An Accurate and Robust Algorithm for Tracking Guitar Neck in 3D Based on Modified RANSAC Homography

Zhao WANG and Jun OHYA
Department of Modern Mechanical Engineering, Waseda University, Tokyo, Japan

Abstract

Towards the actualization of an automatic guitar teaching system that can supervise guitar players, this paper proposes an algorithm for accurately and robustly tracking the 3D position of the fretboard from the video of guitar plays. First, we detect the SIFT features within the guitar fretboard and then match the detected points using KD-tree searching based matching algorithm frame by frame to track the whole fretboard. However, during the guitar plays, due to movements of the guitar neck or occlusions caused by guitar players' fingers, the feature points on the fretboard cannot always be matched accurately even though applying traditional RANSAC homography. Therefore, by using our modified RANSAC algorithm to filter out the matching error of the feature points, perspective transformation matrix is obtained between the correctly matched feature points detected at the first and other frames. Consequently, the guitar neck is tracked correctly based on the perspective transformation matrix. Experiments show promising results such as high accuracy: the total mean tracking error of only 4.17 mm and variance of 1.5 for the four tracked corners of the fretboard. This indicates the proposed method outperforms related tracking works including state-of-art Fully-convolutional Network

Introduction

Recently, with the high-speed growth of the computer vision, achieving automatic guitar teaching systems that can supervise guitar players' fingering practices has been attracting in academic research communities [1~10]. One of the most fundamental functions required for such a system would be automatic tracking of guitar necks while playing guitars: i.e. the automatic guitar neck location at each frame of the videos containing the guitar plays is the first process prior to the subsequent processes such as recognizing or assessing the fingerings. Similar to other object tracking problems, main challenges of the guitar neck tracking include (1) changes in the appearances of the guitar necks as the guitar players could shake or swing the guitars, (2) changes in illumination, (3) guitar necks' occlusions caused by the guitar players' fingers.

Conventional tracking methods are difficult to be applied to the guitar neck tracking, because guitar players tend to move their hands fast while playing guitars, which could result in occluding the guitar necks partially or entirely at almost every frame. Also, the determination of the features on the guitar neck is an issue to be solved, because the above-mentioned occlusion could happen at any place on the neck due to fast movements of the guitar player's hand.

Conventional works related to guitar neck tracing [4~9] own novelties but also problems: Y. Motokawa and H. Saito [4] and C. Kerdvibulvech and Hideo Saito [7,8] attach an AR (augmented reality) tag to track the guitar neck; A. Burns [6] fixes a camera to the guitar neck so that the guitar and the camera are relatively static to each other. However, these tool-attached approaches (a) bring a lot of inconveniences to guitar players and (b) cannot track the guitar

neck if the viewpoint or scale changes. Joseph Scarr and Richard Green [5] use Canny edge detector and Hough Transform to detect the neck area. However, when the finger of the player and the neck fretboard overlap while playing, it is impossible to track the guitar neck accurately. Zhao W. and Ohya J. [9, 10] propose an algorithm that tracks the guitar neck using a recovery-from-overlap approach: it detects and tracks five feature points, and by calculating the distance and angle of each three points, it estimates whether and which feature points and fingers are overlapped. Then it recovers the overlapped feature points and calculate the homography based on the recovered feature points so as to track the guitar neck. Although it handles the overlap by recovering feature points, it still cannot recover tracking failures; also, it takes nearly 20 seconds per frame to calculate the distances and angles of each three feature points: i.e. the computation efficiency is very low.

This paper proposes an accurate and robust guitar neck tracking algorithm to solve the problems mentioned before. Specifically, (1) SIFT feature points are to be detected on every frame as it is invariant to rotation, illumination and scale changes in images; (2) we propose a KD-tree searching based algorithm to match the SIFT features between the first frame and any other frame of input videos; (3) we propose a modified version of RANSAC (Random Sample Consensus) to overcome the above-mentioned occlusion issue. As mentioned earlier, feature points within the guitar neck area cannot be accurately tracked or matched, because it is overlapped and occluded by fingers of guitar players. The proposed modified RANSAC-based filtering algorithm filters out and eliminates the mismatched feature points, and then calculates the homography between the correctly matched feature points on the first frame and on the any other frame to track the guitar neck. Besides, since the homography is calculated between the first frame and any other frame, our methods does not need to concern about the tracking failure problem. (4) to suppress the effect of the guitar neck motion, the tracked guitar neck area on every frame is projected to a new image sequence based on the calculated homography frame by frame. Owing to this projection, no matter how the guitar player shakes or swings the guitar neck while playing, the neck area on every frame is always projected to the center of the new image sequence to facilitate analyzing the fingering, where this analysis is our future work.

In the remainder of this paper, Section 2 outlines the proposed method. Our approach is detailed in Section 3. Then, we evaluate our work by doing self-comparison and comparisons with related works in Section 4. Finally, we conclude this paper and plan our
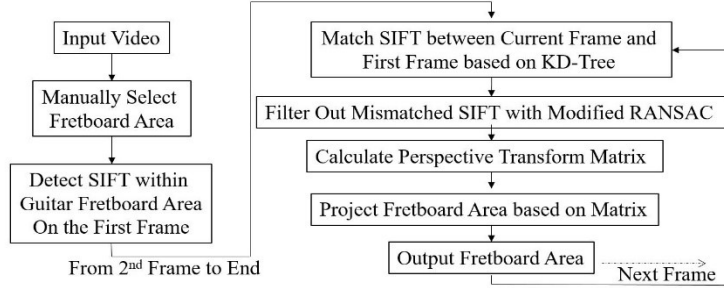
Figure 1: Overflow of our Approach

future work in Section 5. our work by doing self-comparison and comparisons with related works in Section 4. Finally, we conclude this paper and plan our future work in Section 5.

Outline

As depicted in Fig.1, after we input the video of guitar playing, first we manually select the guitar neck area by selecting the four corners of fretboard on the first frame and detect SIFT feature points within that fretboard area. Then, from second frame, we match the SIFT features detected on the current frame with the SIFT features by using a KD-Tree based searching method to accelerate matching efficiency. Furthermore, we filter out the mismatched SIFT feature point pairs between the first frame and the current frame by implementing a RANSAC mechanism. In addition, we obtain the perspective transform matrix based on correctly matched SIFT pairs. Finally, we project the fretboard area based on the obtained matrix and the manually selected fretboard area on the first frame to output the guitar neck area.

Guitar Neck Tracking

Input

In our algorithm, to track our guitar neck in 3D dimensional space, we use Kinect color image and depth image as our input. Note that, as the depth is only used to measure 3rd dimensional distance, our algorithm could also be applied in 2D color image.

Detecting and Matching SIFT Features

Scale Invariant Feature Transform (SIFT), which was proposed by Lowe [11], is proved an efficient algorithm for object or scene recognition. After inputting a color image sequence of Kinect, we detect feature points of SIFT within the guitar fretboard area, which is defined by manually specified four corners at the first frame of the input video. Some of the SIFT feature points may fall outside of neck area in our work, because we boarden our mannully selected area in order to detect the features near the border as many as possible, and the outside features can be eliminated in our next step. From the second to final frame of the video, we detect SIFT feature points at each frame and match the features with the features detected at the first frame. For speeding up the matching process, we apply a KD-Tree based searching algorithm [12,13]. The KD-tree is a binary tree in which every node is a k-dimensional point. Every non-leaf node generates a splitting hyperplane that divides the space into two subspaces. Points left/right to the hyperplane represent the left/right sub-tree of that node. The hyperplane direction is chosen in the following way: every node split to sub-trees is associated with one of the k-dimensions, such that the hyperplane is perpendicular to that dimension vector [13].

Filtering out Mismatched Features Based on Modified RANSAC and Calculating the homography Matrix

RANSAC algorithm can be used to remove the mismatches by finding the transformation homography matrix of these feature points. In our case, the homography matrix is show in Eq(1) and Eq(2):

$$X_i = HX_1 \ , i \in (1,2,3 \dots I) \qquad (1)$$

$$\begin{pmatrix} x_i \\ y_i \\ \omega \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}, i \in (1,2,3 \dots I) \qquad (2)$$

where: $(x_i, y_i, \omega)T$: homographic coordinates of the SIFT feature $(x,y)T$ at the current frame (Frame i); $(x_1, y_1, 1)T$: homographic coordinates of the SIFT feature $(x,y)T$ at the first frame; I is the frame number. H is the homography matrix with 8 parameters $(a,b \dots h)$.

However, in our case since the area in the guitar neck appears nearly same (fret and string), the SIFT feature points in the fret board area tend to share nearly same scales and directions. If the traditional RANSAC is applied to the SIFT features, it is difficult to calculate the homography matrix H in Eq(1) due to too many wrong matches (called outliers), whose examples are shown in the left of Fig.2a. In other words, the traditional RANSAC could not filter out the mismatched feature points, and the correct matches of SIFT (called in inliers) could not be found either, as shown in the middle of the Fig.2a. Figure 2.b shows two cases that the traditional RANSAC is hard to be applied due to guitar neck swing and fast movement of the guitar player's hand.

To solve the above-mentioned traditional RANSAC's problem, this paper proposes a modified RANSAC as follows:

a. The detected SIFT features are matched between the first and other (second or later) frame as mentioned at Section 3.2. As the left of the Fig.2a shows, the matched features are connected between the two frames by line segments (pink line segments in Fig.2) The line segment (called "matching line") is defined in the new two dimensional coordination system stacked with two images vertically. The matching lines between the two frames are represented by:

$$\widehat{L_i} = (l_{1,i}, l_{2,i} \dots l_{N,i}) \qquad (3)$$

where, $\widehat{L_i}$ is the matched line vector on *Frame i*, *N* is the number of matched lines.

b. In $\widehat{L_i}$, matching lines that cross a large number of other lines are removed. Here, if a member $l_{n,i}$ in $\widehat{L_i}$ cross with 70% of the other lines in $\widehat{L_i}$, we eliminate $l_{n,i}$. We loop all the lines in $\widehat{L_i}$ and the remaining matching lines are defined in Eq. (4):

$$L_i = (l_{1,i}, l_{2,i} \dots l_{M,i}), M < N \qquad (4)$$

Figure 2a: Left: SIFT Matching Result; Middle: Traditonal RANSAC (cannot calculate Homography due to too many outliers); Right: Filtered Matching Result Based on Our Modified RANSAC

Figure 2b: Left two: Result of RANSAC Homography under the guitar neck swing condition; Right two: Result of RANSAC Homography under the fast movement of guitarist's hand
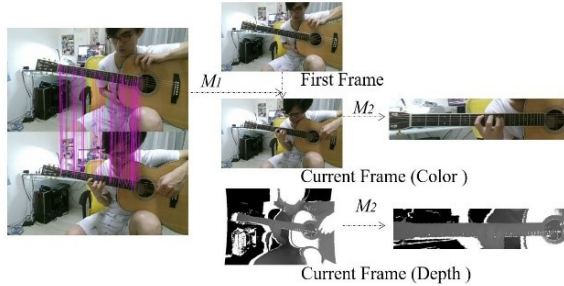

Figure 3: Overflow of Our Approach

where, L_i is called the remaining matching vector, and M is the number of the remaining lines, where the matching line ID m of l_(m,i) in Eq. (4) is rearranged from that of Eq. (3).

c. Based on the remaining matching vector L_i, we apply RANSAC and calculate the homography matrix between the first frame and the current frame.

$$\begin{pmatrix} x_{i,m} \\ y_{i,m} \\ \omega \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x_{1,m} \\ y_{1,m} \\ 1 \end{pmatrix}, i \in (1,2,3\ldots I), m \in$$
$$(1,2..M) \qquad (5)$$

where: $(x_{i,m}, y_{i,m}, \omega)^T$: homographic coordinates of the mth SIFT feature $(x,y)^T$ at the current frame (Frame i); $(x_{1,m}, y_{1,m}, 1)^T$: homographic coordinates of the mth SIFT feature $(x,y)^T$ at the first frame; I is the frame number. M is the number of the removing result of SIFT match in Eq(4). The result of the inliers of our modified RANSAC are shown at the right of Fig.2.

d. Go to the next frame, do from a. to d.

Projecting Fretboard Area

In Section 3.3, the homography matrix for the inliers (3*3 dimensional matrix in Eq.(5)) is the perspective transform matrix of the guitar fretboard area between the first frame and *Frame i* in Eq.(5). Given the coordinates of the four corners of the guitar fretboard at the first frame, we calculate the coordinates of the four corners of the fretboard at frame i based on the homography matrix as follows:

$$\begin{pmatrix} x_{i,c} \\ y_{i,c} \\ \omega \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x_{1,c} \\ y_{2,c} \\ 1 \end{pmatrix},$$
$$i \in (1,2,3\ldots I), c \in (1,2..C = 4) \qquad (6)$$

where: $(x_{i.c}, y_{i,c}, \omega)T$: homographic coordinates of the cth corner of guitar neck at the current frame (Frame i); $(x_{1,c}, y_{1,c}, 1)T$: homographic coordinates of the cth corner of guitar neck at the first frame: Frame 1; I is the frame number; C is corner number of guitar neck equals to 4.

After the four corners of the guitar neck are tracked in Eq.(6), we project the tracked guitar neck area on every frame to a new image sequences, which the guitar neck area is always at the center of the image sequence. The whole projecting process is presented as:

$$X_{i'} = M_2 X_i \;, i \in (1,2,3\ldots I) \qquad (7)$$
$$\begin{pmatrix} x_{i,c'} \\ y_{i,c'} \\ \omega \end{pmatrix} = \begin{pmatrix} i & j & k \\ l & m & n \\ o & p & 1 \end{pmatrix} \begin{pmatrix} x_{i,c} \\ y_{i,c} \\ 1 \end{pmatrix}, i \in (1,2,3\ldots I), c$$
$$\in (1,2..C = 4) \qquad (8)$$

where M_2 is the projecting matrix with 8 parameters *(i,j...p)*; X_i $=(x_{i,c}, y_{i,c})^T$ is the tracked cth corner of guitar neck at the current frame (*Frame i*) in Eq.(6), $X_{i'} =(x_{i,c'}, y_{i,c'})^T$ is position of cth corner

Table 1: Accuracy of Self-comparison (u,v are the two dimensional space of image, d is the depth)

	Mean Error of Left-up Corner			Mean Error of Left-bottom Corner			Mean Error of Right-up Corner			Mean Error of Right-bottom Corner			Mean	Variance
	u	v	d	u	v	d	u	v	d	u	v	d		
Our Work (SIFT + Modified RANSAC) (mm)	2.3	2.4	5.6	3.3	2.3	6.0	2.7	2.8	6.3	4.4	4.5	7.4	4.17	1.5
SURF + Modified RANSAC (mm)	2.5	2.0	5.8	2.8	2.7	7.3	3.3	3.1	6.3	5.6	5.7	8.0	4.59	1.7
SIFT+RANSAC (mm)	5.3	5.7	8.9	4.4	3.9	9.2	5.5	5.6	10	6.4	7.0	11	6.90	3.44
SURF+RANSAC(mm)	6.2	6.6	9.0	6.4	6.4	9.8	4.7	4.3	7	8.8	7.9	13	7.51	3.98

of guitar neck in the new image sequence. As we mentioned before, we want to always project guitar neck area to the fixed center of a new image sequence no matter how guitarist swing the guitar during playing. Given the X_i and $X_{i\prime}$ (X_i is the tracked corner of guitar neck, $X_{i\prime}$ is the fixed position of the corner in new image sequence), we can easily calculate M_2, and project the color image and the depth image to the new image sequence as shown in Fig.3.

Evaluation

We conduct evaluation experiment using the dataset we create by ourselves. The dataset includes 50 videos of guitar playing with nearly 3000 frames of the color images (also 3000 frames of depth) taken by Microsoft Kinect. The whole dataset includes 3 kinds of music pieces, which are the most frequently daily practices for guitarist: (1) *C* major on first fret (2) *a* minor scale and (3) symmetrical excise as all of them are fundamental, classic practices but best way to improve dexterity, speed, strength and stamina to help you overcome obstacles and become a better guitar player [14]. All the data are taken under different illumination situation (day light, incandescent lights etc.) and complex background. All the videos of guitar playing are taken by three different guitars to show the generality usage of our algorithm.

For the experiments, the system we used is a windows 10 desktop with a 3.0 GHz Intel Core i7 processor and DDR3 16GB memory without GPU acceleration. The camera is Microsoft Kinect taking color image sequence and depth image sequence with the same resolution 1200"800. All the algorithms are implemented in Visual Studio 2013 with C++ and OpenCV 2.4.10 library.

Self-comparison

 We self-compare our work by (1) comparing different two kinds of feature points including SIFT, SURF combined with our proposed method (SIFT + Modified RANSAC, SURF + Modified RANSAC) and traditional RANSAC (SIFT + Traditional RANSAN, SURF + Traditional RANSAC) respectively to test the effectiveness of our method; (2) in each combination such as SIFT + Modified RANSAC, we calculate 3D (u,v,d in Table.1) tracking error to measure the 3D tracking error respectively for each corner of guitar neck; (3) we also calculate the total mean tracking error and variance of each combination of all four corners for the whole dataset in Euclid distance. From Table.1, we find out that our proposed method (SIFT + Modified RANSAC) outperforms other combinations (SURF + Modified RANSAC, SIFT + Traditional RANSAN, SURF + Traditional RANSAC) with total mean error of 4.17 mm, variance of 1.5 mm. Also, our modified RANSAC either

combined with SIFT or SURF (SIFT + Modified RANSAC, SURF + Modified RANSAC) highly outperform the traditional methods (SIFT + Traditional RANSAC, SURF + Traditional RANSAC), which indicates our method is more efficient than traditional RANSAC in the guitar neck 3D tracking case while combining with SIFT or SURF. Other detail of selfcomparison could be found at Table.1. We also compare the time efficiency for these methods in Table.2. Traditional RANSAC combined with SIFT meanly runs at 2.3 sec per frame while accelerating with KD-Tree searching. Our modified RANSAC only needs extra 0.2m to filter out the mismatch, which means processes a frame with 2.5s (0.4 FPS) An example of tracking result on 3D image sequence is shown in Fig.4.

Comparison with Related Works

We compare our methods against the related works [10, 5] of guitar tracking algorithm those do not need supporting tools such as AR tag or fixed cameras on our dataset. Besides, we also compare our work with state-of-art deep learning method, which is based on Fully-convolutional network [15]. We experimentally implemented all the works. For [5], it is easy to detect lines by Hough Transform and remain the biggest cluster of lines that have the same slope; for [10], as the paper said, we apply Optical Flow to track the 40 detected Shi-Tomashi Features (as [10] writes 40 points is the best performance); for [15], we implemented a 7-level fully convolutional network, that is identical to VGG16, except replacing the last 3 fully-connected to 3 convolutionals. Fig.5 shows the comparison result. We apply a general comparison method that is widely used in recent tracking researches. The horizontal axis indicates the thresholds of mean error of tracking, while the vertical one means the precision of tracking when each threshold on the horizontal axis is set. Our work outperforms others by achieving 100% when the threshold is set to 8mm. Table. 3 gives a numerical comparison of time efficiency and mean tracking error with the works mentioned before. From Table.3, our work (0.4 FPS) is much

Table 2: Self-comparison of Time Effiency

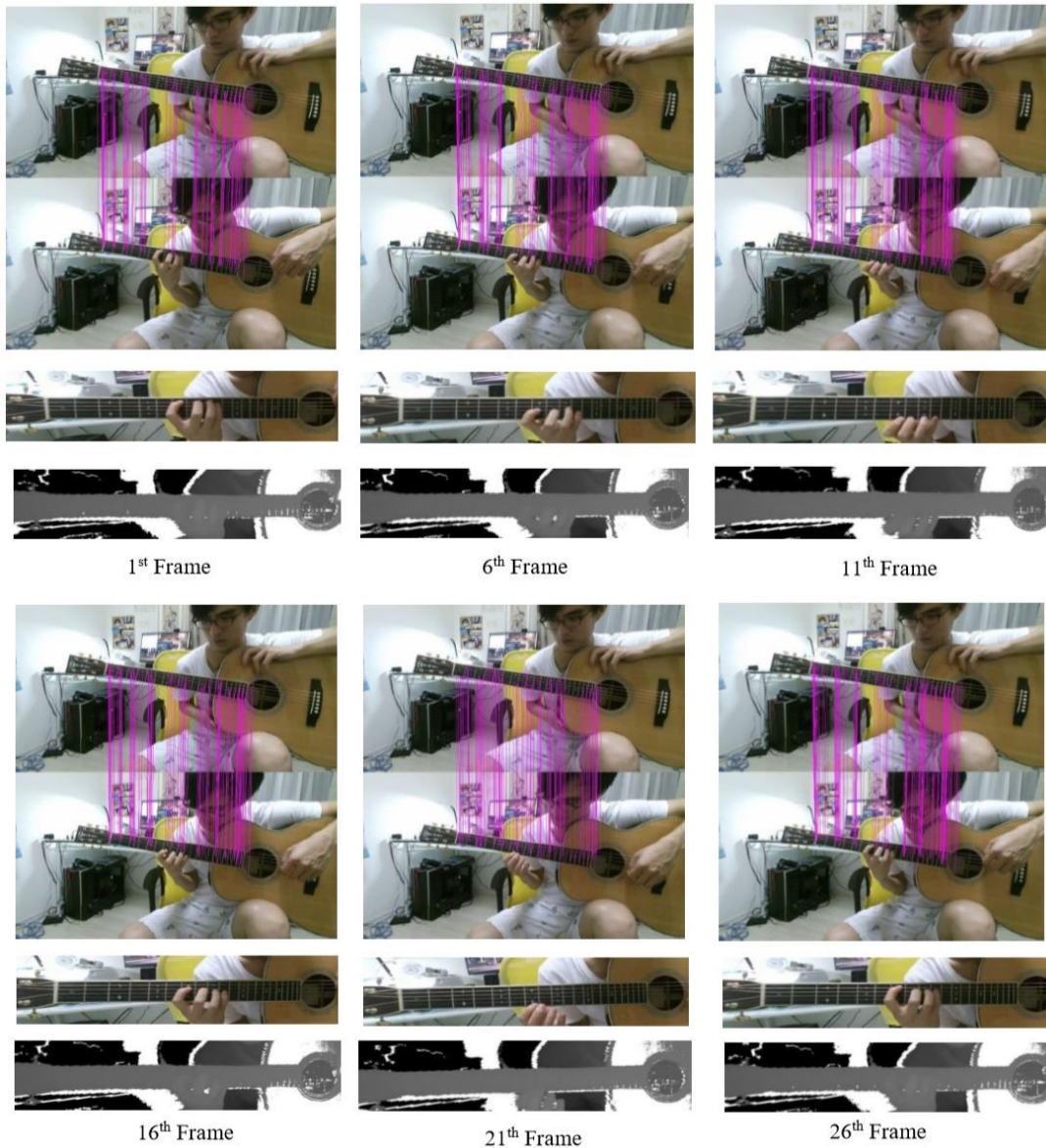|	Processing Time Per Frame	FPS
Our Work (SIFT + Modified RANSAC)	2.5 s	0.4
SUFT + Modified RANSAC	2.1 s	0.47
SIFT + RANSAC	2.3 s	0.43
SUFT + RANSAC	1.7 s	0.58

Figure 4: An Example of Tracking Result based on Our method on 3D Image Sequence

less efficient than Fullyconvolutional net (35 FPS) [15], but Fully-convolutional net also needs at least 400 images to label and train, which would cost 10 hours to train with GPU acceleration. More importantly, the mean error of 4.17 mm is only 4/10 of the distance between adjacent strings on guitar fretboard, while 10 mm (other related work's mean error) is almost the distance between the strings on guitar fretboard, which means if the mean tracking error is over 10 mm, it is very difficult to analyze which is string is pressed by fingers in the future work.

Conclusion and Future Work

This paper has proposed an algorithm for tracking the 3D position of the fretboard from the video of guitar plays. Specifically, we propose a SIFT matching procedure to track the guitar neck in 3D. First, we detect the SIFT features within the guitar fretboard and then match the detected points using KD-tree searching based matching algorithm frame by frame to track the whole fretboard. However, during the guitar plays, since the performer's fingers frequently overlap the fretboard, the feature points cannot always be matched accurately. Therefore, by using our modified RANSAC algorithm that filters out the tracking error of the feature points due to the overlapping issue mentioned before, perspective

Table 3: Comparison of Time Efficiency and Accuracy with Related Works

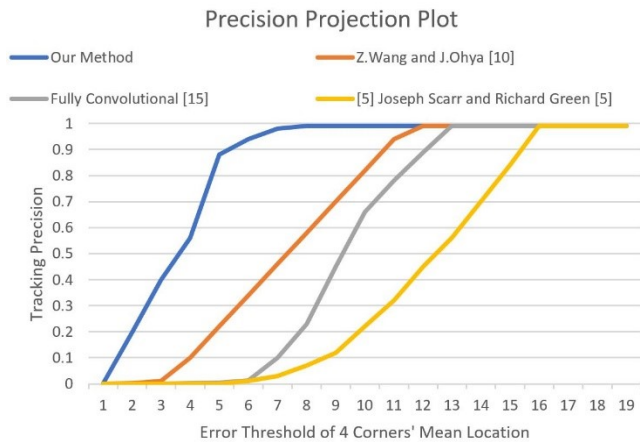|	Time Efficiency (FPS)	Mean Tracking Error (mm)
Our Work	0.4	4.17
Our Previous Work [10]	0.02	8.3
Fully Convolutional Net [15]	35	10.11
J. Scar [5]	0.89	13.3

Precision Projection Plot

Figure 5: Comparison with Ralted Works [5,10,15]

transformation matrix is obtained between the correctly matched feature points detected at the first and other frames. Consequently, the guitar neck is tracked correctly based on the perspective transformation matrix.

Experiments using 3000 frames of different guitar plays under different conditions show promising results of the proposed method. High accuracy: the total mean tracking error is only 4.17 mm and variance is 1.5 mm for the four tracked corners of the guitar fretboard is obtained. This result outperforms related tracking works including state-of-art Fully-convolutional Network.

Future work includes the subsequent processes after this guitar neck tracking algorithm. As we mentioned before, our final purpose of this topic is to analyze fingering of guitarist, our future work would be the hand pose estimation of the guitarist. With the guitar fretboard tracking result and hand pose estimation result, we would be able to assess fingerings.

References

[1] Radisavljevic, Aleksander, and Peter Driessen. "Path difference learning for guitar fingering problem." Proceedings of the International Computer Music Conference. Vol. 28. Sn (2004).

[2] Sayegh,S.I. "Fingering for String Instruments with the Optimum Path Paradigm" Computer Music Journal, vol.13, No. 3, Fall 1989, pp. 76-83 (1989).

[3] Radicioni, Daniele, and Vincenzo Lombardo. "Guitar fingering for music performance." strings. Vol. 40. No. 45 (2005)

[4] Y. Motokawa and H. Saito, "Support system for guitar playing using augmented reality display," in Proceedings of the 2006 Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)- Volume 00. IEEE Computer Society, pp. 243–244 (2006).

[5] Joseph Scarr and Richard Green, "Retrieval of Guitarist Fingering Information using Computer Vision," Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference, ISSN：2151-2191 , pp. 1–7, (2010).

[6] A. Burns, "Visual Methods for the Retrieval of Guitarist Fingering", Proceeding of the 2006 conference on New interfaces for musical expression ISBN:2-84426-314-3, pp.196-199, (2006).

[7] Chutisant Kerdvibulvech and Hideo Saito, "Real-Time Guitar Chord Estimation By Stereo Cameras For Supporting Guitarists". In Proceeding of 10th International Workshop on Advanced Image Technology 2007 (IWAIT), pp.147-152, (2007).

[8] Chutisant Kerdvibulvech and Hideo Saito, "Guitarist Fingertip Tracking by Integrating a Bayesian Classifier into Particle Filters". International Journal of Advances in Human-Computer Interaction (AHCI), pp121-131, (2008).

[9] Wang, Zhao, and Jun Ohya. "Fingertips Tracking Algorithm for Guitarist Based on Temporal Grouping and Pattern Analysis." Asian Conference on Computer Vision (ACCV). Springer, Cham. pp. 212-226 (2016).

[10] Zhao W. and Ohya J. Detecting and Tracking the Guitar Neck Towards the Actualization of a Guitar Teaching-aid System. In The… international conference on advanced mechatronics: toward evolutionary fusion of IT and mechatronics: ICAM: abstracts Vol. 2015, No. 6, pp. 187-188 (2015)

[11] Lowe, D.G. Object recognition from local scale-invariant features. In International Conference on Computer Vision, Corfu, Greece, pp. 1150-1157 (1999).

[12] Sunil Arya and David Mount, Approximate Nearest Neighbor Queries in Fixed Dimensions, In Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms, Austin, United States, pp. 271-280 (1993).

[13] Li, Minjie, Liqiang Wang, and Ying Hao. "Image matching based on SIFT features and kd-tree." Computer Engineering and Technology (ICCET), 2010 2nd International Conference on. Vol. 4. IEEE (2010).

[14] http://www.guitarworld.com/three-steps-shred-fundamental-daily-practice-techniques-about-15-minutes

[15] Wang, Lijun, et al. "Visual tracking with fully convolutional networks." Proceedings of the IEEE International Conference on Computer Vision (2015).

Author Biography

Zhao WANG is now a current PhD candidate in the department of MME (Modern Mechanical Engineering) in Waseda University. He got the Bachelor Degree in Sun Yet-sun University in China (2010), and Master Degree in Waseda University (2015). Now he is mainly working on tracking algorithm of Computer Vision and Machine Learning.

Dr. Jun Ohya is a professor at the Department of Modern Mechanical Engineering, Waseda University, Japan. He earned his B.S., M.S., and Ph.D. degrees in Precision Machinery Engineering from the University of Tokyo in 1977, 1979, and 1988, respectively. Dr. Ohya is a member of IEEE, IEICE, the Information Processing Society of Japan, etc. His research fields include image processing, computer vision, virtual reality, multimedia, pattern recognition.