Journal of Computer and Communications, 2016, 4, 99-108
Published Online April 2016 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2016.44009

A Robust Alternative Virtual Key Input Scheme for Virtual Keyboard Systems

Prince Owusu-Agyeman*, Wei Xie, Yao Yeboah

Department of Electrical and Computer Engineering, School of Automation Science and Engineering, South China University of Technology, Guangzhou, China
Email: *princeagyeman@hotmail.com

Received 3 March 2016; accepted 15 April 2016; published 20 April 2016

Abstract

Towards virtual keyboard design and realization, the work in this paper presents a robust key input method for deployment in virtual keyboard systems. The proposed scheme harnesses the information contained within shadows towards robustifying virtual key input. This scheme allows for input efficiency to be guaranteed in situations of relatively lower illumination, a core challenge associated with virtual keyboards. Contributions of the paper are two-fold. Firstly the paper presents an approach towards effectively applying shadow information towards robustifying virtual key input systems; Secondly, through morphological operations, the performance of this input method is boosted by means of effectively alleviating noise and its impacts on overall algorithm performance, while highlighting the necessary features towards an efficient performance. While previous contributions have followed a similar trend, the contribution of this paper stresses on the intensification and improvement of both shadow and finger-tip feature highlighting schemes towards overall performance improvement. Experimental results presented in the paper demonstrate the efficiency and robustness of the approach. The attained results suggest that the scheme is capable of attaining high performances in terms of accuracy while being capable of addressing false touch situations.

Keywords

Virtual Input, Shadow-Based Input, Input Robustication, Virtual Key Input

1. Introduction

Human-computer Interaction and its recent developments in computer technology in modern years have grown

*Corresponding author.

How to cite this paper: Owusu-Agyeman, P., Xie, W. and Yeboah, Y. (2016) A Robust Alternative Virtual Key Input Scheme for Virtual Keyboard Systems. *Journal of Computer and Communications*, **4**, 99-108.
http://dx.doi.org/10.4236/jcc.2016.44009

rapidly. Additionally, the inventions and accompanying research efforts aimed at portable input devices are becoming an increasingly crucial factor in development of electronic devices such as laptops, smart phones, tablet PCs and others. While traditional keyboard technology continues to remain the most predominant mode of interaction between man and machine, this technology has also been associated with a limited reliability, robustness and flexibility which tends to limit its adoption in seamless and natural interaction between man and machine [1]. The need for alternative methods of interaction that are capable of overcoming the flaws associated with the traditional keyboard could therefore not be exaggerated. Virtual or otherwise referred to as projection keyboards are one of the fastest growing solutions designed to draw from the strengths while remedying the short-comings of traditional key input systems. With an application potential that stretches across various industries including but not limited to personal electronics, industrial automation, medicine and finance, these alternative means of interaction with machines have received significant research attention and this has in turn contributed to their rapid evolution and growth over the past decade. The growing progress and research effort in the field of image processing has been a key contributing factor in the rapid growth of virtual keyboard technology. Virtual keyboards are in line with the current technological trends that do not only require flexibility but scalability and efficient implementation. These keyboards have the capability of alleviating the space constraints associated with the design of hardware systems. From a general point of view, conventional virtual keyboards are essentially a combination of the following active components: imaging equipment, keyboard templates, processing component and data links.

The research community has contributed a vast amount of effort towards virtual keyboard research and implementation and this has in turn yielded significant results that are covered below. Firstly, a thorough and in-depth literature review is covered in [2] in which the authors focus on bringing to light the challenges and milestones that have been encountered in the development of virtual keyboard technology for cross platform hardware and industry. The paper presents a survey which covers existing strategies and critically examines their applicability to the design and realization of virtual keyboards. Generally speaking, a majority of related work has aimed at specific components of the virtual keyboard technology as a whole. The works seen in [3], [13] and [18] have addressed the camera component of virtual keyboards through an implementation approach. It is undeniable that the camera component or otherwise referred to as the acquisition component or imaging component is a core part of virtual keyboard design and merits intensive research focus. The research works that have addressed the camera design component establish certain design rules that are crucial towards the evaluation of the performances and efficiencies of virtual keyboard systems. Implementation is further addressed in the work seen in [4] to [9]. The works seen in these literatures on the other hand, address the keyboard design in from various aspects. A significant number of the research effort applied in this category of literature is aimed at establishing certain optimum keyboard dimensions which are key towards boosting input efficiency and speed. Additionally, machine learning has also been applied [4] towards equipping the virtual keyboards with the capability to adjust the key size and inter-key distances on-the-fly in line with the perceived user input behavior. With user experience core to the virtual keyboard technology, further effort has also been invested into this sector with groundbreaking work seen in [5], [7] and [8]. Drawing from the high prevalence of virtual keyboard in the financial and information-sensitive sectors, security in implementation has been addressed in [6] and [16]. An anti-screen shot virtual keyboard is implemented in [6] with the capability of circumventing attacks on user data during critical routines such as password entry. [12] and [15] address performance models which are crucial towards ensuring overall performance of virtual keyboards while feature selection and matching is tackled in [10]. It is necessary to reiterate that flexibility and efficiency remain the core concepts that have motivated the wide-spread interest in virtual keyboard technology and this is evident in how the research community has attempted to expand flexibility and incorporate other more natural modes of entry including but not limited the Brain Computer Interfaces (BCIs) [11], Electroencephalography (EEG) [14] and Gaze [17].

The work in this paper presents and addresses robust key input in a virtual keyboard environment using shadow information as mode of entry. The proposed scheme is capable of harnessing variations in illumination towards achieving robust key input. Based on implementation with image processing techniques, shadow features are extracted towards reinforcing virtual key input. Through morphological image processing, efficiency of input is achieved in an online key input system. The achieved performance proves capable of identifying and distinguishing between false touch, near touch and true touch situations. Experimental results achieved with the scheme demonstrate the efficiency of the scheme while highlighting the applicability of the scheme in online systems. The effort in the paper is manifold. Firstly the paper presents a scheme towards effectively applying shadow information towards robustifying virtual key input systems; Secondly, through morphological operations, the performance of this input method is boosted by means of effectively alleviating noise and its effects, while highlighting the necessary features towards an efficient performance. Experimental results are presented in support of the performance of the scheme with some overview of how hardware systems can be implemented towards realizing the scheme presented in the paper. The work in this paper is relevant to ongoing research in the field as it provides a mean by which virtual keyboard systems can be robustified against common challenges encountered in implementation. Through implementation with shadow features in the scheme presented in this paper, current systems are provided with a means of robustification without the need for any excessive additional hardware costs or system modifications. The rest of the paper is organized as follows: Section 1 presents a background and review of existing state-of-the-art. Section 2 presents the proposed algorithm with experimental results presented in Section 3. The paper concludes in Section 4.

2. Proposed Algorithm

This section presents comprehensively the theoretical and implementation details of the proposed robust virtual key input scheme. Our proposed scheme is partially motivated by the work in [8] and is generally represented in **Figure 1**.

The algorithm operates simultaneously on the current input image while retaining some level of memory via a reference image storage and retrieval mechanism. Within the processing pipeline as well, processing of the images is subdivided into two mine categories.

a) Finger Processing

b) Shadow Processing

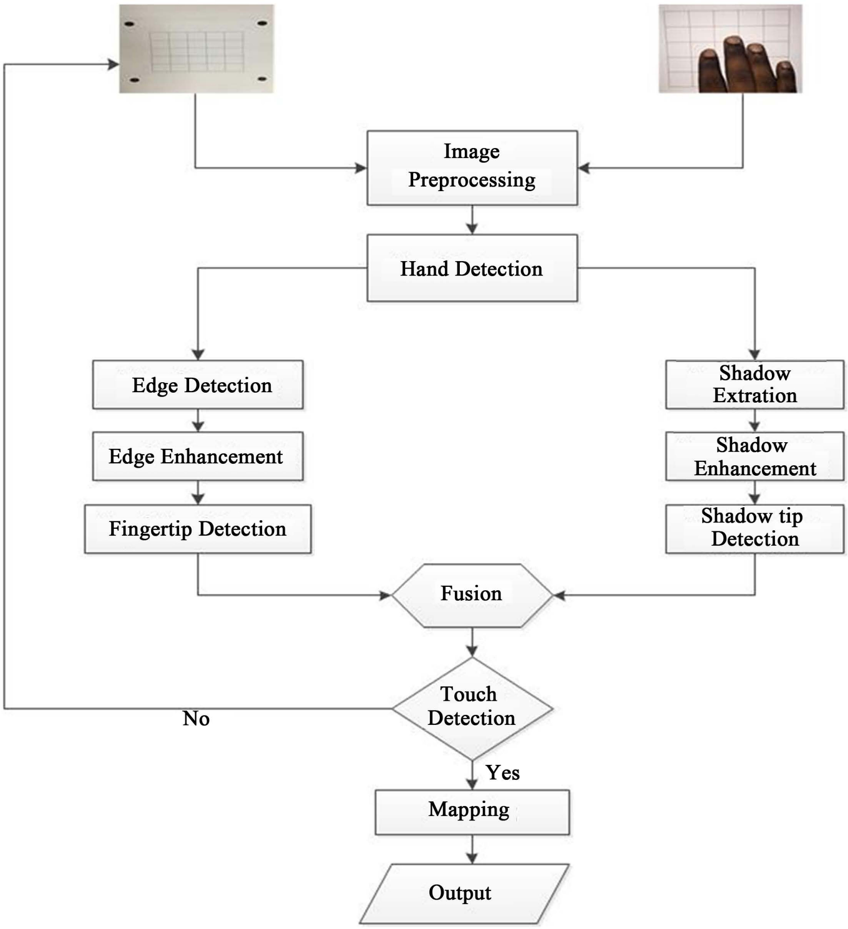


Figure 1. Processing Flow of the proposed robust virtual key input.

In operation, when an image is captured by the image acquisition device (camera), it is expected that some level of noise or interference from the environment will be present within the image and the first crucial step in the processing pipeline is efficient preprocessing of the input image aimed at reducing the signal to noise ratio (SNR) of the input image and hence removing the adverse effects of noise. This is important in enhancing the performance of the analysis components of the algorithm and achieving stable shadow extraction.

2.1. Detection-Driven Image Preprocessing

At the initial stages of the algorithm, image preprocessing is required in enhancing the information within the target image. As already well established from image processing, the Gaussian filter or otherwise referred to as the blur filter has the capability of removing noise occurring at specific frequency regions of the image and allows for deblurring of the target image and alleviating the effects of salt of pepper noise. This filter is applied to the raw input image to achieve a blurring effect which removes the noise present within the image at the moment of capture. Once this initial preprocessing has been achieved, we then proceed to perform a hand detection on the resulting image. As opposed to other literature [19] that performs this hand detection operation in the HSV colour space, we propose to perform this task within the RGB space for the purpose of preserving pixel information and cutting down computation load and time. This is more in line with the online requirements of the system. Towards hand detection, based on the intuition that most of the pixels belonging to the hand are within the red region of the colour space, a thresholding scheme is setup towards a systematic control and filtering of pixels based on their RGB composition. The pixels with red intensities high enough to meet the threshold are allowed to pass while blocking out all other pixels. Furthermore, the pixels that meet the threshold requirements are then represented with white values while all pixels cut-off are represented in black. This achieves a binarization effect and the corresponding output image only possesses two categories of pixels where white represents hand regions and black regions is the background.

Due to the manner in which the hand detection routines operate, it is to be expected that some regions of the background could also contain pixels red enough to pass through the thresholding scheme and hence the resulting image after hand detection could further contain sparse noise which needs to be dealt with in order to guarantee algorithm stability. In tackling this sparse noise in the resulting image, it is crucial to take into consideration that the image at this point contains very sensitive information (hand pixels) and therefore the filtering scheme applied at this stage is required to preserve as much information as possible. For this reason, due to the inability of the Gaussian filter to preserve edge information, the median filter is relied upon in further preprocessing the hand image. The adopted median filter is selected intuitively to possess a 5 × 5 neighborhood mask as this has shown to maintain a proper balance between computational load and filtering efficiency. The structure of the filter is illustrated in the Equation (1) below:

$$M_{5\times 5} = \left\{ \sum_{n=i-2}^{i+2} \sum_{k=j-2}^{j+2} \left(x_n, y_k \right) \right\} \tag{1}$$

According to the Equation (1) above, the pixels located within the bounding region of a 5 × 5 window are selected, sorted into an array, the median computed and the result applied in a substitution operation to replace the target pixel within $\left(x_n, y_k \right)$.

This can further be represented in terms of the structuring element as following:

$$\left(x_i, y_j \right) = \begin{cases} 255, M_{5\times 5} \geq 255 \times \left(\dfrac{N}{2} \right) \\ 0, M_{5\times 5} < 255 \times \left(\dfrac{N}{2} \right) \end{cases} \tag{2}$$

The Equation (2) presents a modification of the Equation (1) by featuring a structuring element, the dimensions of which are represented by *N*. $\left(x_i, y_j \right)$ represents the resultant image derived after this median operation is applied to the target image. Once the median-based preprocessing is attained, the image features are further enhanced by means of morphological operations. While various morphological operations are relied upon in previous work, the closing operation which has the capability of connecting disjoint regions within an image is adopted towards enhancing the hand information present within the target image at this point. Generally speaking, the closing operation is a sequential combination of dilation and erosion operations and can be mathematically represented by the Equation (3) below. In the Equation (3) where *I* represents the input or target image and *K* represents the structuring element, the closing operation can be illustrated as follows.

$$I \cdot K = (I \oplus K) \ominus K \tag{3}$$

where $\oplus$ represents dilation and $\ominus$, the erosion operation. By taking the input *I* to be of the form, (x, y) and the structuring, *K*, of the form, $\acute{x}, \acute{y}$, the constituent erosion and dilation operations can be generalized as follows.

$$\text{erosion} = \min_{(\acute{x},\acute{y})} \text{target}\left(x + \acute{x}, y + \acute{y}\right) \tag{4}$$

And

$$\text{dilation} = \max_{(\acute{x},\acute{y})} \text{target}\left(x + \acute{x}, y + \acute{y}\right) \tag{5}$$

The interested reader is referred to [20] for further reading on morphological image processing. While this operation succeeds in enhancing the information within the image, some minute noise features still remain but since their effects are not very significant at this point, they are tackled later on after the edge detection operations have been completed (see **Figure 2**).

2.2. Edge Detection

It is arguable that the edge detection stage of the algorithm is one of the most crucial stages in the proposed algorithm. The importance of this stage is based on the fact that the edge detection is crucial in extracting the contour information that represents the finger and the shadow. In achieving this edge detection we rely on the Sobel edge operator, a choice that is motivated by intuitive experiments with the Sobel, Canny and Prewitt Operators. Although the Sobel edge operator in itself is capable of extracting the edges of the hand in the form of the contour, the output at this stage could be broken and non-continuous in manner due to a number of reasons including but not limited to uneven illumination and the presence of noise. This therefore requires certain edge enhancement techniques in order to highlight the contour information that has been obtained at this point.

In enhancing the edge information obtained at this point in the processing pipeline, contour analysis is relied upon. The contour analysis traverses the extracted contour and operates on the individual pixels in a manner which tracks the contour in a specific direction. The result of this operation is an enhancement of the contour making up the finger and unifying all closely lying pixels into a single blob. The result attained at this stage is illustrated in **Figure 3**.

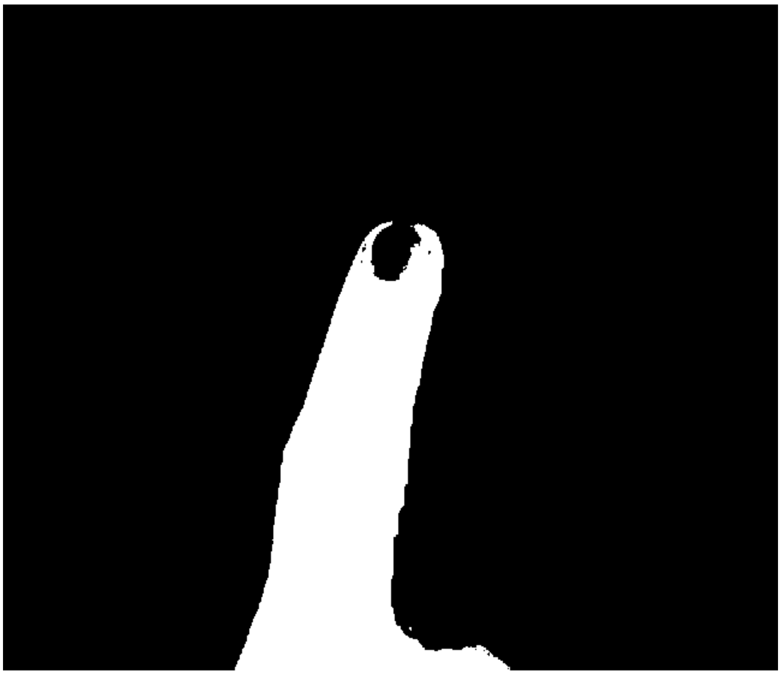
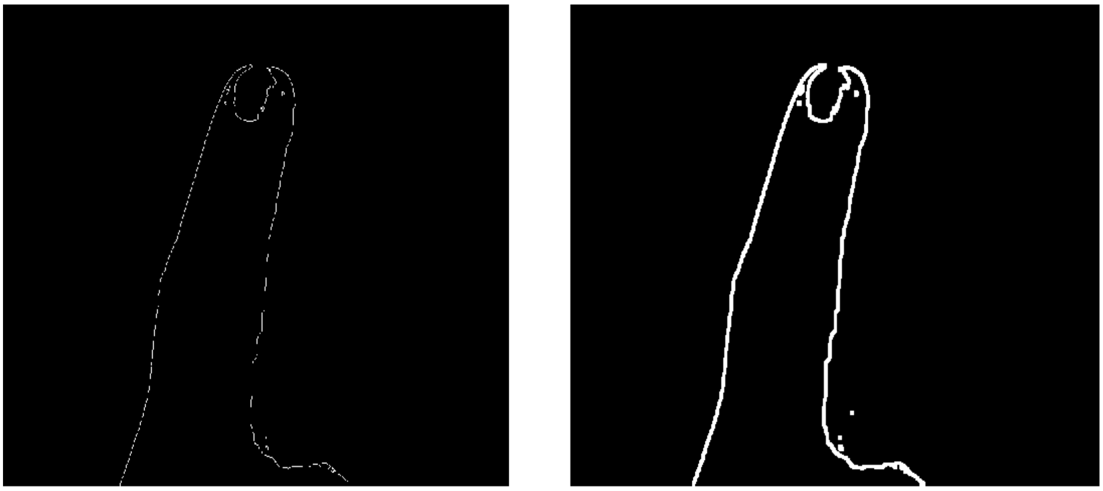


Figure 2. Results of hand extraction and morphological operations.



(a)

(b)

Figure 3. The results attained with (a) Edge Detection and (b) Edge Enhancement Schemes.

2.3. Shadow Extraction

Perhaps the most straight-forward execution stage in the proposed scheme is the shadow extraction. This straight-forwardness is facilitated by implementation with the reference image attained at the initial staged of processing. A basic subtraction operation is applied in separating the target image from the reference image, thereby leaving behind only the shadow information. The shadow information attained at this stage is retained and the scheme transitions into the fingertip detection stage. The algorithm applied in detecting the finger tip is the same as that which is applied in detecting the extracted shadow tip. For simplicity of representation, this algorithm is explained once at the fingertip detection stage. **Figure 4** presents the results of this shadow extraction scheme.

2.4. Finger Tip Detection

The fingertips need to be detected and processed in order to achieve key input and this is realized in this section in a manner in line with the work presented in [8]. Since we have the contour information of the hands at our disposal at this point, a discretization function is applied towards efficiently converting the contour information into coordinate data allowing for discrete processing to be achieved. The processing is achieved by mapping the triangular coordinate system onto the available contour information, and then by assuming that in this three coordinate system, the middle coordinate represents the highest point of the triangle, the angle of the triangle could then be computed as follows:

$$\delta = \mathrm{arcos}\left(\frac{x^2+y^2+z^2}{2xy}\right) \tag{6}$$

where, x, y, z, represent the edges of the triangle which can be represented as. This operation is used in traversing along the edge of the contour in an iterative manner and storing all output values. Once the operation is completed, the output with the smallest value is considered as the finger tip based on the assumption that the finger tip is the only area within the contour which produces the smallest output angle. The shadow extracted in the previous section is also processed for its tip in a similar manner. The results attained in fingertip detection are presented in **Figure 5**.

2.5. Touch Detection

At this stage in processing, the shadow tip information along with the fingertip information are both available and sufficient in achieving touch detection. This touch detection is based upon the observation that, the finger touching the keyboard surface causes its tip to converge with the tip of the shadow and therefore the distance between these two tips is sufficient in achieving touch detection. Therefore assuming that the fingertip is stored

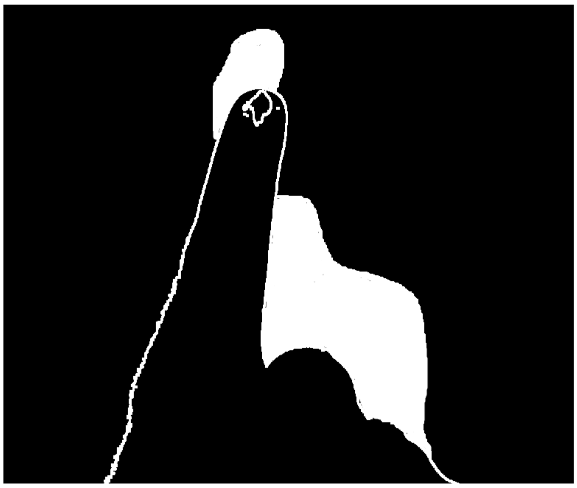


Figure 4. The results achieved at the shadow extraction detection stage of execution.

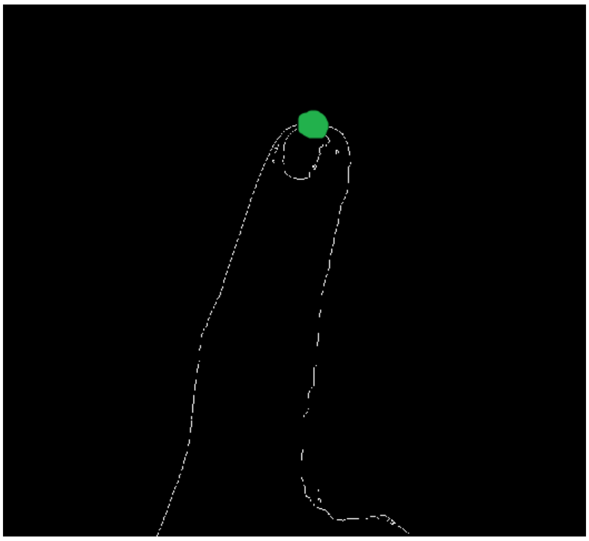


Figure 5. The results achieved at the Finger Tip detection stage of execution.

as a coordinate variable, $\left(x_f, y_f\right)$ and the shadow information is stored as $\left(x_s, y_s\right)$, the distance between the two coordinates is computable as:

$$\gamma = \left\|\left(x_s - x_f, y_s - y_f\right)\right\| \tag{7}$$

The results of this operation are illustrated in **Figure 6** and demonstrate that the scheme indeed suffices in achieving touch detection.

Although previous literature has proceeded into input mapping as the final stage of virtual key input, the mapping scheme is treated as beyond the scope of this paper as we deem this stage as tightly coupled with the application domain of the virtual keyboard system. This paper therefore focuses on how to efficiently realize robust virtual input while leaving the mapping scheme to be addressed by application specific research efforts.

3. Experimental Implementation and Results

In order to verify the sufficiency and efficiency of the proposed robust virtual key input scheme, computer-based experiments are carried out. The experiments are carried out on a conventional virtual keyboard hardware system with hardware specifications illustrated in **Table 1**.

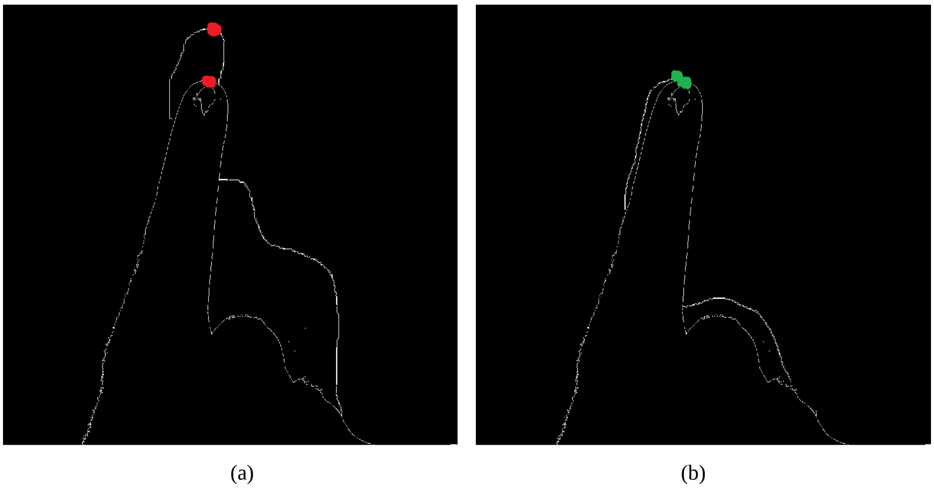


Figure 6. Touch detection according to the proposed scheme. (a) Illustration of no-touch scenario; (b) Illustration of touch scenario.

Table 1. Experimental platform towards verification of proposed scheme.

Equipment	Description
Acquisition Device	Logitech C210
Keyboard template	Standard QWERTY
Distance between keys	0.2 cm
Distance between camera and keyboard template	≈26 cm

This experimental setup takes into account the variability of shadows as well as the possibilities of false positives and false negatives. The experiments are broken down into two main categories. Firstly we evaluate the performance of the proposed key input scheme in a quantitative manner. This experimental approach is aimed at evaluating the key input efficiency of the scheme. As the experimental results illustrate, the proposed key input scheme is capable of handling situations where a key input generates multiple tip points leading to ambiguity which may ultimately result in a false touch being categorized as a true touch. This is a phenomenon that is common in such approaches where shadow information is extracted towards key input. In order to handle such situations, the proposed text input scheme is designed in order to continually search the given space for the fingertip as well as the shadow-tip distance that minimizes the overlapping pixels towards one. Before proceeding to the experimental results, we briefly define the measurement metrics adopted towards the evaluation of the algorithm. The *False Touch* metric represents a touch scenario in which a user does not actually intend to enter a certain key but due to a certain level of overlap between fingertip and shadow tip, a touch event is recorded. In this scenario, although an overlap occurs between the two tips, the level of overlap is far from the predefined threshold, representative of a key input. The *Near Touch* metric is similar to the *False Touch* with the only distinguishing feature being that in near touch events, the level of overlap between shadow and finger tips is significantly closer to the predefined threshold representative of a touch event. The final metric, the *True Touch*, signifies a scenario in which the user intends to enter a particular key and a touch event is appropriately recorded. The proposed approach is shown to be effective in handling cases of false positive even in situations when illumination is significantly low.

As illustrated in **Table 2**, the proposed approach is capable of achieving a true touch detection rate above 90 %, with an average false touch rate of 12.9% across all four experiments illustrated in **Table 2**. This suggests that the key input scheme is applicable as an acceptable key input method. The ability of the scheme to achieve compatibility in online systems is also further established by its ability to operate at an average of 14 frames per second.

Furthermore, in order to establish the proposed scheme alongside the classical and state-of-the-art virtual key input schemes via shadow analysis, we compare the proposed scheme with the work in [9] and [21] respectively. The results of this comparison are presented in **Table 3**.

Table 2. Quantitative evaluation of key input accuracy of proposed scheme.

Experiment	False Touch (%)	Near Touch (%)	True Touch (%)
1	13.3	10.7	92.4
2	12.6	12.3	92.2
3	13.5	11.0	94.8
4	12.4	9.3	91.6

Table 3. A comparison of the proposed scheme with classical and state-of-the-art.

Parameter	Algorithm in [8]	Algorithm in [19]	Proposed Scheme
Color Space towards hand detection	RGB	HSV	RGB
Filtering Scheme	None	Median filter with 3 × 3 mask	Median Filter by 5 × 5 mask
Edge Enhancement Scheme	Edge Thickening	Morphological Processing	Morphological Processing
Tip extraction schemes	Sequential Execution	Sequential Execution	Parallel Execution

4. Conclusion

As the results attained through experiments demonstrate, the proposed key scheme is capable of attaining accuracies up to 94.8 in true touch situations. This performance is accompanied by relatively lower false touch and near touch scores which demonstrate that the scheme is not only efficient in facilitating key input but also has the ability to tackles and address false and near touch situations. This performance is attributed to the efficient preprocessing applied towards the alleviation of noise and its effects. Morphological operations which are further relied upon in highlighting necessary components of the target image such as shadow tip information and finger tip information. This efficient and targeted preprocessing allows for the features crucial to key input to be highlighted while dimming out unnecessary features by treating them as noise. This proposed virtual key input via shadow analysis is also lightweight and achievable without the need for additional hardware or without significantly increasing software complexity. Furthermore, the parallel execution capability of the scheme, which allows for hardware implementations to breakup and execute simultaneously, the shadow and finger tip feature extraction and processing components, allows for considerable speed improvements and optimizations to be achieved in real-time applications. This is a topic that merits further research effort. In conclusion, the proposed scheme is highly feasible with existing virtual key input systems towards robustification of their performances through the adoption of already existent and sometimes abundant shadow information.

Acknowledgements

The authors would like to express appreciation for the support received from the 604 research lab under the School of Automation Science and Engineering which has facilitated the realization of this research effort.

References

[1] Kölsch, M. and Turk, M. (2002) Keyboards without Keyboards: A Survey of Virtual Keyboards. UCSB Technical Report 2002-21.

[2] Sarcar, S., Ghosh, S., Saha, P.K. and Samanta, D. (2010) Virtual Keyboard Design: State of the Arts and Research Issues. *IEEE Students' Technology Symposium* (*TechSym*), Kharagpur, 3-4 April 2010, 289-299. http://dx.doi.org/10.1109/techsym.2010.5469165

[3] Hagara, M., Pucik, J. and Kulla, P. (2013) Specification of Camera Parameters for Virtual Keyboard. 23*rd International Conference on Radioelektronika* (*RADIOELEKTRONIKA*), Pardubice, 16-17 April 2013, 227-231. http://dx.doi.org/10.1109/radioelek.2013.6530921

[4] Ghosh, S., Sarcar, S., Sharma, M.K. and Samanta, D. (2010) Effective Virtual Keyboard Design with Size and Space Adaptation. *IEEE Students' Technology Symposium* (*TechSym*), Kharagpur, 3-4 April 2010, 262-267.

[5] Gelormini, D. and Bishop, B. (2013) Optimizing the Android Virtual Keyboard: A Study of User Experience. *IEEE*

International Conference on Multimedia and Expo Workshops (*ICMEW*), San Jose, 15-19 July 2013, 1-4. http://dx.doi.org/10.1109/icmew.2013.6618310

[6] Gong, S., Lin, J. and Sun, Y. (2010) Design and Implementation of Anti-Screenshot Virtual Keyboard Applied in Online Banking. *International Conference on E-Business and E-Government* (*ICEE*), Guangzhou, 7-9 May 2010, 1320-1322. http://dx.doi.org/10.1109/icee.2010.337

[7] Topal, C., Benligiray, B. and Akinlar, C. (2012) On the Efficiency Issues of Virtual Keyboard Design. *IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems* (*VECIMS*), Tianjin, 2-4 July 2012, 38-42. http://dx.doi.org/10.1109/vecims.2012.6273205

[8] Adajania, Y., Gosalia, J., Kanade, A., Mehta, H. and Shekokar, N. (2010) Virtual Keyboard Using Shadow Analysis. 3*rd International Conference on Emerging Trends in Engineering and Technology* (*ICETET*), Goa, 19-21 November 2010, 163-165. http://dx.doi.org/10.1109/icetet.2010.115

[9] Zheng, Z., Yang, K. and Pei, J. (2011) Design and Implement of a Kind of Virtual Keyboard Based on Microcomputer and CMOS Camera. *IEEE* 13*th International Conference on Communication Technology* (*ICCT*), Jinan, 25-28 September 2011, 333-336. http://dx.doi.org/10.1109/icct.2011.6157891

[10] Perez, C.A., Pena, C.P., Holzmann, C.A. and Held, C.M. (2002) Design of a Virtual Keyboard Based on Iris Tracking. *Proceedings of the Second Joint* 24*th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference on Engineering in Medicine and Biology*, **3**, 2428-2429. http://dx.doi.org/10.1109/iembs.2002.1053358

[11] Du, H. and Charbon, E. (2008) A Virtual Keyboard System Based on Multi-Level Feature Matching. *Conference on Human System Interactions*, Krakow, 25-27 May 2008, 176-181. http://dx.doi.org/10.1109/hsi.2008.4581429

[12] Usakli, A.B. and Gurkan, S. (2010) Design of a Novel Efficient Human-Computer Interface: An Electrooculagram Based Virtual Keyboard. *IEEE Transactions on Instrumentation and Measurement*, **59**, 2099-2108.

[13] Babic, R.V. (2002) Sensorglove—A New Solution for Kinematic Virtual Keyboard Concept. *Proceedings of* 1*st International IEEE Symposium on Intelligent Systems*, **1**, 358-362. http://dx.doi.org/10.1109/is.2002.1044282

[14] Sharma, M.K., Dey, S., Saha, P.K. and Samanta, D. (2010) Parameters Effecting the Predictive Virtual Keyboard. *IEEE Students' Technology Symposium* (*TechSym*), Kharagpur, 3-4 April 2010, 268-275. http://dx.doi.org/10.1109/techsym.2010.5469160

[15] Fan, P., Hao, X. and Zhou, H. (2010) Design and Implementation of Network-Based Virtual Keyboard for the Remote Alarm Supervisory System. 2*nd Pacific-Asia Conference on Circuits*, *Communications and System* (*PACCS*), **1**, 242-244.

[16] AlKassim, Z. (2012) Virtual Laser Keyboards: A Giant Leap towards Human-Computer Interaction. *International Conference on Computer Systems and Industrial Informatics* (*ICCSII*), Sharjah, 18-20 December 2012, 1-5. http://dx.doi.org/10.1109/iccsii.2012.6454614

[17] Kwon, T., Na, S. and Park, S.-H. (2014) Drag-and-Type: A New Method for Typing with Virtual Keyboards on Small Touchscreens. *IEEE Transactions on Consumer Electronics*, **60**, 99-106.

[18] Na, S. and Kwon, T. (2014) RIK: A Virtual Keyboard Resilient to Spyware in Smartphones. *IEEE International Conference on Consumer Electronics* (*ICCE*), 10-13 January 2014, Las Vegas, 21-22.

[19] Posner, E., Starzicki, N. and Katz, E. (2012) A Single Camera Based Floating Virtual Keyboard with Improved Touch Detection. *IEEE* 27*th Convention of Electrical & Electronics Engineers in Israel* (*IEEEI*), Eilat, 14-17 November 2012, 1-5. http://dx.doi.org/10.1109/eeei.2012.6377072

[20] Beham, M.P. and Gurulakshmi, A.B. (2012) Morphological Image Processing Approach on the Detection of Tumor and Cancer Cells. *International Conference on Devices*, *Circuits and Systems* (*ICDCS*), Coimbatore, 15-16 March 2012, 350-354.