Automatic Piano Transcription using Computer Vision

Annabelle Ritchie
Computer Science Department
University of Canterbury
Christchurch, New Zealand
annabelleritchie42@gmail.com

Richard Green
Computer Science Department
University of Canterbury
Christchurch, New Zealand
richard.green@canterbury.ac.nz

***Abstract*—This paper proposes a method for automatic music transcription from piano performance. A solution for a more flexible approach is implemented, allowing easy transference of the method to a makeshift paper piano.**

The keyboard is recognized using Canny edge and Hough line detection algorithms. Background subtraction is then implemented by masking for hand colour, and contouring is applied after opening and closing. A convex hull is found around the hand contours, and convexity defects on the convex hull are found to identify fingertips. Simple gesture analysis is then applied to the fingertips instead of using standard difference imaging as in previous work.

The solution does not perform as well as previous research, with an accuracy of 17.8%, a precision of 19.0%, a recall of 72.7%, and an overall F_1 score of 0.301.

***Index Terms*—computer vision, automatic music transcription, piano transcription**

I. Introduction

One of the most time-consuming aspects of composing music is physically writing down the music. Modern transcription programs exist, which speed up the process, but the transcription must still be done manually. Automatic transcription of music from an instrument could accelerate this process immensely, which is why it is a useful area of research. A standard composing instrument is the piano. Many electronic keyboards offer a MIDI output which can then be translated into an appropriate format for reading. However, this is not an option for real pianos.

It makes sense, while tackling the problem of piano transcription from a physical piano, to think about the ways in which this method could be expanded. For instance, if the user doesn't have a piano ready to hand, it would be interesting to see if a makeshift paper piano could replace an actual piano keyboard. This would offer a much more cost-effective solution for budding composers with a reliable implementation.

We propose a computer vision method that will be more flexibly converted to analysing and transcribing performance from a cheap, makeshift piano.

II. Background

Using computer vision to transcribe music from an instrument is a well-researched area. As well as many methods of transcription from piano [1][2][8][12][13][5][14][9], transcription from guitar [11], violin [15][10], and drums [4] as well as many other instruments have been attempted. There are several different approaches to piano transcription in particular to note:

- Purely audio-based, using neural networks to identify notes [8],
- Purely vision-based [1], and
- A combination of audio-visual analysis [13].

This paper focusses on purely vision-based methodology. Within this, there are three clear steps to the analysis required for piano transcription:

- Key segmentation and recognition,
- Hand detection and fingertip recognition, and
- Pressed key detection.

Various approaches to each of these from established research is discussed in detail. The implemented solution is then outlined, with quantitative results given and discussed, and some future work suggested.

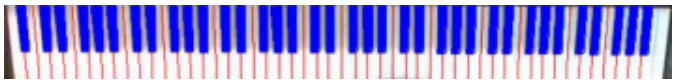

Fig. 1. Canny and Hough combined with Otsu thresholding for key segmentation [2].

A. Key Segmentation

The first step is to identify the individual piano keys in an image. This requires the corners of the piano, the lines between white keys, and the difference between black and white keys to be detected.

Akbari's ClaVision [2] and Deb's image analysis approach [3] both use Canny edge detection and Hough line transforms to detect the boundaries between the white keys, and an Otsu thresholding method to differentiate between the black and the white keys (Figure 1). This combination of methods yielded an accuracy of 95.2% in ClaVision, which was also a result of their illumination normalization. However, in Deb's work, errors were seen under bright lighting conditions (a quantifiable accuracy is not given).

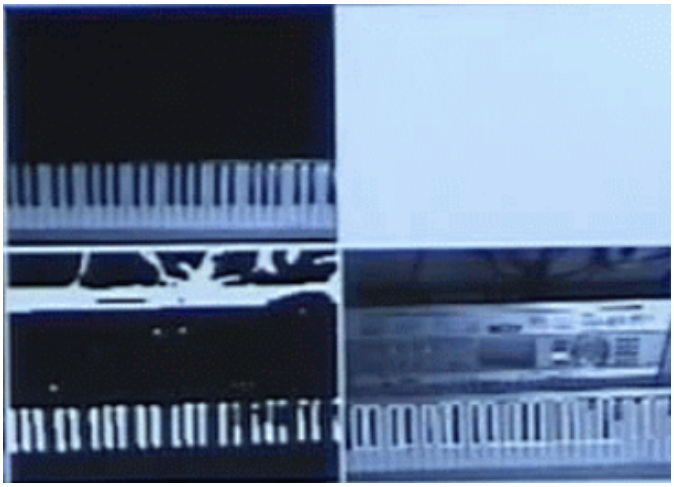

Fig. 2. Morphological strategy for key segmentation [5].

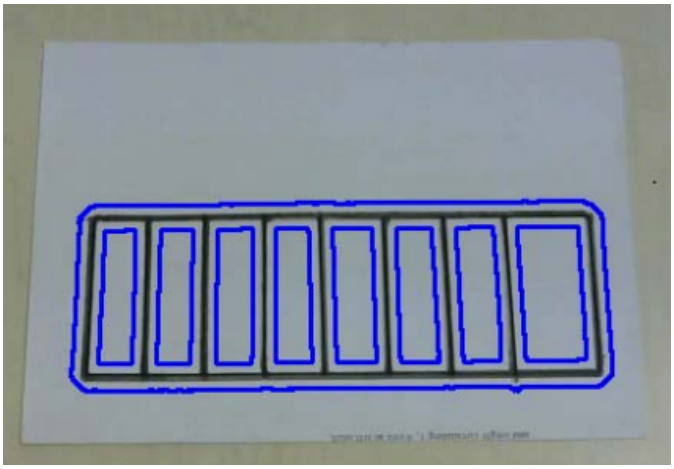

Fig. 3. Key segmentation on a paper piano [12].

A different method for key segmentation is shown in Gorodnichy's work [5], where the keyboard is detected based on a morphological masking strategy that uses a threshold to differentiate between black and white (Figure 2). Lines are then detected as best fits the blob patterns vertically. This yielded 'almost 100% accuracy', given that the piano keyboard was oriented square relative to the camera frame.

A solution by Vishal [12] also uses Canny edge detection, but because this paper analyses a two-dimensional paper piano, the solution also implements contouring and morphological reconstruction to detect the individual keys (Figure 3).

B. Hand Detection

The method of hand detection depends on how the system detects key presses. For Deb and Akbari, key presses are detected using the actual key image rather than through gesture analysis, so the hand and fingertips are not explicitly detected. These systems instead only detect the hands using a difference image technique compared to the background piano key image to determine which keys the hand is currently occluding.

Fig. 4. Hand identification using crevice detection[5].

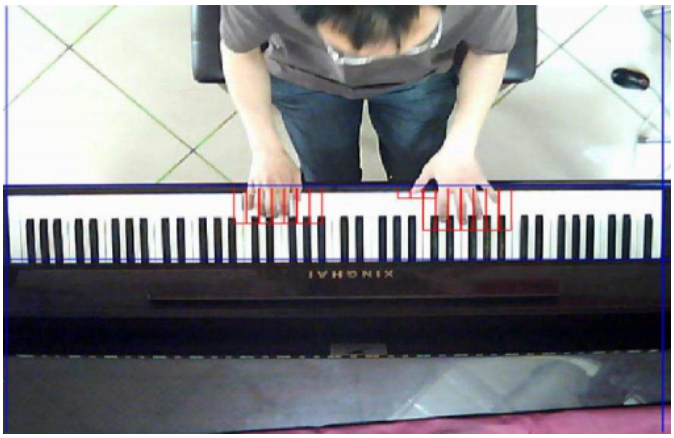

Fig. 5. Hand identification using skin colour detection and morphology [13].

In Gorodnichy's work, background subtraction is also used to detect the mask outline of the hands. Then a crevice detection algorithm (which detects crevices in a convex hull) is used in combination with the MIDI output of the piano to place a hand template on the image over where the hand is detected to be (Figure 4). Because the system relies so heavily on the gaps between fingers, it is limited when fingers become too close or occlude one another.

Wan's system for automatic transcription [13] uses skin colour detection instead of background subtraction in order to detect the hand mask by detecting human skin's specific chromatic distribution. The resulting mask then undergoes contouring, dilation and erosion to close gaps and erase noise, and the result of this is overlapped with the detected keys to show which keys are currently being occluded (Figure 5). This was deemed 'sufficient for detecting the ranges of the pianist's hands' positions'.

Vishal's two-dimensional paper piano solution not only needs to differentiate between the background and the hands, but also between the hands and the shadows. This is done using an Otsu thresholding technique (Figure 6). The distance between the fingertip and the shadow fingertip is then calculated. This relies heavily on a specific lighting condition, but offers valuable insight into a unique method for a paper piano.

C. Key Pressed Detection

Once the hand and key segmentation boundaries have been detected, the pressed keys can be detected. Akbari does this by combining the hand occlusion data with a difference imaging technique, in order to detect which keys have changed positions (Figure 7). This, combined with illumination normalization, yields a high accuracy result of ¿95%.

Deb uses a machine learning method to detect key presses

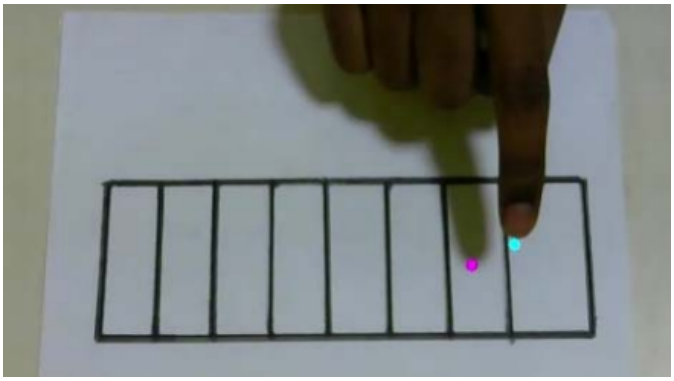

Fig. 6. Separation of fingertip and fingertip's shadow [12].

Fig. 7. Pressed keys identification using difference imaging[2].

by observing the shadows that occur on the side of a key when a key is pressed down. This doesn't work when two adjacent keys are pressed, however, which is a limitation of their system. Their results show an average accuracy of 87%.

Vishal's two-dimensional paper piano solution implements key touch detection using the distance calculated between the fingertip and the fingertip's shadow. The contour that the fingertip lies within is then identified as the key being pressed. This method is said to be 'accurate and effective', but no quantifiable results are given.

D. Illumination Normalization

It is worth noting that Akbari, Deb, and Vishal all implement illumination normalization in order to account for variance in lighting. This helps to remove errors associated with shadows and glare that would occur otherwise. For example, Akbari implements a Move-Towards filter, which compares the shift between a background image and the current image and shifts the rest of the pixels accordingly [1].

III. Solution

While Akbari and Deb show a high accuracy rate from difference imaging to detect key presses, this method wouldn't work for a paper piano system, as it relies on the shadows caused by adjacent keys not being pressed. The finger shadow detection method shown in Vishal's system works well in specific cases, but would only work with an appropriately angled light overhead, which is not always feasible. In order to find a better way of determining key presses for a paper piano, the solution implemented here models the hand based on contouring and convex hull methodology (similar to Gorodnichy's work), and then uses the movement of each fingertip to detect when a key is pressed.

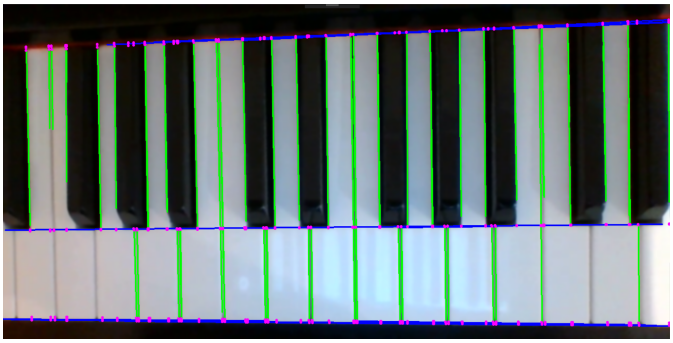

Fig. 8. Canny Edge and Hough Line detection.

A. Canny Edge and Hough Line Detection

First, Canny edge detection is applied to the keyboard to detect the edges of each piano key. Then, based on the edges detected, Hough line detection is applied to filter out noise and identify the appropriate length lines in order to identify individual keys on the keyboard (Figure 8). This is similar to the methods implemented in [2] and [3]. Once appropriate threshold values were selected, this solution proved to be very accurate, detecting 92% of key segmentation lines.

B. Background Subtraction

The next step is to differentiate the hand from the background. This is done by masking for skin colour and applying opening and closing to the result. Opening takes away small specks of background noise, while closing closes up small holes in the mask. This method is vulnerable to changes in skin colour and lighting. The mask is currently manually changed based on skin colour, but it could be selected automatically for a more reliable result. The resulting morphological transform is shown in Figure 9.

C. Contour and Convex Hull Application

Once the hand shape is segmented, contouring is applied around the mask, and a convex hull is applied to the contour (Figure 10). To avoid putting a hull around any background noise, a specific threshold area is required for the contour in order to get a convex hull. The fingertips are then detected by finding the convexity defects in the contour, and selecting the defects that stick out of the contour rather than the ones that point in. This is done by calculating the angle between points on the convex hull and the point of the convexity defect:

$$cos(\theta) = \frac{b^2 + c^2 - a^2}{2bc}$$

Fig. 9. Morphological transform resulting from masking for skin colour.

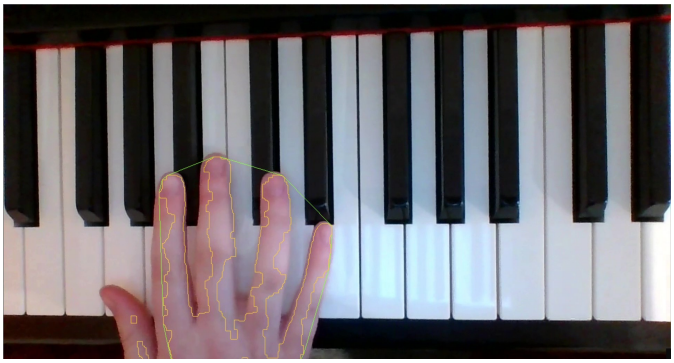

Fig. 10. Contouring and convex hull based on morphological transform.

Where a, b, and c's lengths form a triangle between two points on the convex hull and the convexity defect. The value of θ was tested experimentally and an appropriate threshold at 60^o was determined for identifying fingertips.

D. Key Press Identification

Simple gesture identification was performed on the fingertip positions to identify key presses. This was done by measuring the movement of the fingertip position along the y axis and comparing it to a threshold. For fingertips that were identified as currently pressing, the keyboard segmentation lines nearest to the fingertip were determined and that key was identified as being pressed. An example of a correctly identified key press is shown in Figure 11.

IV. Results

To implement the proposed method, the following equipment was used:
OS: Windows 10 Home
IDE: IDLE
Language: Python 3.7
Device: Desktop PC
Processor: i5-7400
Camera: Logitech 720p/30fps webcam
OpenCV version: 4.3.0
Test data: Test data in the form of short video clips from a webcam set up over a real piano was taken.

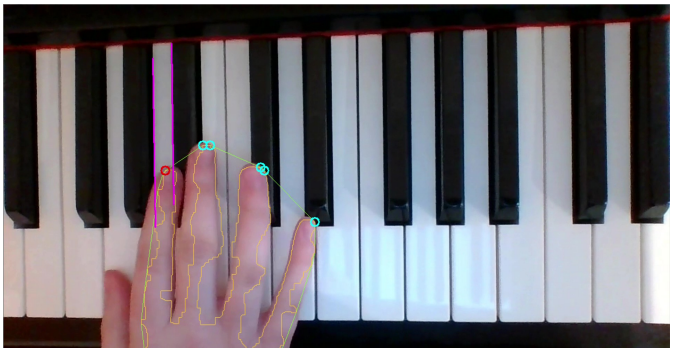

Fig. 11. Key press detection based on fingertip movement.

To determine quantifiable results, the algorithm was observed with several differing values of threshold for the y-velocity of the fingertips for gesture analysis. For each value, the number of key presses were identified, as well as the number of key presses not identified and the number of falsely identified key presses (see 13 for a visual example) recorded by the proposed method. These values are shown in Tables 1 - 5.

Results were then analysed using an approach standard to classification algorithms [7]: finding accuracy, precision, and recall, defined as:

$$accuracy = \frac{correct\ identifications}{all\ total\ identifications}$$

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

These terms are defined below:

- **True Positive** - an identified key press somewhere within the timeframe of an actual key press.
- **False Positive** - an actual key press with no key press identified by the method.
- **False Negative** - an identification of a key press with no actual key press.

The F_1 score, which is an overall measure of the quality of the algorithm, was then calculated for each y-velocity threshold using the formula:

$$F_1 = \frac{2 \cdot recall \cdot precision}{recall + precision}$$

F_1 score is a standard metric to report for studies on music transcription [6]. These results are shown in Tables 6 and 7. The solution with the highest F_1 score has a y-velocity threshold of 10. Therefore, this method has an accuracy of 17.8%, a precision of 19.0%, a recall of 72.7% and a F_1 score of 0.301.

TABLE I
REAL AND IDENTIFIED KEY PRESSES DURING A 10S VIDEO CLIP OF PIANO PERFORMANCE, Y-VELOCITY THRESHOLD = 10.

	Key Press	No Key Press
Key Press Identified	8	34
No Key Press Identified	3	-

TABLE II
REAL AND IDENTIFIED KEY PRESSES DURING A 10S VIDEO CLIP OF PIANO PERFORMANCE, Y-VELOCITY THRESHOLD = 20.

	Key Press	No Key Press
Key Press Identified	6	31
No Key Press Identified	5	-

A. Comparison with Previous Work

In comparison, Akbari's piano transcription system has an F_1 score of 0.796, with a recall of 74.6% and a precision of 85.4%. Precision is where the proposed solution here lacks quality, with a high amount of false positives. Deb's study reports F_1 scores of over 0.9 without major changes in illumination. It is clear that this solution is not at the same level as previous work. There are improvements that need to be made to this gesture-focussed method of piano performance recognition before it can compete with the standard difference imaging method.

B. Limitations

The solution is vulnerable to any change in lighting, especially because the mask for the hand relies on specific colour values. Unfortunately the lighting available made it difficult to mask for specifically black and white colour values. If this were possible with the test data on hand, the mask could then be inverted, allowing for any skin tone to be masked for and detected as part of the hand. However, it is likely that very pale and very dark skin tones would still require good calibration and steady lighting conditions. An example of the limitations of the hand recognition is shown in Figure 12, where the convex hull has mistakenly been split into two smaller parts as the centre of the hand has not been appropriately recognized.

While the solution is able to identify fingertips correctly, it often identifies multiple duplicate fingertips in a position where there should just be one (see Figure 11). This results in the y-velocity thresholding algorithm picking up multiple key presses where there should just be one.

TABLE III
REAL AND IDENTIFIED KEY PRESSES DURING A 10S VIDEO CLIP OF PIANO PERFORMANCE, Y-VELOCITY THRESHOLD = 30.

	Key Press	No Key Press
Key Press Identified	6	30
No Key Press Identified	5	-

TABLE IV
REAL AND IDENTIFIED KEY PRESSES DURING A 10S VIDEO CLIP OF PIANO PERFORMANCE, Y-VELOCITY THRESHOLD = 40.

	Key Press	No Key Press
Key Press Identified	5	26
No Key Press Identified	6	-

TABLE V
REAL AND IDENTIFIED KEY PRESSES DURING A 10S VIDEO CLIP OF PIANO PERFORMANCE, Y-VELOCITY THRESHOLD = 50.

	Key Press	No Key Press
Key Press Identified	4	24
No Key Press Identified	7	-

TABLE VI
RATE OF ACCURACY, PRECISION AND RECALL FOR DIFFERENT Y-VELOCITY THRESHOLDS.

Y-Velocity Threshold	Accuracy	Precision	Recall
10	0.178	0.190	0.727
20	0.143	0.162	0.545
30	0.146	0.167	0.545
40	0.135	0.161	0.455
50	0.114	0.143	0.364

V. CONCLUSIONS

A. Summary

The solution implemented works with an accuracy of 17.8%, a precision of 19.0%, a recall of 72.7%, and an overall F_1 score of 0.301. This solution, overall, performed worse than previous work, but has the potential to offer greater flexibility when transferring the method to a paper piano.

B. Future Work

In future, calibration for varying light conditions could be implemented. This has been part of several previous successful solutions [2], with good results. This would allow the solution

TABLE VII
F_1 SCORE FOR DIFFERING Y-VELOCITY THRESHOLDS.

Y-Velocity Threshold	F_1 score
10	0.301
20	0.250
30	0.256
40	0.238
50	0.205

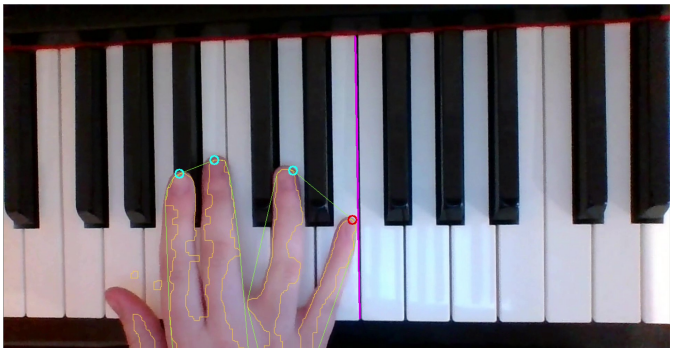

Fig. 12. Convex hull split in two due to poor hand recognition.

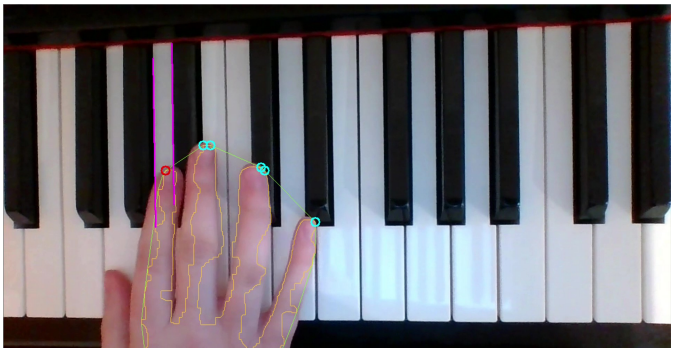

Fig. 13. A falsely identified key press (false positive error).

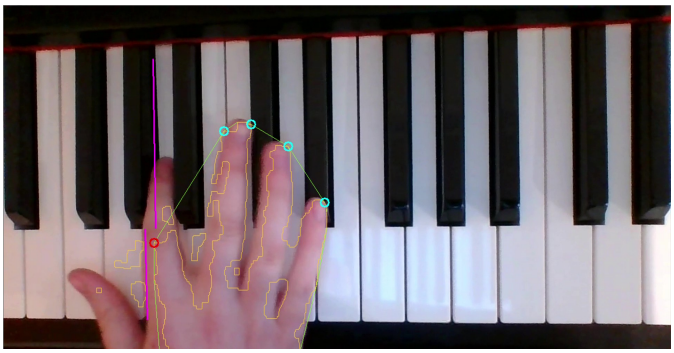

Fig. 14. A correctly identified key press.

to work in different settings and provide a more robust and reliable result.

Masking for different hand colours would be a good step to take. This could be done either by manually expanding the range of skin colours masked for, or by masking for black and white and inverting the colour mask. More testing would need to be done to determine the best approach here.

The obvious next step to determine how flexible this solution is would be to do rigorous testing and calibration for a paper piano instead of a physical keyboard. It would be interesting to try this with a number of light sources to determine how robust the algorithm is with respect to varying illumination and shadows.

VI. Acknowledgements

The authors would like to thank the University of Canterbury Computer Science Department for facilitating the project.

References

[1] Mohammad Akbari and Howard Cheng. "Clavision: visual automatic piano music transcription." In: *NIME*. 2015, pp. 313–314.

[2] Mohammad Akbari, Jie Liang, and Howard Cheng. "A real-time system for online learning-based visual transcription of piano music". In: *Multimedia Tools and Applications* 77.19 (2018), pp. 25513–25535.

[3] Souvik Sinha Deb and Ajit Rajwade. "An image analysis approach for transcription of music played on keyboard-like instruments". In: *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*. 2016, pp. 1–6.

[4] Olivier Gillet and Gaël Richard. "Automatic transcription of drum sequences using audiovisual features". In: *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* Vol. 3. IEEE. 2005, pp. iii–205.

[5] Dmitry O Gorodnichy and Arjun Yogeswaran. "Detection and tracking of pianist hands and fingers". In: *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. IEEE. 2006, pp. 63–63.

[6] Curtis Hawthorne et al. "Onsets and frames: Dual-objective piano transcription". In: *arXiv preprint arXiv:1710.11153* (2017).

[7] Renuka Joshi. "Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures; 2018". In: *URL: https://blog. exsilio. com/all/accuracy-precision-recall-f1-score-interpretation-ofperformance-measures/(visited on 09/30/2018)* ().

[8] Rainer Kelz et al. "On the potential of simple framewise approaches to piano transcription". In: *arXiv preprint arXiv:1612.05153* (2016).

[9] R.D. Milligan. In: *URL: https://rdmilligan.wordpress.com/2015/10/22/paper-piano-using-python-and-opencv/(visited on 28/05/2020)* ().

[10] Albert Nisbet and Richard Green. "Capture of Dynamic Piano Performance with Depth Vision". In: ().

[11] Joseph Scarr and Richard Green. "Retrieval of guitarist fingering information using computer vision". In: *2010 25th International Conference of Image and Vision Computing New Zealand*. IEEE. 2010, pp. 1–7.

[12] B. Vishal and K. D. Lawrence. "Paper piano — Shadow analysis based touch interaction". In: *2017 2nd International Conference on Man and Machine Interfacing (MAMI)*. Dec. 2017, pp. 1–6. DOI: 10.1109/MAMI.2017.8307890.

[13] Yu Long Wan et al. "Automatic transcription of piano music using audio-vision fusion". In: *Applied Mechanics and Materials*. Vol. 333. Trans Tech Publ. 2013, pp. 742–748.

[14] Ihab Zaqout et al. "Augmented piano reality". In: *International Journal of Hybrid Information Technology* 8.10 (2015), pp. 141–152.

[15] Bingjun Zhang et al. "Visual analysis of fingering for pedagogical violin transcription". In: *Proceedings of the 15th ACM international conference on Multimedia*. 2007, pp. 521–524.