show an accuracies of 86% and 73% under similar conditions. The obtained results show that our algorithm produces superior results with negligible latency. The turn around time for Hegde et al. was 0.885s while the average response time for the proposed approach was found to be 0.19s.

5. Conclusion

The paper presents a novel approach for hand swipe recognition from FPV, using a wearable smartphone based AR device. Hand gestures are an intuitive approach for user interaction modes in AR/VR based applications. Our approach

[7]tested using Wave-o-rama application for iOS

Predicted Gesture / True Gesture	Up	Down	Left	Right	Unclassified
Up	21	0	0	0	4
Down	0	22	0	0	3
Left	0	0	21	1	3
Right	0	0	2	22	1

Table 2. Confusion matrix using Nanogest SDK, yeilding an accuracy of 86%

Predicted Gesture / True Gesture	Up	Down	Unclassified
Up	33	2	15
Down	3	40	7

Table 3. Confusion matrix using Hegde et al., yeilding an accuracy of 73%

presents a simple algorithm which works with the RGB channel stream from a smartphone monocular camera without any built-in depth sensors. We demonstrate that the proposed approach can yield robust swipe detection and classification, without requiring any complex modeling procedure. This can be easily ported on any device with ordinary hardware configuration and can demonstrate real-time performance in dynamic background settings. We have validated the approach on a group on diverse subjects belonging to different race, age groups. However, the evaluation of proposed approach on a larger collection of hand videos and a bigger lexicon is part of future work.

References

[1] A. Amer and P. Peralez. Affordable altered perspectives: Making augmented and virtual reality technology accessible. In *GHTC*, pages 603–608. IEEE, 2014.

[2] L. Baraldi, F. Paci, G. Serra, L. Benini, and R. Cucchiara. Gesture recognition in ego-centric videos using dense trajectories and hand segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–707, June 2014.

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

[5] 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.

[6] 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.

[7] 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.

[8] Z. Lv, L. Feng, H. Li, and S. Feng. Hand-free motion interaction on google glass. In *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications*, SA '14, pages 21:1–21:1, New York, NY, USA, 2014. ACM.

[9] 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.

[10] J. S. N. A. Abdul Rahim, C. W. Kit. Rgb-h-cbcr skin colour model for human face detection. In *MMU International Symposium on Information and Communications Technologies (M2USIC)*, Petaling Jaya, Malaysia, 2006.

[11] 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.

[12] S. S. Rautaray and A. Agrawal. Vision based hand gesture recognition for human computer interaction: A survey. *Artif. Intell. Rev.*, 43(1):1–54, Jan. 2015.

[13] 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.

[14] J. Song, G. Sörös, F. Pece, and O. Hilliges. Real-time hand gesture recognition on unmodified wearable devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[15] 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.