Digital Beethoven - An Android Based Virtual Piano

Hira Begum*, Saima Shaheen*, Momina Moetesum*, and Imran Siddiqi*
*Department of Computer Science
Bahria University Islamabad.
Emails: hira.asif63@yahoo.com, raniishmal@gmail.com, momina.moetesum@bui.edu.pk, imran.siddiqi@bahria.edu.pk

***Abstract*—Smart phones with high resolution cameras have enabled innovative computer vision applications that apply techniques like real-time image processing and virtual reality to provide a close approximation of reality. In this paper, we present an Android application that allows users to play a virtual piano by using a keyboard drawn on a piece of paper. The application allows the user to point the camera of a hand held device towards the keyboard, and process image of the paper keyboard in real time. The application then detects fingers placed on a key and after key detection plays the corresponding sounds. Initial results demonstrate the potential of such an application in providing a viable replacement for heavy and expensive instruments.**

I. Introduction

There is a rapidly growing interest of the research community in the design and development of innovative human computer interfaces [2]. The desire to improve user experience and to enable ubiquitous computing is accelerating the transition from desktop and web-based applications to mobile applications.

As mobile devices are becoming smaller and more powerful, users' expectations are growing higher. Advancement in augmented and virtual reality has opened new vistas for mobile application development [1], [4], [19]. Nevertheless mobile applications greatly depend on the device characteristics, platform and operating system. Diversity present in the mobile devices makes it difficult to provide methods to make such applications more expressive and user friendly.

Musical applications on mobile devices also suffer from challenges like small keyboard and limited screen size making it inconvenient for the user to play the instrument. The externally connected keyboard may provide convenience of feel but lacks portability. Computer vision and image processing techniques have allowed developers in every field to come up with interesting solutions and possibilities for various challenges. An image of an instrument or a real time capture of frames can be processed to overcome issues like small key size and portability. However, as attractive as it may sound, most vision based solutions come with their own set of challenges. This paper proposes a mobile virtual piano based on vision based techniques and discusses the pertaining challenges and their possible solutions. The developed application captures the image of the paper keyboard in real-time and prompts the user to press the keys just as playing a regular piano keyboard. The application detects the fingers using skin tone. Once localized the application maps the finger with the corresponding key. Once correctly detected, the pressed key note sound is played on the mobile.

The paper is organized as follows. Section II details the related works in expressive piano playing applications, Section III explains the proposed methodology, Section IV describes the experimental setup and performance requirements while Section V concludes the paper.

II. Related Works

With the advent of supporting technologies, new interfaces for musical expressions are also emerging. New musical interfaces [6], is a multidisciplinary field that relates technologies like portable audio and computer generated sounds with aesthetics.

Mapping gestures and computer commands has been an area of interest for the HCI community for long [16]. Nevertheless mapping gestures and sounds offers a variety of challenges mainly providing the feel and expressiveness of a contemporary musical instrument. In the following, we discuss some of the significant works that have been proposed to provide innovative interfaces for expressive piano playing.

Authors in [8] present the design and implementation of an augmented piano using a Kinect depth camera, video projector and supporting hardware. The proposed application runs on a desktop and applies gesture recognition techniques to map hand movements to piano keynotes. A similar application is presented in [18], where authors utilize a Kinect profundity camera and a video projector to make a 3-D motion space over the console. The Kinect depth camera catches 3-dimensional information on the motion space, as a crude video stream. The stream is then passed through foundation and clamor evacuation and sustained into a blob identification calculation. With multi-dimensional motions a solitary hand can likewise control numerous parameters as opposed to attempting to control various physical controls.

Another study [5] presents a desktop application which projects an augmented piano keyboard on the computer screen. The prototype uses sensor information to detect finger movements on the projected keyboard via two modes i.e. infrared and ultrasonic and simulates piano playing. Infrared visual tracking systems such as Microsoft Kinect and Leap Motion cameras are used to detect speed of the moving fingers. For ultrasound detection, a system using an Arduino board is designed. It triggers signals through a pin corresponding to it and then waits for its response. The response time is directly proportional to the distance of objects with that of corresponding sensors. The study describes the challenges of finger detection and corresponding key stroke and states that ultrasound based detection produced better results than infrared based detection.

The above mentioned applications are characterized by the need of special hardware and complex programming. Another form of interface, based on computer vision techniques, is gaining popularity. In a study [11], authors classify human expressions and play a music track best suited for the detected emotion. An inbuilt camera is used to capture the facial expressions of the user which omits the requirement for supporting hardware. On a similar principle, another study [13], presents a desktop application which also uses the front camera of a laptop to capture hand movements of the user. By processing the captured frames in real-time, the system recognizes users actions and outputs appropriate music to the speakers.

Recently mobile based human computer interfaces have attracted the research community. Smartphones with high resolution cameras and greater processing power enable real time video processing which can be applied to perform various activities. A number of interesting applications [9], [14], [20] have been designed. Literature review suggests that the amount of work carried out in this area needs further exploration to design applications and interfaces which provide the feel and expressiveness of playing a musical instrument with less amount of supporting hardware. Keeping in view these requirements, a vision based Android application is presented in this paper which uses a piano keyboard drawn on a paper. Users can point the camera of a hand held device on the paper piano and play it in real time. The details of the employed methodology are presented in the following section.

III. Proposed Methodology

The proposed system is an Android based application that can be used on smart phones. The application applies real-time image processing techniques to provide simulated functionality to hand drawn/printed piano keys. The methodology can be divided into following phases: Initialization, Detection and Recognition of keys, Detection and Localization of fingers and playing of sounds. An overview of these steps is shown in Figure 1.

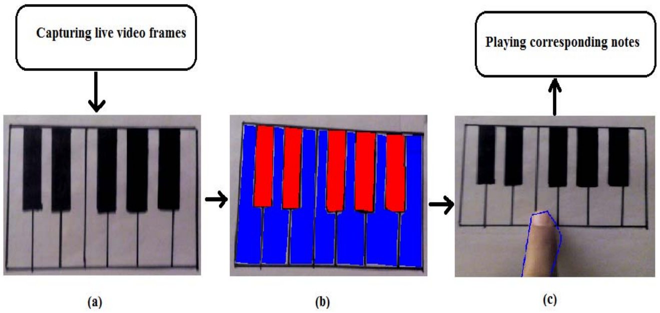

Fig. 1. An overview of the proposed system: (a) Initialization (b) Keys detection and recognition (c) Finger detection and localization

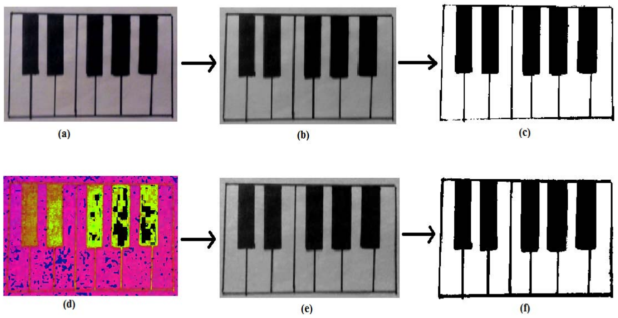

Fig. 2. (a) Image in RGB space (b) Grayscale image (c) Binarized image (d) Image in HSV space (e) Channel with intensity information (f) Binarized image

A. Initialization

At the start of the process, a frame of the keyboard without fingers is captured. Since the application is intended to be used in different environments with varying light and shadow effects, the captured frame is converted from RGB to HSV color space. HSV (Hue, Saturation and Value) color space enables separation of color information from intensity which is not possible in RGB color space. Hence, the binary image resulting from intensity/value channel of HSV color space is better than that acquired from an RGB image. Figure 2 shows the resultant binary images from both RGB and HSV color space.

B. Keys Detection and Recognition

Once the captured frame is binarized, key detection and recognition is initiated. Object detection techniques can be broadly divided into two categories [21] i.e. Contour detection techniques and Region detection techniques. Contour based techniques detect and connect edge pixels to form contours. These techniques are usually less complex, hence preferable, however edge of a region can often be hard to compute because of noise and occlusion. On the contrary, region based methods cover more pixels than edges and thus provide more information to characterize the area of interest. Nevertheless these techniques usually require color and texture information in addition to structure, hence are complex and

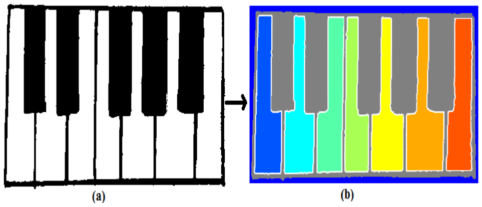

Fig. 3. (a) Binarized image (b) Detection of white keys

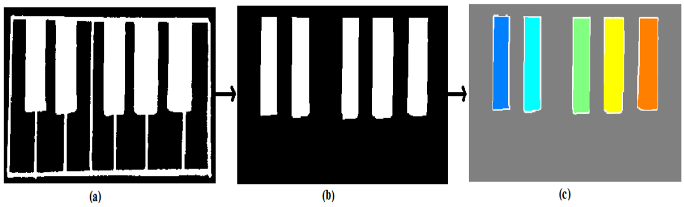

Fig. 4. (a) Inverted binarized image (b) Image after morphological erosion (c) Detection of black keys

computationally expensive. Our piano keyboard consists of seven white and five black keys. In order to detect white keys, we have applied contour finding technique which utilizes border following algorithm based on Moore-Neighbor tracing algorithm [7]. Once the boundaries of white keys are detected (Figure 3), their coordinates are stored and associated with corresponding sounds. Once white keys are detected, the image is inverted. By applying morphological erosion on the inverted image, keyboard border and white key borders are removed. The remaining components are the black keys. Same border following algorithm is then applied to detect contours of the black keys as depicted in Figure 4. This method is deemed suitable for its fast performance and reliability.

C. Finger Detection and Localization

After the initialization and key detection and recognition phases, the application prompts the user to press the keys on the paper keyboard. The camera captures the frame with the finger and processes it to play the corresponding notes. Finger detection and localization can be divided into two steps. Before deciding the corresponding pressed key, the application should first detect the finger. A skin tone detection technique similar to [15] is applied for finger detection. In computer vision, skin tone detection is a challenging task. Various schemes have been proposed in the literature [10], [17]. Studies [3], [12] suggest that skin tone detection in HSV color space yield better results especially in unconstrained environments. Converting the image to HSV splits it into three different components as hue, saturation and value based on the color (chrominance) and intensity information. The corresponding histogram are then generated and (skin tone) threshold values are computed (Figure 5), by experimenting on a number of images.

Masking is applied to the original image to detect skin pixels. Objects smaller than a threshold size are removed. Borders are smoothed and region filling is then applied. Once finger is detected, the application determines its position on the keyboard by locating position of the finger tip as shown in Figure 6. This is achieved by tracing the boundary of the region with skin tone pixels and then computing maxima of the boundary pixels. The pixel representing the maxima is also the member of a key which was recognized earlier. After determining the corresponding key location, respective sound is played. Hence giving an impression of playing a real piano.

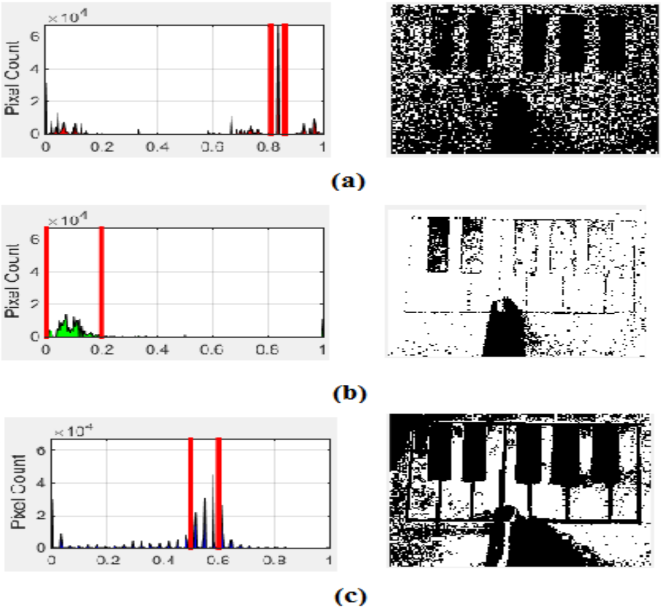

Fig. 5. (a) Hue mask and its histogram with selected thresholds (b) Saturation mask and its histogram with selected thresholds (c) Value mask and its histogram with selected thresholds

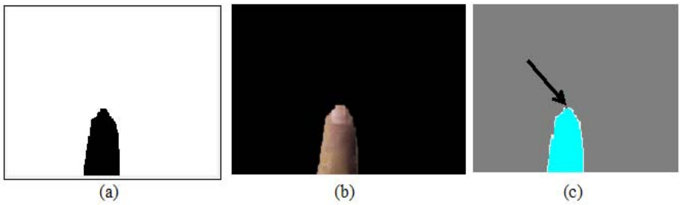

Fig. 6. (a) Mask (b) Finger detection (c) Finger tip localization

IV. Experimental Setup and Results

As mentioned in the previous sections, our proposed application runs on Android by processing real-time frames to provide simulated functionality to hand drawn/printed piano keys. Android Studio was used as the development environment. The performance of our proposed application is greatly dependent on the hardware on which it is being installed. We tested our application on five smartphones with various specifications to determine the optimal functional requirements of our application. Table I gives an overview of the hardware specifications of the machines involved in this study.

TABLE I
HARDWARE SPECIFICATION OVERVIEW

Device	Android Version	Clock	Camera	Display
Galaxy S5	4.2.2 (KitKat)	Quad Core 2.5 GHz	16MP	5.1 inches
Galaxy J1	Ace 4.4.4 (KitKat)	Dual Core 1.3 GHz	5MP	4.3 inches
Huwawei Honor 3C	4.2.2 (JellyBeans)	Quad Core 1.3 GHz	8MP	5.0 inches
HTC A12	4.4.4 (KitKat)	Quad Core 1.2 GHz	8MP	4.7 inches
QMobile A400	4.2.2 (JellyBean)	Dual Core 1.2 GHz	5MP	6.0 inches

TABLE II
SYSTEM PERFORMANCE IN A SERIES OF 10 EXPERIMENTS

Subject	Keys Pressed	Correctly Played	Accuracy
1	12	10	83%
2	15	14	93%
3	14	12	86%
4	10	9	90%
5	12	11	92%
6	10	9	90%
7	15	14	93%
8	15	15	100%
9	10	10	100%
10	14	12	93%
Total	**127**	**117**	**92%**

Although latest smartphones feature wider screens, nevertheless limited display size is a constraint for applications such as ours. Due to this reason, we have restricted our paper piano keyboard to twelve keys i.e. seven white keys and five black keys. We tested our proposed application on smartphones with different screen sizes ranging from 4.0 inches to 6.0 inches. The performance degraded as we increased the number of keys. This is due to the fact that inclusion of more keys required reduction of key width which caused overlapping and incorrect localization of finger tip.

Since the proposed application processes video frames in real time, therefore For optimal performance it will require at least 1 GB of RAM and 1.2 GHz of processor to improve latency issues. To ensure compatibility, the application was tested on different versions of Android as shown in Table I. The application performed well on all of these.

As with all computer vision based applications, good quality camera and light conditions are necessary for optimal performance. Our proposed application produces good results in both indoor and outdoor environments under suitable light conditions. A camera with 5 MP and above provides good support for our application.

From the view point of quantitative evaluation of the effectiveness of the image analysis techniques employed in our study, ten subjects were asked to use the application and play the piano. Each subject pressed a different number of keys and for each session the number of correctly played keys was recorded. The results of these ten experiments are summarized in Table II where it can be seen that an overall accuracy of 92% is realized. The errors mainly stem from the inability to detect the finger or correctly localize it. Nevertheless, correctly playing 92% of the keys in real time on a virtual piano is indeed promising.

V. CONCLUSION AND FUTURE WORKS

In this paper, we presented a vision based mobile application which allows user to simulate piano playing by pressing keys on a paper keyboard. The proposed application can be categorized as a new interface for musical expressions. The idea presented in this study can be modified to perform other tasks like paper QWERTY keyboard and virtual calculator etc. The main objective is to provide improved user experience. Although the proposed application is in its infancy and is limited in its functionality, nevertheless it can enhanced into a complete prototype that can provide users with the feel of playing a real piano ubiquitously.

ACKNOWLEDGMENT

The authors would like to thank Bahria University Islamabad for providing necessary financial and technical support for carrying out this research.

REFERENCES

[1] Hyojoon Bae, Mani Golparvar-Fard, and Jules White. High-precision vision-based mobile augmented reality system for context-aware architectural, engineering, construction and facility management (aec/fm) applications. *Visualization in Engineering*, 1(1):3, 2013.

[2] Paul Cairns and Anna L Cox. *Research methods for human-computer interaction*, volume 12. Cambridge University Press Cambridge, 2008.

[3] Jose M Chaves-González, Miguel A Vega-Rodríguez, Juan A Gómez-Pulido, and Juan M Sánchez-Pérez. Detecting skin in face recognition systems: A colour spaces study. *Digital Signal Processing*, 20(3):806–823, 2010.

[4] Tosti HC Chiang, Stephen JH Yang, and Gwo-Jen Hwang. An augmented reality-based mobile learning system to improve students' learning achievements and motivations in natural science inquiry activities. *Journal of Educational Technology & Society*, 17(4):352, 2014.

[5] Yuri De Pra, Federico Fontana, and Linmi Tao. Infrared vs. ultrasonic finger detection on a virtual piano keyboard. In *ICMC*, 2014.

[6] Dalia El-Shimy and Jeremy R. Cooperstock. User-driven techniques for the design and evaluation of new musical interfaces. *Computer Music Journal*, 40(2):35–46, 2016.

[7] Abeer George Ghuneim. Contour tracing, 2009.

[8] Aristotelis Hadjakos. Pianist motion capture with the kinect depth camera. In *Proceedings of the Sound and Music Computing Conference*, pages 303–310, 2012.

[9] Rabia Jafri, Syed Abid Ali, Hamid R Arabnia, and Shameem Fatima. Computer vision-based object recognition for the visually impaired in an indoors environment: a survey. *The Visual Computer: International Journal of Computer Graphics*, 30(11):1197–1222, 2014.

[10] Praveen Kakumanu, Sokratis Makrogiannis, and Nikolaos Bourbakis. A survey of skin-color modeling and detection methods. *Pattern recognition*, 40(3):1106–1122, 2007.

[11] Sushmita G Kamble and AH Kulkarni. Facial expression based music player. In *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*, pages 561–566. IEEE, 2016.

[12] Rehanullah Khan, Allan Hanbury, Julian Stöttinger, and Abdul Bais. Color based skin classification. *Pattern Recognition Letters*, 33(2):157–163, 2012.

[13] Hui Liang, Jin Wang, Qian Sun, Yong-Jin Liu, Junsong Yuan, Jun Luo, and Ying He. Barehanded music: real-time hand interaction for virtual piano. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 87–94. ACM, 2016.

[14] Zhihan Lv, Alaa Halawani, Shengzhong Feng, Shafiq Ur Réhman, and Haibo Li. Touch-less interactive augmented reality game on vision-based wearable device. *Personal and Ubiquitous Computing*, 19(3-4):551–567, 2015.

[15] VA Oliveira and A Conci. Skin detection using hsv color space. In *H. Pedrini, & J. Marques de Carvalho, Workshops of Sibgrapi*, pages 1–2, 2009.

[16] Siddharth S. Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, Jan 2015.

[17] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreeva. A survey on pixel-based skin color detection techniques. In *Proc. Graphicon*, volume 3, pages 85–92. Moscow, Russia, 2003.

[18] Qi Yang and Georg Essl. Augmented piano performance using a depth camera. *Ann Arbor*, 1001(2012):48109–2121, 2012.

[19] Zornitza Yovcheva, Dimitrios Buhalis, and Christos Gatzidis. Smartphone augmented reality applications for tourism. *E-review of tourism research (ertr)*, 10(2):63–66, 2012.

[20] Ihab Zaqout, Samar Elhissi, Aya Jarour, and Heba Elowini. Augmented piano reality. *International Journal of Hybrid Information Technology*, 8(10):141–152, 2015.

[21] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19, 2004.