Paper Piano – Shadow Analysis based Touch Interaction

Boga Vishal
Department of Electronics and Communication
Manipal Institute of Technology
Manipal, India
bogavishal7@gmail.com

Deepak Lawrence K
Department of Aeronautical and Automobile Engineering
Manipal Institute of Technology
Manipal, India
deepak.lawrence@manipal.edu

***Abstract*—Recent years have marked a sharp increase in the number of ways in which people interact with electronic devices. Keyboards and computer mouse devices are not the only way users interact with their computers. This paper aims to investigate and demonstrate a human-computer interaction application in which a user can interact with the layout of a piano on a sheet of paper using computer vision. The image is acquired using a general-purpose camera from a single direction. The work presents details about methods of piano key detection and labelling, hand movement detection, fingertip detection and tracking, and touch interaction based on the shadow of the finger. K-means clustering is used to segment the hand and its shadow. Contour analysis is used in determining the shape and position of the keys. Shadow analysis aids in determining if the finger is touching the paper, even without any depth data from the camera. The detected key press then triggers a sound. The paper covers the details of the development of the virtual piano from concept to realization.**

***Index Terms*—Image segmentation, fingertip detection, finger tracking, shadow analysis, contours, human-computer interaction.**

I. Introduction

With the advent of technological advancements in the world of machines, it has become increasingly apparent that the design and implementation of those machines plays a massive role in making the lives of people easier [1]. However, for decades, the nature of the interaction between humans and machine seems to be unchanged. Considering the case of personal computers, the way users interact with them still is the same as it was about 40 years ago. Keyboards and computer mouse devices have proven to be irreplaceable, but with the increase in the number of portable devices like smartphones and virtual reality headsets, there is a need to explore other avenues and platforms with similar functionalities to those interaction devices existing today.

The requirement for devices that aid in interaction with machines also poses a question regarding affordability and cost-effectiveness. Although advancements in holographic interaction, augmented reality, virtual reality are changing the way humans interact with machines [2]; they are sometimes very costly because they are relatively new concepts and use advanced hardware components. Thus, it is essential to find substitute methods of interaction which address these problems, especially in the domains of cost and portability.

This paper discusses a proposed solution for the problem presented above. Paper Piano is an application of human-computer interaction that demonstrates the interaction with a paper with the layout of a piano printed to produce sounds. This device is effectively a piano with no actual keys. A camera which is attached above the setup (see Figure 1) takes the images in real time, processes it and triggers a sound based on the position of the finger. It just uses a single camera and paper. Cameras are ubiquitous in the present world, making them very economical. Thus, it is a straightforward, easy to use, portable and a low-cost way for musicians and learners alike to use the piano.

One significant challenge in making this algorithm successfully work is to detect whether the finger is touching the paper precisely. Touch detection becomes difficult because the camera has no depth information [3] and it just has a single view of the scene. This difficulty is surpassed using the concept of shadow analysis.

This concept also has the potential to make many devices affordable to people. The piano is just one device where this concept can be applied. The same concept can be used in numerous ways to interact with different types of machines, for example, a desktop setup with no keyboard and mouse. This algorithm was implemented using the OpenCV library.

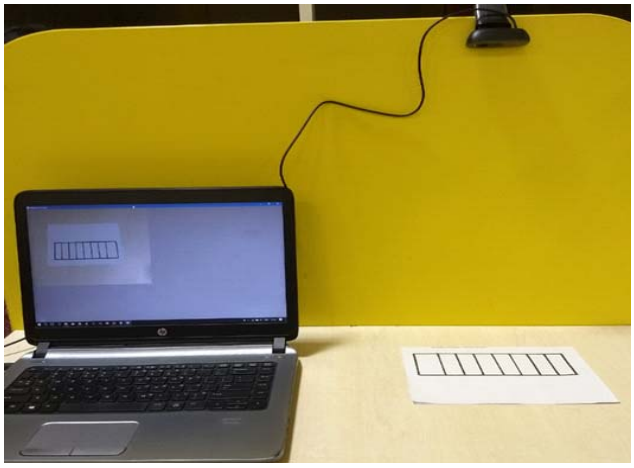

Fig. 1. The experimental setup of the Paper Piano

978-1-5386-2989-5/17/$31.00 ©2017 IEEE

II. Image Acquisition

The first task is to acquire image data from the camera. The algorithm that this paper presents takes an input of a continuous series of images. However, before starting this process, the system must be calibrated to the ambient light conditions. Calibration is an essential process because the overall working, accuracy, and precision depend highly on the ambient lighting conditions. Factors like the direction of incident light are also important.

The camera is set up above the working area, overlooking the paper and the surface on which it is placed. This working area is the area where the piano layout is present. For this calibration task, two images are captured for setting up the initial parameters of the algorithm and understanding the intensity components of various elements of the scene captured by the camera.

All the images that are captured by the camera are first converted to grayscale and then smoothed using a Gaussian kernel of size 5 × 5 before any other operations are performed on the images. Equation 1 shows the mathematical expression for a Gaussian kernel. This removes the high-frequency and noisy components in the image.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

The first reference image is that of the piano layout and its surroundings. It has to be ensured that there is good contrast between the colour of the piano layout and the surface on which it is placed. Contrast is essential as it ensures proper segmentation of the piano outline present on the paper in later stages of the process. The second reference image that is captured is used to learn the colour intensities of the performer's hand and its shadow. It has to be ensured that this picture has in it the person's hand up to the wrist and that its shadow is visible. This image is required as a set of specific intensity values of the shadow is supposed to be determined based the ambient as they vary with the lighting conditions. The values also change according to the position and time of the experiment.

Now that the two reference images are captured, the next step is to understand the image content, i.e., determine the location of the piano keys and extract essential information like the position of the fingers for further processing.

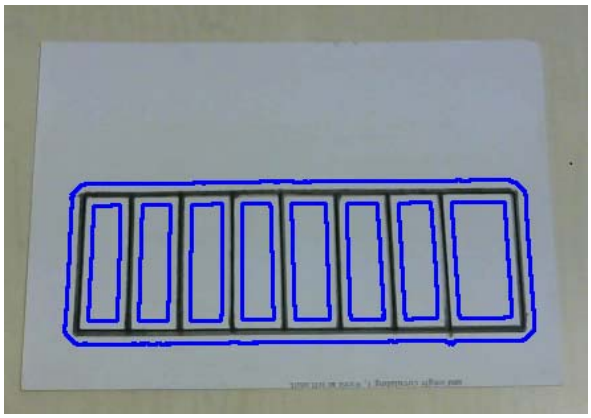

Fig. 2. The contours of all the keys of the layout. The outer boundary is also seen here

III. Calibration

This section will describe the procedure to calculate certain pixel intensities from the reference images. These values will be used later to extract the hand and its shadow. Calibration is an essential task because it is responsible for the adaptivity of the algorithm to different conditions. K-means clustering [4] is used to calculate these values for calibration.

Currently, there are no methods to determine the exact position of the shadow directly. There are research papers that describe the process of removing a shadow from an image. Those techniques are computationally complex and require extensive memory space. For this application, it is desired to have a technique that is easier and involves simple computations. The concept of image differences is used for this procedure. The first reference image is subtracted from the second one to remove the background and the piano layout. This subtraction results in a picture with a black background and pixels representing the hand and its shadow in it. Although a shadow is black, it is not entirely dark due to the ambient light.

The k-means clustering algorithm with k = 3 returns three sets of pixel values where each set represents either the hand, shadow or the background. The task now is to extract the intensity values of the shadow. It is evident that the background is the darkest (least pixel intensity values) and that the colour of the hand has the highest grayscale intensity value in the image. This knowledge of the general intensity values is used in calculating the grayscale intensity range of the shadow.

On arranging the mean of the total number of pixel values in each cluster in the ascending order, it is known that the shadow pixels belong to the second cluster. Pixel intensities belonging to this set of values represent the shadow intensities. The maximum and minimum values of this set are later used to segment the shadow region from the continuous image stream.

In a similar method, the hand's maximum and minimum intensity values can also be extracted. It has been experimentally observed that detection of skin using pixel intensity values is more robust in the YCrCb colour space compared to other colour spaces [5]. Hence, it is preferable to convert the image to YCrCb colour space before extracting the intensity values.

IV. Piano Key Labelling

This section deals with extracting and identifying the layout of the piano using information from the first reference image.

A. Edge Detection

First, the image is processed to identify its edges. This step is crucial as it reduces the amount of data to be processed in later stages. There are many methods in which edge detection can be performed. In this case, since the piano keys are straight lines, techniques like Hough Transform can be used in combination with an edge detector like Sobel operator or Scharr operator [6]. Though these methods are easy to implement to find the straight lines, it is to be noted that in the case of the piano layout the edges may not be straight. These methods also make it

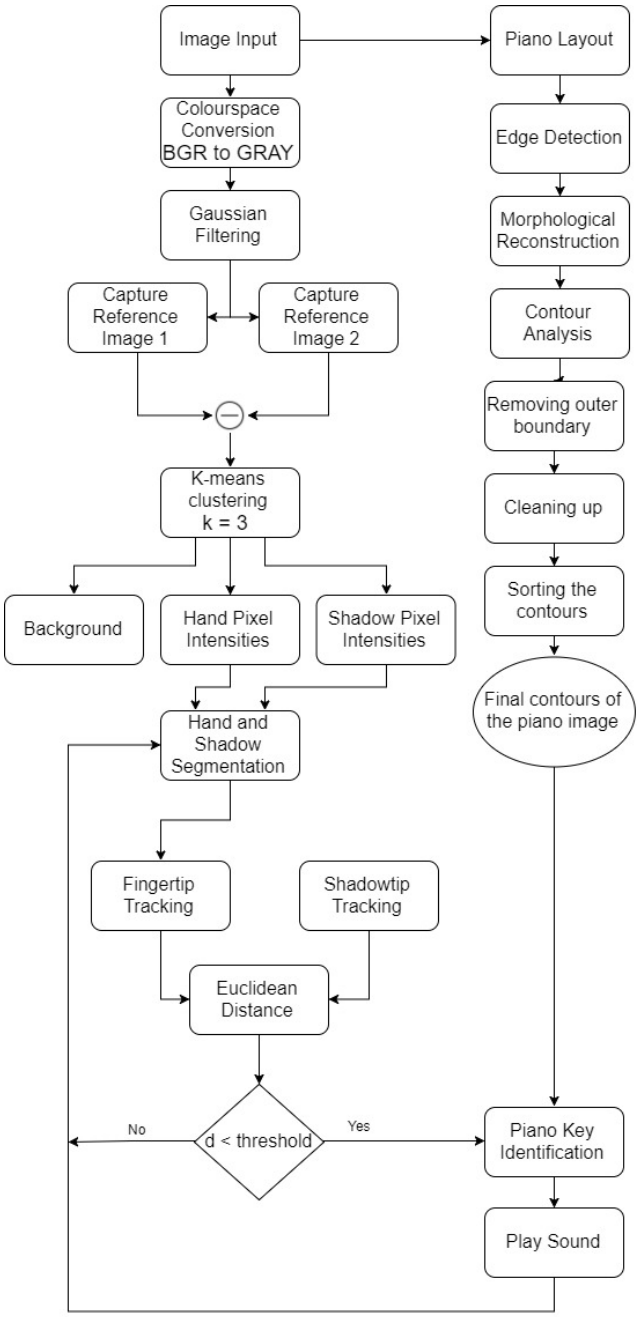

Fig. 3. The flowchart of the Paper Piano algorithm

challenging to locate a particular key based on the location of the touch detected. Hence, the algorithm being presented in this article takes a different approach and uses the Canny edge detection algorithm.

Canny edge detection algorithm [7] is a prevalent edge detection method that works on a simple thresholding principle known as hysteresis. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. This algorithm was tested using the hysteresis threshold values to be 75 and 200 for the grayscale images.

At the end of this process, the result is an image with many edges. These edges are very thin. It is easy to lose these thin edges in a smoothening or filtering process. To prevent that the edges are thickened. This also has other advantages. For example, if the user drew a layout or printed one with thick edges, this process would reconstruct the thin borders to form the right borders of the piano keys.

B. Morphological Reconstruction

Morphological reconstruction [8] is useful for constructing an image from small components or for removing features from an image, without altering the shape of the objects in the image. Morphological reconstruction works on grayscale images and binary images.

The morphological reconstruction process is based on a source image, a marker image, and marker points.

- Source Image—The source image (also referred to as the mask image), is the reference image used in the morphological reconstruction.
- Marker Image—The reconstruction process occurs on the marker image, which is created by applying dilations or erosions on the source image. The marker image can also be taken from existing images. The marker image should have the same dimensions as the source image.
- Marker Points—Marker points are user-specified points in the image that specify where the reconstruction process should start.

Dilation is a process that reconstructs bright regions. This effectively extends the bright regions outward in all sides resulting in an increase in their area.

Reconstruction by dilation reconstructs bright regions in grayscale images and reconstructs particles in binary images. Starting at the marker points, neighbouring pixels are reconstructed by spreading the brightness value. Reconstruction by dilation starts with the maximal gray valued pixels of the marker and reconstructs the neighbouring pixels ranging from 0 to the maximal valued pixel.

This paper's algorithm used the morphological image processing technique to make the edges of the piano layout more prominent. For this, a structuring element of 5 × 5 matrix cross shape was used.

Hence, dilation is applied to extend the thin edge lines outward. This makes them brighter and easier to segment, and in case there were thick borders, they will now be filled with bright intensity pixels in the final image after reconstruction. The next task would be to segment out these edges for further processing.

C. Contour Analysis

In the previous step, the final image had the edges of the piano layout and its background. Now, the edges have to be segmented out to determine the total number of piano keys and determine their position. It is important to note that the reference image is not solely comprising of the piano layout. It might have other objects (whose shape and size are not known) in the same image. Hence, the role of contour analysis is twofold. While being responsible for extracting all the edges of the keys, the algorithm also has to remove any other objects in the image from

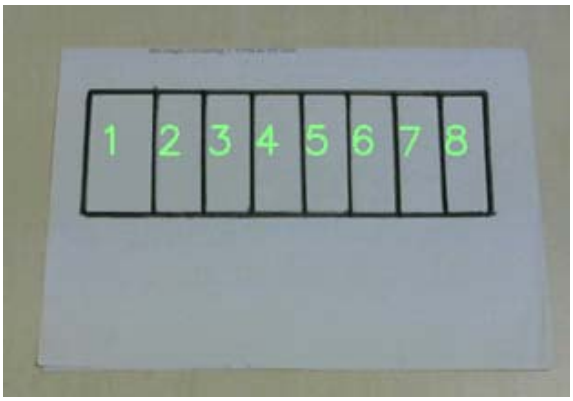

Fig. 4. Contours – Sorted and labelled in order

being falsely classified as a piano key. The process of eliminating these unwanted edges or objects is explained in detail in the following sections.

Contour analysis [9] is used here for extracting the key edges. Contours are the set of boundary pixels of an object that are connected. These points are stored in an array which represents the contour. For computing the contours, the objects in the image should have clear boundary edges to accurately distinguish its location from the image background, and this was taken care of in morphological reconstruction.

Once the contours are extracted (see Figure 2) and saved in the memory, the algorithm then verifies if they are piano keys or not by checking their shapes and sizes. The contours are first approximated as polygons. A piano key is always rectangular, and hence, every contour is checked for the number of sides after it is approximated. If the number of sides is 4, it is highly likely that the contour is a part of the piano. The algorithm also checks for size. If for a given image, the area of the contour is less than a certain amount (this value is dependent on the size of the piano layout), that contour is ignored and won't be used for further processing. For the present setting, if the area of the contour is less than 100 pixels, the contour was discarded. Once this procedure is complete, there still might exist a few contours in the list that do not correspond to piano keys. However, most of these contours are piano keys. The procedure for removing these outlier contours are described later.

D. Removing the Outer Boundary

The next task is to remove the unnecessary contours from the list that was compiled in the last step. One contour that needs to be removed is the outer boundary contour of the paper. This boundary looks like that shown in Figure 2. This contour forms due to the contrast between the paper colour and the surface on which it has been placed.

It is known that this is the largest contour in the image. So, to remove this contour, the areas of all the contours are calculated, and then the largest contour is removed from the list. The set of contours now is void of the largest contour that was supposed to be removed.

E. Cleaning Up

The algorithm performs one last size check. This step removes all the contours whose area is less than the threshold. Since almost all the contours in the list belong to piano keys and are rectangular in shape, they have almost similar areas to each other. However, this similarity is not true of other redundant contours. In this step, the algorithm calculates the average of the areas of all the contours. It then sets the threshold to be around 0.3 times the average. Any contour whose area is less than this area will be eliminated. At the end of this step, the set of contours only have the boundaries of the piano keys on the paper.

F. Sorting Contours

The next step is to assign a number to all these contours. This algorithm works for any number of keys. Sorting contours makes it easier to check for the key that is pressed. The assignment of numbers is done from left to right, and the final list of sorted contours is saved in memory for further use. Finally, at the end of the key labelling process, the outcome is a sorted in a left to right order and numbered list of contours that represent each of the piano keys on the paper as shown in Figure 4. This is a global list that will be used later in the algorithm to check for the key that the user pressed at a certain point in time.

V. Hand and Shadow Segmentation

This section discusses the how the hand and its shadow are segmented separately. From this section forward, the algorithm will be working on a continuous series of images. The continuous stream is processed to check for the hand and the shadow and their positions concerning the piano layout.

For segmenting the shadow region in each of the images from the continuous image stream, the first reference image is subtracted from it. This results in an image with the hand and the shadow in it. However, because of the edges of the piano layout, the hand region and its shadow are not exactly as accurate as the actual image. They now have lines (the piano layout's edge pixel intensities are a non-zero value; thus, they result in a different value that is different from the rest of the hand in the subtracted image) passing in between them. This is shown in Figure 5. These lines can be filled using morphological reconstruction [9].

For each image from the continuous stream, each subtracted image is thresholded using Otsu's binarization technique, and

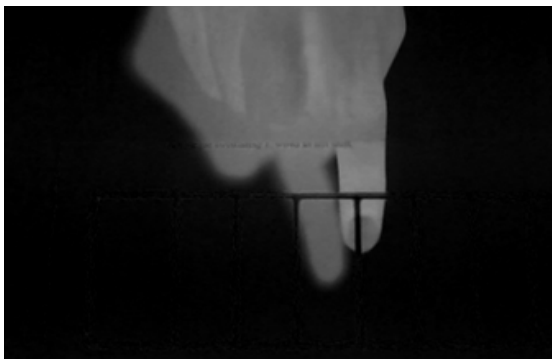

Fig. 5. The lines in between the fingers are due to the piano layout

this returns a binary image with hand and its shadow. This image undergoes dilation with an elliptic structuring element of size 5 × 5 so that the dark lines in between the bright regions and other small holes are filled. Once dilation is complete, the contours of the bright region are computer and a rectangular bounding box is computed around the contour. This bounding box is the region of interest (ROI). The bounding box is about 10 or 20 pixels longer on all sides. This is to ensure that the fingertip can be calculated accurately in a later section. If the tip of the finger is on the boundary of the ROI, it gives false results. So, extending the rectangular box improves the accuracy of the algorithm.

Performing computations only on the ROI instead of the complete image reduces the computational time because the size of the image (the number of pixels) to process has decreased. This increases the overall response time of the system which is a really important factor because it is running in real-time and has to produce sound instantaneously.

A. Hand Segmentation

The brightest cluster of the result of the k-means clustering is that of the hand. After segmenting the ROI, it is converted to YCrCb colour space. After experimenting with a lot of different colour spaces like HSV [10], and CIELAB, YCrCb was found to offer the best results in varying lighting conditions and illumination changes [5]. This makes the segmentation of the hand very simple. The ranges of values for detection the skin in YCrCb colour space are known. A thresholding function filters out all the pixels in the ROI that belong in this range of YCrCb intensities. Let this output image be the ROI's hand mask. In the present setting, the threshold values of the image components used were as follows as described in Equation 2.

$$0 < Y < 255;\ 133 < Cr < 173;\ 77 < Cb < 127 \quad (2)$$

The drawback of segmenting based on pixels is that it sometimes cannot segment some parts of the hand, especially the nails. This is due to changes in lighting conditions and spectral properties of different materials. This problem could be overcome by the use of reconstruction. The holes in the ROI's hand mask are filled by dilating the image.

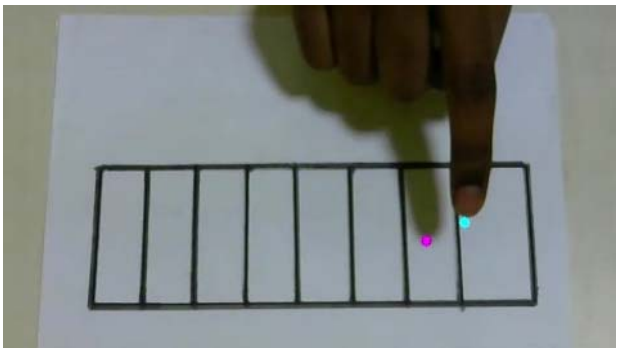

Fig. 6. Detection of fingertip and shadow tip. The pink dot indicates the shadow tip while the cyan dot indicates the fingertip

Once the dilation is done, the ROI's hand mask is a very accurate description of the actual hand. All further computations involving the hand and the fingers performed using this mask.

B. Shadow Region Segmentation

K-means clustering previously performed returned a cluster with pixel intensities of the shadow. From that cluster, the algorithm was able to compute the maximum and minimum intensity values. These values will be used to threshold the ROI and extract the shadow image.

Another reason that the ROI is larger than the actual contour bounding box is that the shadow can be further away from the hand region in the image. In cases where the shadow is very far, it falls outside the bounding box. This would not be a problem because the finger is obviously not touching the paper (this concept of shadows is explained in later sections). This further reduces the number of computations and improves the performance of the algorithm. This thresholding results in the ROI's shadow mask which will be used for tracking the fingers and determine the piano key that they are pressing.

At the end of this segmentation procedure, the algorithm has computed both the hand and the shadow regions of the present incoming image. It has been shown that the computation time is reduced significantly by working only on a region of interest instead of the complete image. In the trial run, it has been observed that the algorithm can works in real time with a speed of about 24 to 28 frames per second.

VI. Finger and Shadow Tracking

A. Finger Tracking

To track the user's finger the outer contour of the mask is computed, and the topmost point is calculated. This topmost point is the tip of the finger. This is done by finding the highest vertical pixel component, which is then labelled as the tip (topmost part) of the finger. Similar methods were used previously to detect and track fingertips [10]. However, those methods cannot be applied here as the viewing angle is perpendicular to the surface in the present case.

Other methods like tracking the fingernail based on its reflection properties [11] are also possible. However, for the present application of shadow analysis, this does not help because the shadow tip corresponds to that of the fingertip. Locating the nail does not directly determine the location of the fingertip. Hence, this algorithm uses contour analysis to accurately locate the fingertip in the image stream.

Once the fingertip is determined as in Figure 6, the corresponding shadow tip also has to be located. The algorithm currently works on one finger. It can be extended to work on multiple fingers in two ways. The algorithm can either be implemented to parallelly work in multiple fingers using multithreading, or it can sequentially check for fingertips.

B. Shadow Tracking

One crucial idea here is that since the camera is placed perpendicularly above the paper, the shadow is very close to the finger as long as it is close to or on the paper. The farther away that the finger or hand goes from the paper, the farther away is

the shadow. As the user lifts his hand, the shadow keeps moving away from the hand. This concept is the backbone of shadow analysis. One more intuitive idea is that when a touch is detected, both the finger and the shadow will be very close to each other.

Since the shadow is very close when the finger is touching the paper, there is no need to scan the whole ROI for the shadow tip. It is always near the finger. Hence, a bounding box is placed around the fingertip. It is usually 40 or 60 pixels in any direction. The coordinated of the bounding box is computed from the original reference image. This helps in later stages as the algorithm has to detect which key is being pressed.

The ROI's shadow mask was already computed in the previous section. Now, the algorithm checks if the topmost shadow point is in the bounding box region. This point is marked as the shadow tip as shown in Figure 6. This process is repeated for every incoming image. Now that both the finger's and the shadow's tips are located, it is time to detect which key is being pressed.

VII. Touch Detection

The touch detection algorithm works on the basis of Shadow Analysis that was discussed in the previous section. The fingertip and the shadow tip are already located. In this step, the Euclidian distance between them is calculated. A larger Euclidean distance means that that the shadow is far from the finger and thus the finger is not touching. A smaller Euclidean distance means that the shadow is closer to the finger and the finger is close to the paper. Based on experiments, a threshold value of this Euclidean distance can be calculated. Based on this distance, it can be determined whether the finger is touching the piano. When they are close to each other, their coordinates with respect to the original reference image are calculated. The corresponding contour in which the coordinates lie is computed using the shortest distance method which finds the shortest distance between a point in the image and a contour.

Once it is determined that the point is inside a particular contour, the sound of the key that contour represents is triggered and plays through the sound system. This proposed scheme and the program worked successfully in the present setting based on the various parameters values that are presented in various sections of this paper. However, there is a need for modification of the parameter values by the user based on the settings in which it is being used.

The shadow analysis method is accurate and effective in varying lighting conditions because all the thresholding intensity values are computed in real time based on the ambient lighting conditions. Overall, this method of touch detection is very effective, fast and cost-effective as the setup is simple and requires a single view camera.

A flowchart depicting the complete algorithm and different techniques used is shown in Figure 3.

VIII. Conclusion

The paper presents the design and implementation of a virtual piano based on computer vision. It has been demonstrated that it is possible to utilize a piece of paper as a piano, based on the principle of contour and shadow analysis that detects the position and touch of user's finger. The proposed approach is cost effective and portable and has the potential for positively influencing future applications such as online piano tutoring classes. The method presented is generic in nature, thus, giving people the ability to interact with any kind of surface. This can be extended to a large number of domains and can open many doors to future research about human-machine interactions.

References

[1] Eva Hudlicka, "To feel or not to feel: the role of affect in human-computer interaction," Int. J. Human-Computer Studies, vol. 59, pp. 1–32, 2003.

[2] Kil-Soo Suh and Young Eun Lee, "The effects of virtual reality on consumer learning: an empirical investigation," MIS Quarterly, vol. 29, no. 4, pp. 673–697, December 2005.

[3] Hui Liang, Jin Wang, Qian Sun, Yong-Jin Liu, Junsong Yuan, Jun Luo, and Ying Hey, "Barehanded music: real-time hand interaction for virtual piano," ACM Symposium on Interactive 3D Graphics and Games, pp. 87–94, February 2016.

[4] S. Lloyd, "Least Squares Quantization in PCM." IEEE Trans. On Information Theory, vol. 28, pp. 129–137, March 1982.

[5] Ekta Rewar, and Saroj Kumar Lenka, "Comparative analysis of skin colour based models for face detection," Signal and Image Processing: An International Journal, vol. 4, no. 2, April 2013.

[6] Adam Goodwin, and Richard Green, "Key detection for a virtual piano teacher," 28th International Conference on Image and Vision Computing, pp. 282–287, 2013.

[7] John Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, 1986.

[8] Luc Vincent, "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms," IEEE Transactions on Image Processing, vol. 2, no. 2, April 1993.

[9] Theo Pavlidis, Algorithms for Graphics and Image Processing, Computer Science Press, Rockville, Maryland, 1982.

[10] Chia-Hung Yeh, Wen-Yu Tseng, Jia-Chi Bai, Ruey-Nan Yeh, Sun-Chen Wang, and Po-Yi Sung, "Virtual piano design via single-view video based on multifinger actions recognition," Int. Conf. on Human-Centric Computing (HumanCom), pp. 1–5, 2010.

[11] T. S. Kumaran, J. Arul Murugan, and T. Sengolrajan, "Bare finger based touch detection in camera system," International Journal on Applications in Engineering and Technology, vol. 1, no. 2, pp. 12–15, February 2015.