



# Guitar Tablature Generation Using Computer Vision

Brian Duke and Andrea Salgian<sup>(✉)</sup>

The College of New Jersey, Ewing, NJ 08628, USA  
{dukeb2, salgian}@tcnj.edu

**Abstract.** Traditionally, automatic music transcription uses audio recordings to generate the score of a musical piece. However, extracting pitch from an audio recording can be difficult. Vision-based approaches can address this by tracking the musician's gestures, and generating the score based on the way the musician plays the instrument.

In this paper we present a system that uses a vision-based approach to generate musical notation for guitar. Most guitar transcription systems to date rely on machine learning, specialized gear, or drawn-on markers. Our approach is vision based, yet markerless. We track the guitar's strings and frets, and in each frame we use skin detection to localize the guitarist's fingers on the fretboard. We use this information to generate tablature notation, a guitar specific notation that shows which strings and frets should be played for every beat.

The approach achieves a significantly higher accuracy than similar systems described in the literature. The system runs and displays tablature in real time, making it especially useful for educational purposes.

**Keywords:** Guitar fingering recognition · Guitar tablature · Automatic music transcription

## 1 Introduction

Writing musical notation can be a tedious process. A number of software applications are available to automatically transcribe audio files into staff notation that can then be read by musicians playing various instruments. Since audio processing is not an easy task, newer approaches have looked at vision-based methods, where musical notation is generated based on the gestures performed by the musician playing the instrument. This is especially important for the guitar, which is different from most instruments in that the same sound can be produced in multiple ways.

The tablature notation, described in more detail in Sect. 2, is a musical notation specific to the guitar, which shows which strings and frets should be played for every beat. This notation is difficult to extract from audio files. Verner [14] uses a midi guitar, with different midi channels associated to each string. Such guitars are expensive and not readily available to all musicians. Traube [13] uses

the timbre of the guitar, since two notes with the same pitch can have different timbre. The drawback of this method is the need for a-priori knowledge about the timbre of the guitar.

Existing vision-based transcription systems rely on markings or specialized gear that make guitar playing cumbersome. In [4], Burns and Wanderley describe a system where a webcam is mounted on the headstock of the guitar to get a stabilized, close-up view of the hand. The downside is that the mounted webcam can only capture the first five frets, and its weight disturbs the player. Kerdvibulvech’s system [7] used two webcams, colored fingertips, and an ARTag. Scarr and Green [11] used a markerless approach and showed promise during preliminary testing. However, the system had difficulties because of the lack of fretboard tracking and reliance on individual finger detection.

Multi-modal systems combine the analysis of audio and image data. An early multi-modal system described by Paleari et al. [10] relies mainly on computer vision and uses audio analysis to resolve ambiguous situations. However, the system was only tested on single notes.

In this paper we describe a vision-based automated music transcription system that generates tablature (or tab, for short) notation using a video recording of a guitar player. Using a simple webcam and computer vision techniques, our system detects the strings and frets of the guitar and the location of the guitarist’s fingertips, generating the tab notation with a higher accuracy than previously published systems. We show results on three different video recordings, for a total of 2298 frames.

## 2 Background






### 2.1 Guitar Tablature

Musical score for most instruments is recorded in staff notation as sheet music. Staff notation is an arrangement of notes of various durations played at various times, which amounts to pitches and their durations and attack times. This type of notation is suitable for instruments where there is a unique way to play each pitch, such as the piano.

Fretted instruments, such as the guitar, can produce the same pitch from different fingerings. For example, on a guitar in standard tuning, the fifth fret held down on the low E string produces the same pitch as the A string played openly. If the A note were represented in staff notation, a guitar player would have to decide which fingering to use. While a classically trained player may be able to sight-read staff notation and determine appropriate fretting positions on the fly, this is a time-consuming task for others. For this reason, amateur guitar players often prefer tablature.

A tablature shows the combination of frets and strings that should be played on a given beat. The six lines represent the six strings from the perspective of a player looking down at their guitar, with the low E string on the bottom and the high E string on the top. The numbers represent the fret position that should be held down [6].

Along with being easier to read, guitar “tabs” can be created without special software. Players type tabs using the ASCII character set and share them on websites such as [2]. Figure 1 shows the staff notation and tablature for a chord progression in C major.

	C	Am	Dm	G7	C
s					
T	0	0	1	1	0
A	1	1	0	0	1
B	2	2	2	2	2
3	3	0	0	3	3

**Fig. 1.** Staff notation and corresponding tablature for a chord progression in C major.

## 2.2 Recording Preparation

An ideal video recording for our system is one where the guitar is the main subject of the frame. To achieve this, a player can sit at a close distance in front of the camera, or the camera’s lens can be zoomed in. To make sure that the fingertips of the player are not obscured by the rest of their hand, the camera should be positioned at a slightly higher elevation than the guitar. Figure 2a shows an image obtained using a poor camera angle. Since the fingertips are not visible, the system would have to rely on the positioning of the knuckles, leading to many errors.



(a) An example of a poor camera angle.

(b) Correct setup.

**Fig. 2.** Setting up for a recording session

A setup that can provide contrast between the fretboard and background is recommended. The background should not be cluttered. Our videos were recorded in front of a solid white wall. To maintain the contrast of the scene, the guitarist should avoid wearing a patterned or striped shirt. The color of the fretboard of the guitar should not be the same as the player’s skin tone.

The color balance/white balance of the video should be set to ensure proper skin detection. This can be done within the camera’s settings, or using the Gray World algorithm. The latter would add overhead to the system’s runtime.

Videos were recorded by a camera mounted on a tripod and scaled down to have a width of 1280 pixels. No sensors or drawn-on markers were used.

Figure 2b shows the correct setup.

### 3 Methodology

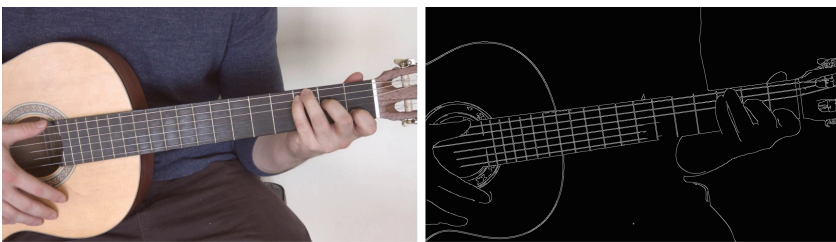
Our system starts by detecting and tracking the strings and the frets on the fretboard of the guitar. We then use skin color segmentation to detect the fingers, and we use their positioning relative to the strings and frets to obtain the tablature notation for the given frame.

We explain our algorithms in more detail below.

#### 3.1 Fretboard Detection

The first step of the algorithm is to find lines that represent the fret bars, strings, and outline of the guitar fretboard. The accuracy of these lines is crucial for the system’s performance. A classical guitar, such as the Yamaha guitar used in the test videos, allows for easy detections due to the thickness and contrast of the nylon strings. Strings on a steel-string acoustic guitar are harder to detect because they are thinner. Strings could also be rusted or missing.

We started by extracting the edges in each frame, using OpenCV’s Canny edge detector [5], and carefully selecting hysteresis threshold values to minimize the amount of extraneous edges. Figure 3 shows a frame and the results of the Canny edge detection.



(a) Frame from a test video.

(b) Edge detection results.

**Fig. 3.** Original frame and Canny edge detector results.

We then use the Progressive Probabilistic Hough Transform (PPHT) algorithm to extract straight lines. Just like for the Canny edge detection, thresholds were determined experimentally, to maximize accuracy. Lines are extrapolated to fill in gaps and to extend them to edges of the fretboard. Lines that are very

close and have similar slopes are averaged, as they usually represent multiple detections of the same string or fret bar.

The positions of strings can be predicted using knowledge of the top and bottom of the fretboard wood, using the formulas below:

$$\frac{\text{Outer gap}}{\text{Fretboard height}} = 0.086 \quad (1)$$

$$\frac{\text{Inner gap}}{\text{Fretboard height}} = 0.17 \quad (2)$$

The outer gap is the distance between the top or bottom string and the top or bottom of the board. The inner gap is the distance between two adjacent strings. Equation (1) can be used to predict the positions of the first and sixth string, while Eq. (2) can predict the position of an inner string.

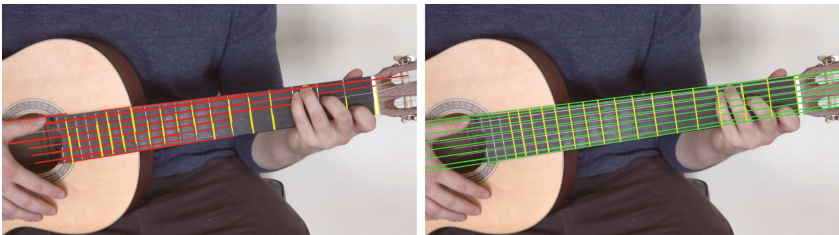
While these predictions are not accurate enough to replace line detection, they can be used to remove duplicate detections, or to signal undetected strings.

Fret bar lines can also be predicted using a luthier's formula [1], but they will be even less accurate due to camera lens distortion or the guitar not being flat against the lens.

To address the problem of occasionally undetected strings or fret bars, we use tracking. We start by picking a calibration frame at the beginning of the video sequence, before the player places their hand over the fretboard. Lines from this frame are used to initialize the tracking values. Future detections are compared to the previously tracked values. If the string lines'  $y$  coordinates or the fret lines'  $x$  coordinates fall within a close pixel distance of the tracked values, the tracked values can be updated with the current frame's detections. Otherwise, lines were probably not properly detected, and the tracked values from the previous frame should be used in the current frame.

This process allows the system to perform well on frames where the hand obscures the fret bars and prevents them from being detected.

Figure 4a shows the lines detected using PPHT, while Fig. 4b shows the strings and frets detected after line correction.



(a) PPHT detected line segments.

(b) Detected strings and frets.

**Fig. 4.** Fretboard detection.

Finally, the angle of the fourth (middle) string is used to rotate the frame of the video so that strings are horizontal, and fret bar lines are vertical. Once the fingertips are detected, this positioning will allow us to determine their correct location on the fretboard with less calculation.

### 3.2 Finger Detection and Localization

The second objective of the algorithm is to detect the guitar player's fingers, and determine where the fingertips are in relation to the guitar's strings and the frets.

Skin pixels are detected using their RGB values, according to the Kovac model [8]. A pixel is considered skin if its RGB values follow these rules:

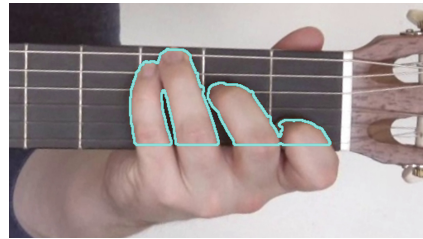
$$\begin{aligned} R &\geq 95 \\ B &\geq 20 \\ G &\geq 40 \end{aligned} \tag{3}$$

The procedure is followed by a Gaussian blur to remove noise. Finally, pixels with R and G values that fall within a distance of 15 from each other are eliminated.

Results are shown in Fig. 5. The body and the headstock of the guitar were classified as skin. Since we are only interested in the fingers placed on the fretboard, and the fretboard has already been detected, we remove other parts of the image from consideration. We then use Suzuki's border following algorithm [12], as implemented in OpenCV, to find the contour of the fingers. Figure 6 shows the results of this step.

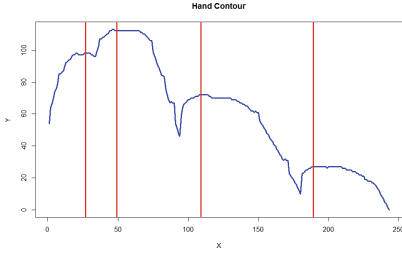


**Fig. 5.** Skin detection and segmentation.

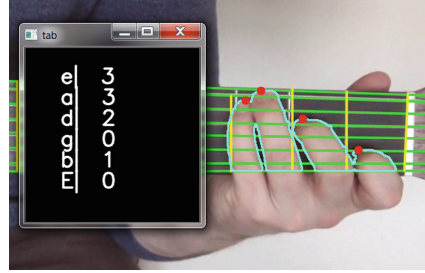


**Fig. 6.** Finger isolation and contour detection.

Next, we keep only the top half of the contour, and detect the fingertips at the local maxima of the  $y$  coordinate. Figure 7 shows the process. This approach is more robust than other approaches which require detection of each finger, and can fail when fingers are touching [11]. Then, we produce a tablature by analyzing where the points are in relation to the strings and frets. The system output is shown in Fig. 8.



**Fig. 7.** Fingers contour with fingertips located at local maxima.



**Fig. 8.** System output. Tablature notation was flipped to correspond with strings in the video.

## 4 Results

We tested our approach on three video recordings: a chord progression in C major whose notes are shown in Fig. 1 (659 frames), an open C major scale followed by an E major scale that begins on the seventh fret (533 frames), and a chord progression from “Autumn Leaves”, a jazz standard popular with beginner guitar players (1106 frames). The videos were manually transcribed by a volunteer who, for each frame, recorded what appeared to be the current fretting position in tablature notation. The volunteer also noted which frets had a finger hovering over them.

We then compared the manually transcribed tab to the system’s output. Correct detections were those where the fret for a string matched the one listed in the manually transcribed tab. Hovering notes were also checked for correctness. Open notes, where there are no fingers on a given string, marked as 0 in tab notation, were not considered in the performance calculation, as they would have unfairly inflated performance numbers.

Tables 1, 2 and 3 show the performance of the system on each of these videos, separated by strings. Since the tab output lists the fret for each string, we reported separate performances for each string, as well as an overall average performance. Since open notes are not considered, the number of frames compared varies from string to string.

While testing on “Autumn Leaves” is unique to our paper, other systems were tested on the chord progression in C major and the C major scale. Table 4 shows a comparison between our results and results published in [11] and [3].

Our system’s best performance was on the C major chord progression, where the overall accuracy was 86%. This is significantly higher than the performance of Scarr’s proposed system [11], which achieved a 52% accuracy on the same chord progression, or Burns’ system [3], with an accuracy of 14%.

“Autumn Leaves” had the worst performance at 71% (Table 3). This is mainly due to the fact that in several chords the little finger was under the ring finger, and our approach failed to detect its tip (see Fig. 9). In addition, hovering fingers are ambiguous, as it is hard to tell whether the finger is pressing down on a string

**Table 1.** Chord progression in C major. 659 frames.

	String					
	1	2	3	4	5	6
Errors	2	126	28	40	131	12
Frames compared	383	384	382	275	622	302
Accuracy	99%	67%	93%	85%	79%	96%
Overall accuracy	86%					

**Table 2.** Open C major scale followed by an E major scale that begins on the seventh fret. 533 frames.

	String					
	1	2	3	4	5	6
Errors	–	7	30	56	103	–
Frames compared	–	102	188	183	330	–
Accuracy	–	93%	84%	69%	69%	–
Overall accuracy	76%					

**Table 3.** “Autumn Leaves”. 1106 frames.

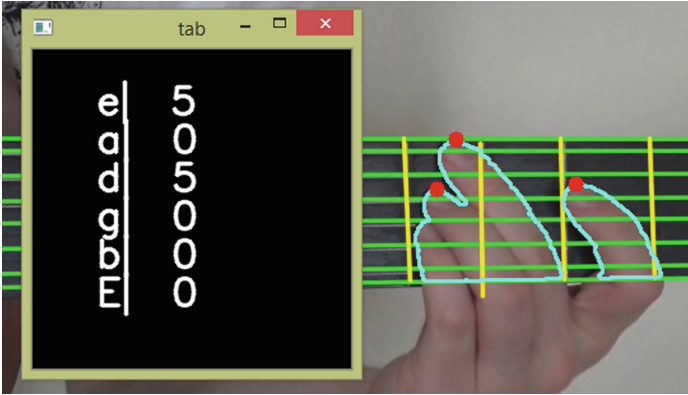
	String					
	1	2	3	4	5	6
Errors	81	188	283	318	–	–
Frames compared	571	574	962	857	–	–
Accuracy	85%	67%	71%	63%	–	–
Overall accuracy	71%					

**Table 4.** Comparison between our results and previous methods

Video	Burns’ method	Scarr’s method	Our method
C major scale	34%	78%	76%
C major chords	14%	52%	86%
Autumn Leaves	N/A	N/A	71%



or not. This problem is usually solved by assuming that the highest detected fret on a string is the one being played. However, that assumption can backfire when a scale is played.



**Fig. 9.** Missed fifth finger detection on the first chord of “Autumn Leaves.”

As our initial goal was to assess transcription accuracy, we did not put too much effort into optimization. Nevertheless, our system runs in real-time. Processing time can be further reduced by cutting down the number of frames that are analyzed, as described below.

When playing a chord, a guitarist’s fretting hand remains mostly still, while their strumming hand moves. In the current version of the system, tab output is calculated at every frame, even as it stays constant for the duration of a note. This results in unnecessary overhead, as well as an occasional jitter in the output due to a detection glitch. To address this problem, we are currently working on using key frames, i.e. frames where the guitarist’s fretting hand moves significantly relative to the previous frame. These can be detected by calculating pixel differences in consecutive frames, as described by Wang and Ohya [15].

## 5 Conclusion and Future Work

In this paper we presented a real-time, computer vision-based, markerless system for transcribing guitar music. Our approach starts by detecting and tracking the strings and frets of the guitar. It then uses skin detection and contour following to extract and localize the fingertips on the guitar’s fretboard.

We tested the system on three recordings of well-known guitar pieces, and achieved accuracies of 86%, 76%, and 71% on the C major chord progression, C major and E major scales, and the “Autumn Leaves” piece respectively. In particular, the 86% performance on the C major chord progression is significantly higher than that achieved by other systems. This can be attributed mostly to

the fact that our system does not rely on detecting individual fingers. Instead, it localizes fingertips based on local maxima of the hand contour.

Tracking the guitar strings and frets from one frame to the next, rather than relying solely on detection in individual frames, also improves performance.

Another advantage of our system is that it can display the tablature output in real time, making it especially useful for educational purposes.

Future work includes implementing key frames as described in the previous section, as well as improvements to tracking and skin detection. Using optical flow based tracking such as the Lucas-Kanade [9] algorithm can make the system more robust to unexpected, fast moves by the guitarist. Skin detection could be improved to handle more varied skin tones.

Finally, better results could be obtained by analyzing the other hand of the guitarist to determine if strings are picked or strummed, and by using a multi-model approach that combines audio and video data.

**Acknowledgment.** The authors would like to thank Matthew Van Soelen for manually annotating the video files and helping with the error analysis.

## References

1. Calculating fret positions. <https://www.liutaiomottola.com/formulae/fret.htm>. Accessed 10 Jun 2019
2. Ultimate guitar tabs - 1,100,000 songs catalog. <https://www.ultimate-guitar.com>. Accessed 10 Jun 2019
3. Burns, A.M.: Computer vision methods for guitarist left-hand fingering recognition (2007)
4. Burns, A.M., Wanderley, M.M.: Visual methods for the retrieval of guitarist fingering. In: Proceedings of the 2006 Conference on New Interfaces for Musical Expression, pp. 196–199. IRCAM-Centre Pompidou (2006)
5. Canny, J.: A computational approach to edge detection. In: Readings in Computer Vision, pp. 184–203. Elsevier (1987)
6. Harrison, E.: Challenges facing guitar education. *Music Educators J.* **97**(1), 50–55 (2010)
7. Kerdivulvech, C., Saito, H.: Real-time guitar chord estimation by stereo cameras for supporting guitarists. In: Proceeding of 10th International Workshop on Advanced Image Technology (IWAIT 2007), pp. 256–261 (2007)
8. Kovac, J., Peer, P., Solina, F.: Human skin color clustering for face detection, vol. 2. *IEEE* (2003)
9. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of Image Understanding Workshop, pp. 121–130. *IEEE* (1981)
10. Paleari, M., Huet, B., Schutz, A., Slock, D.: A multimodal approach to music transcription. In: 2008 15th IEEE International Conference on Image Processing, pp. 93–96. *IEEE* (2008)
11. Scarr, J., Green, R.: Retrieval of guitarist fingering information using computer vision. In: 2010 25th International Conference of Image and Vision Computing New Zealand, pp. 1–7. *IEEE* (2010)

12. Suzuki, S., et al.: Topological structural analysis of digitized binary images by border following. *Comput. Vis. Graph. Image Process.* **30**(1), 32–46 (1985)
13. Traube, C.: *An Interdisciplinary Study of the Timbre of the Classical Guitar*. Ph.D. thesis, McGill University (2004)
14. Verner, J.A.: Midi guitar synthesis: yesterday, today and tomorrow. *Recording Mag.* **8**(9), 52–57 (1995)
15. Wang, Z., Ohya, J.: Tracking the guitarist's fingers as well as recognizing pressed chords from a video sequence. *Electron. Imaging* **2016**(15), 1–6 (2016)