Obtaining sequence of notes from guitar play-through

Mihir Sahasrabudhe Dhwanit Gupta Ayush Tewari

1 Introduction

This document is presented as part of a project for the course, *Digital Image Processing*, Monsoon 2012. It gives an introduction of the idea, what we wish to accomplish, and what we have accomplished as of the date mentioned above.

1.1 The Idea

The idea of this project is very simple: Given a video of a guitarist, playing his guitar, we wish to output the sequence of notes that was played. First, the terminology: A *guitar* is a string instrument consisting of a fretboard, on which are frets, and (normally) six strings. It is played by pressing a string on a specific fret, thereby giving the string a mount point, and allowing it to go in standing waves. A string in standing waves will produce sound. A *note* is obtained by pressing one string on one fret. Unlike *chords*, a note will normally be played using only one finger. Chords aren't being considered in this project, because that would be considerably difficult image segmentation task.

The idea now, is to perform segmentation and obtain, for every frame, the note that the guitarist played. Then, we'll output a video/audio/sheet music file which indicates what exactly was being played. As it is, this is a difficult problem, so we have made it a little easier by imposing several several constraints, which shall be introduced in later sections.

1.2 The Tool

We will be using the widely used software MATLAB as a tool and coding language for this project. MATLAB offers a vast library of image operations, and thus is the preferred language for several image processing tasks.

2 Block diagram

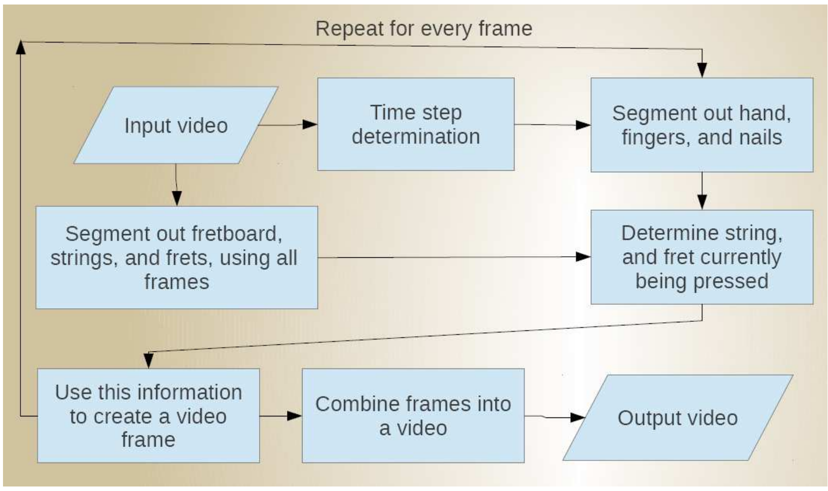
Given below is the block diagram for program flow. It describes how a video will be processed by our program. The following are important points:

1. **Input:** Our input will be a video of the guitarist playing his guitar.
2. **Output:** We shall output a video/audio/sheet music file, which will show/play/describe what the guitarist played throughout the video

3. **Time Step Determination:** In this step, we determine the frame rate at which we shall process the video. Right now, this is set to 4 frames per second. But this might change, as there's a possibility of choosing this dynamically.

4. **Segmentation tasks:** There are two segmentation tasks - segmentation of the fretboard, frets and strings; and segmentation of the hand/ finger/nails. The first task is to extract the fretboard from the image. The fretboard is the only area of our interest. The second task is to extract the finger pressing onto a string/fret. This finger shall determine which note was played, and hence, is important.

5. **Creation of video frame:** We will use the information extracted from the segmentation tasks, to determine which note is being played, and hence create a video frame that shows the appropriate fret pressed on the fretboard. Several such frames will be concatenated to produce a video.



Repeat for every frame
Input video
Time step determination
Segment out hand, fingers, and nails
Segment out fretboard, strings, and frets, using all frames
Determine string, and fret currently being pressed
Use this information to create a video frame
Combine frames into a video
Output video

Figure 1: The block diagram of program flow

3 Constraints

We have imposed several constraints to define the problem and approach:

1. The guitar should be held almost horizontally by the guitarist. A small tolerance of about ±5 degrees works.

2. The guitarist shall play no faster than 240 beats per minute - that gives us a frame rate of 4 frames per second.

3. The guitarist shall play a note with only one finger and keep the rest of the hand away from the fretboard.

4. The video focuses primarily on the guitar (although this doesn't seem very important, as we found from our interim segmentation results).

5. The guitarist plays the string that he presses, instead of only pressing it, and not actually playing it.

These constraints help us define our problem in a better way, and we can target specific areas, instead of finding a genralized approach, which will be a very tedious and time-consuming task.

4 Methods used

As of now, we used the following methods for segmentation:

1. **Taking Gaussian gradients for edge detection:** This is used primarily for finding the frets. As is mentioned in the constraints, we have assumed that the guitar is always held almost horizontally. Thus, frets will contribute to vertical edges, which can be found using Gaussian gradients.

2. **Pixel thresholding:** Thresholding is used primarily on the edge-detected image, because frets, being edges, will be brighter.

3. **Morphological opening:** To make the frets obtained using edge detection continuous - we don't want several patches of the same fret.

4. **Noise removal:** An operation remove stray pixels.

5. **Region labelling in binary images:** We know that frets do not touch each other. Hence, two regions that denote frets shall not touch each other. This operation can thus be used find the number of frets we have in the segmented image. This shall be used later in a brute force approach to find frets

6. **Convex hull construction using morphological opeartions:** Once we have a rough estimation of frets, we can use this to segment the fretboard - we find the end-points of these frets and take the convex hull of all these points. As frets have endpoints on edges of the fretboard, we will get the fretboard from the convex hull.

5 Issues and proposed methods to resolve them

The following are issues we are facing with the current techniques:

1. **Getting the complete fretboard:** This method is omitting the first fret because of pixel values. We plan on extending the current mask to automatically balance this error.

2. **Difficulty in removing unwanted edges:** Several unwanted edges in the edge-detected image are hampering our mask, as can be seen in the last set of images. We plan to overcome this using a brute force thresholding method, accompanied with a knowledge of the fretboard - the distance between two frets.

3. **Getting a good estimate of the mask:** This problem is mainly due to the one above, but can also be attributed to very thin frets being detected. In a situation where we get a smaller mask (than it should be), we shall performing repeated segmentation until we have 24 frets - which is the number of frets on the fretboard.

We plan on resolving these issues till the next deadline.

6 Interim Results

We include in this section, five sets of results from our segmentation tasks. We have included both - good and bad results. Bad results show that we have some refining to do in the methods. The first three sets of images are good results - they show the fretboard almost completely. The next set has a smaller fretboard, while the last one has extra regions.

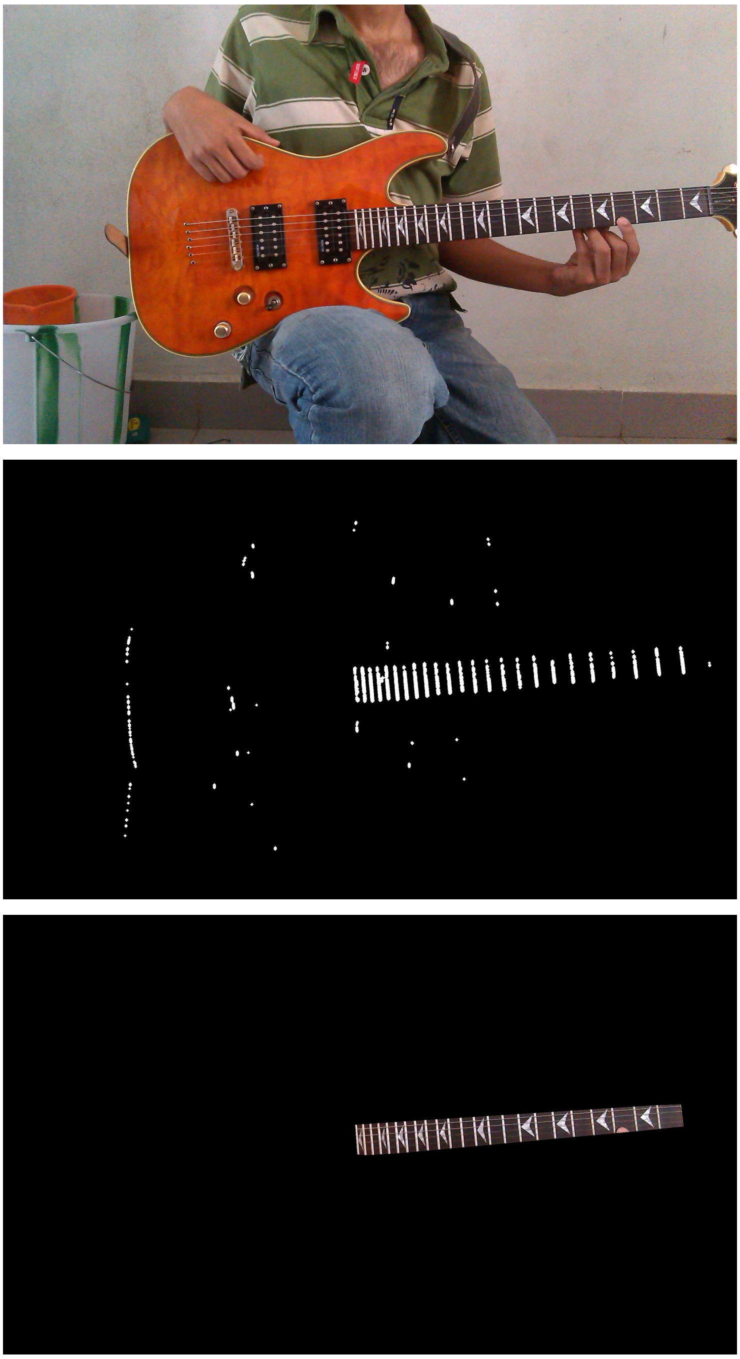


Figure 2: *From top to bottom:* The original image; the edge-detected image, after thresholding and morphological opening; the masked image showing only the fretboard obtained using the above methods

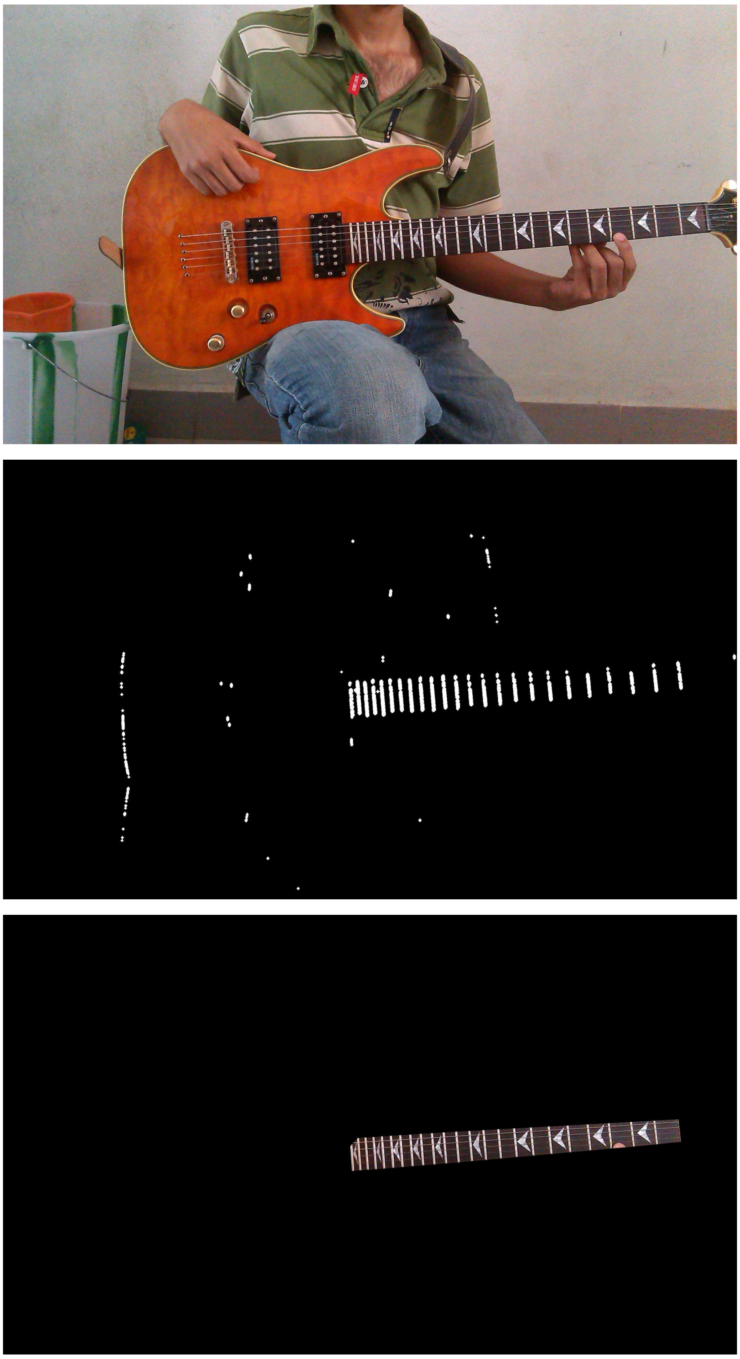


Figure 3: *From top to bottom:* The original image; the edge-detected image, after thresholding and morphological opening; the masked image showing only the fretboard obtained using the above methods

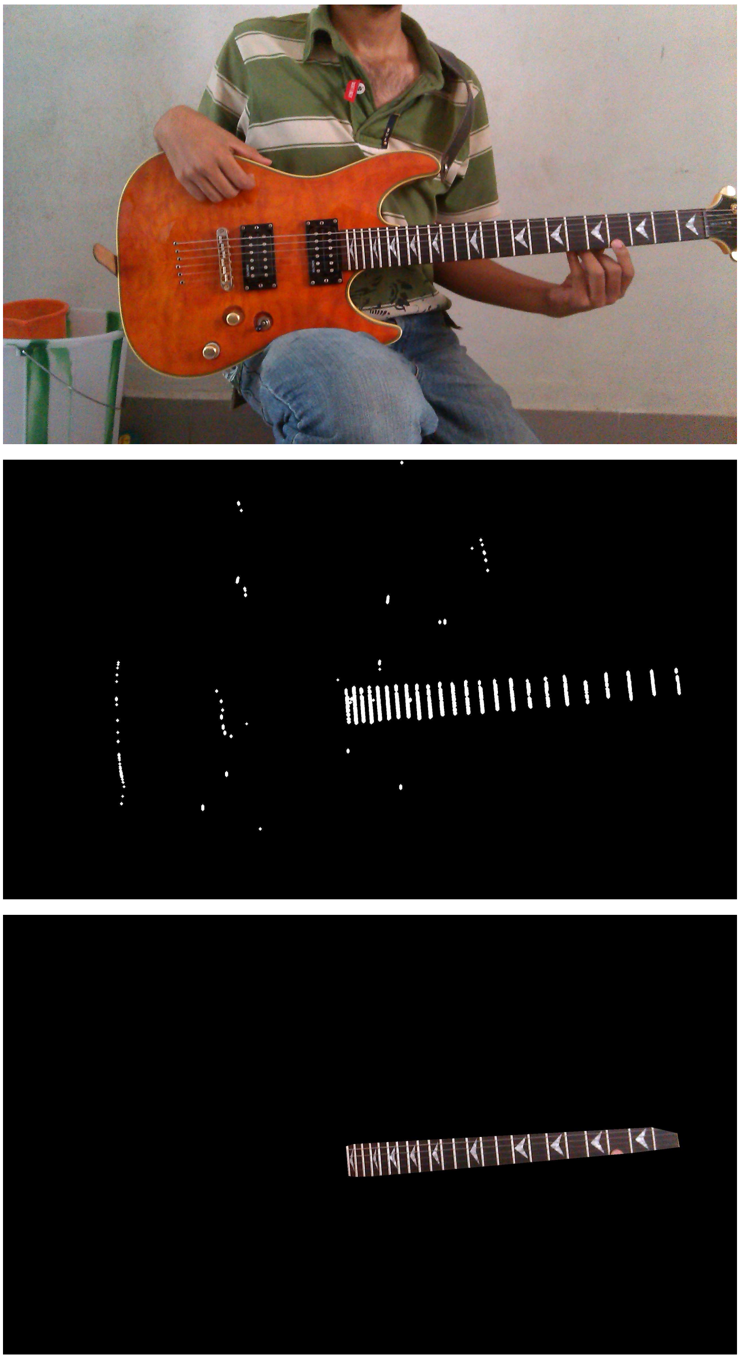


Figure 4: *From top to bottom:* The original image; the edge-detected image, after thresholding and morphological opening; the masked image showing only the fretboard obtained using the above methods

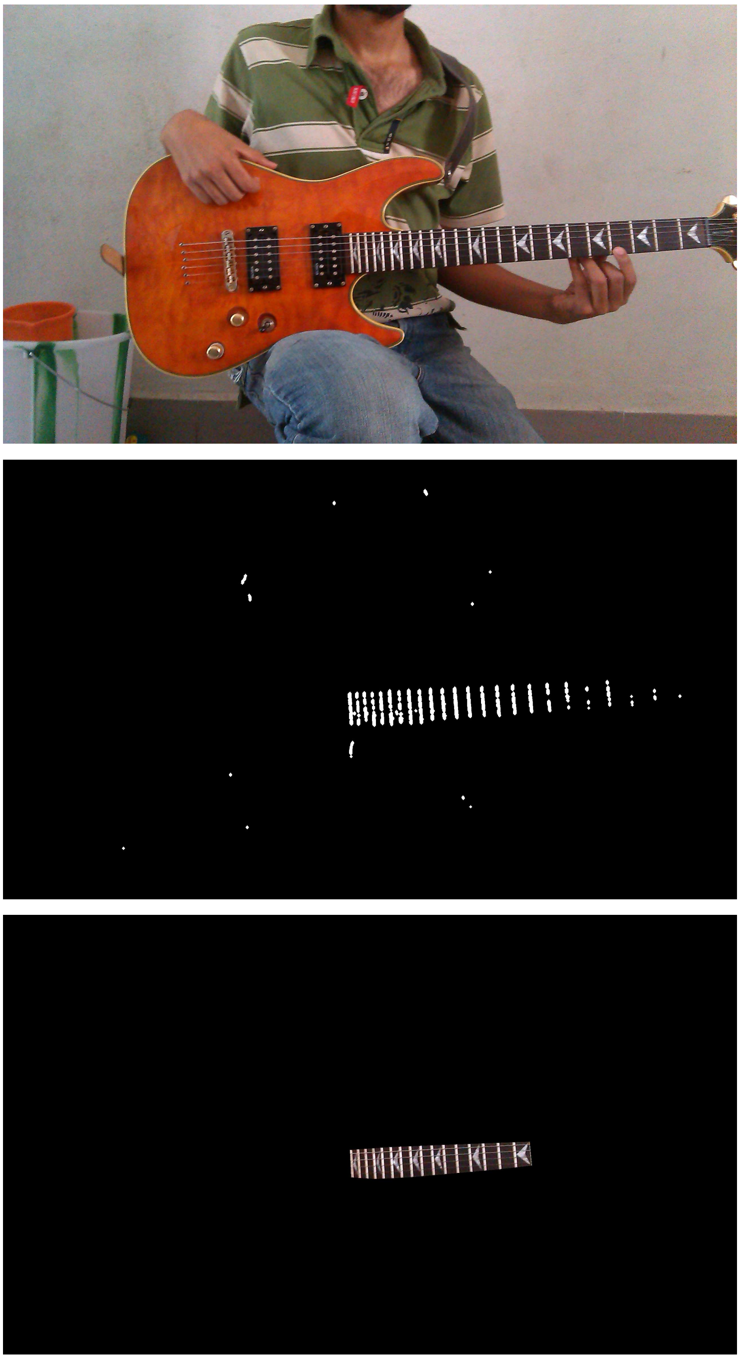


Figure 5: *From top to bottom:* The original image; the edge-detected image, after thresholding and morphological opening; the masked image showing only the fretboard obtained using the above methods. This mask has omitted regions belonging to the fretboard.

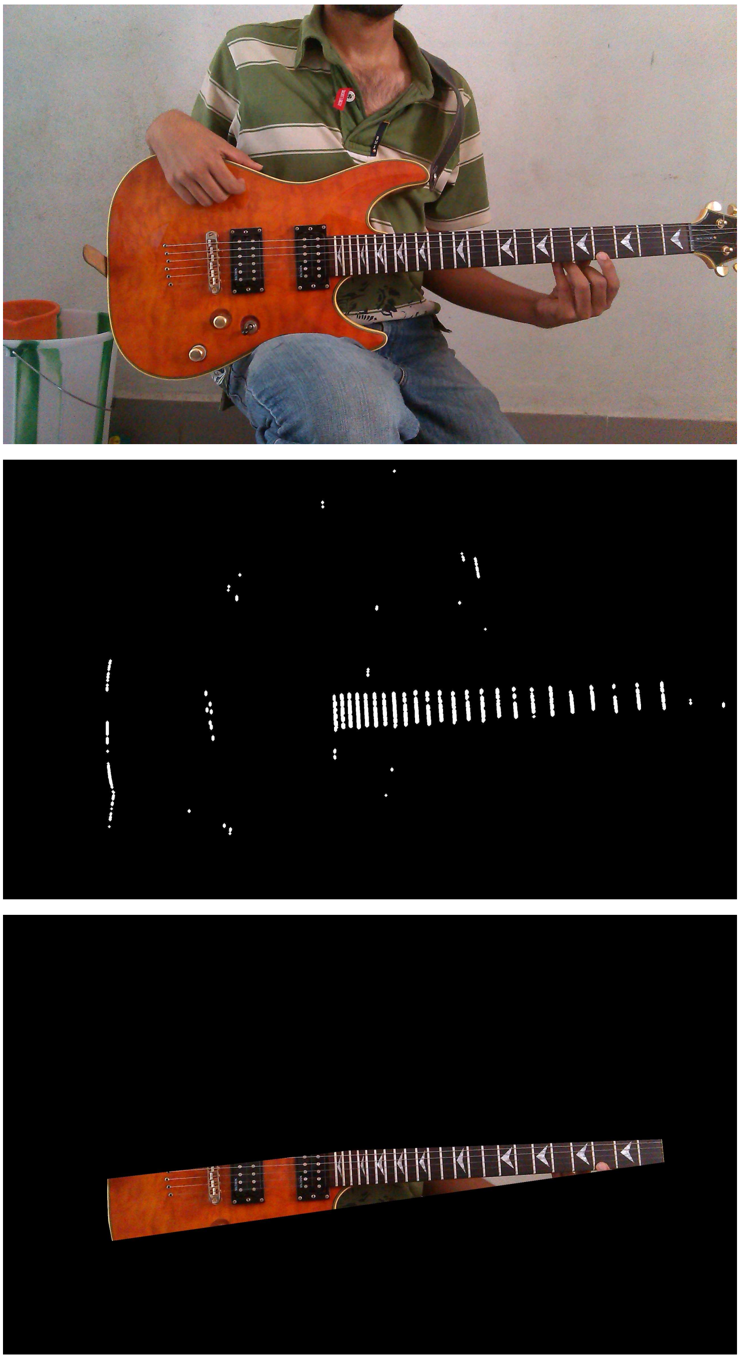


Figure 6: *From top to bottom:* The original image; the edge-detected image, after thresholding and morphological opening; the masked image showing only the fretboard obtained using the above methods. This mask has regions outside the fretboard too.