Visual music transcription of clarinet video recordings trained with audio-based labelled data

Pablo Zinemanas
Universidad de la República
Uruguay
pzinemanas@fing.edu.uy

Pablo Arias
ENS Cachan, Université Paris Saclay
France
pablo.arias@cmla.ens-cachan.fr

Gloria Haro
Universitat Pompeu Fabra
Spain
gloria.haro@upf.edu

Emilia Gómez
Universitat Pompeu Fabra
Spain
emilia.gomez@upf.edu

Abstract

Automatic transcription is a well-known task in the music information retrieval (MIR) domain, and consists on the computation of a symbolic music representation (e.g. MIDI) from an audio recording. In this work, we address the automatic transcription of video recordings when the audio modality is missing or it does not have enough quality, and thus analyze the visual information. We focus on the clarinet which is played by opening/closing a set of holes and keys. We propose a method for automatic visual note estimation by detecting the fingertips of the player and measuring their displacement with respect to the holes and keys of the clarinet. To this aim, we track the clarinet and determine its position on every frame. The relative positions of the fingertips are used as features of a machine learning algorithm trained for note pitch classification. For that purpose, a dataset is built in a semiautomatic way by estimating pitch information from audio signals in an existing collection of 4.5 hours of video recordings from six different songs performed by nine different players. Our results confirm the difficulty of performing visual vs audio automatic transcription mainly due to motion blur and occlusions that cannot be solved with a single view.

1. Introduction

Music can be represented in different formats and modalities, e.g. audio, video, musical scores or text from lyrics [7]. In this work, we focus on the auditory and visual modalities of a music performance, which are complementary facets of music perception.

Automatic transcription aims to compute a symbolic music representations (e.g. MIDI) from music audio recordings. State-of-the-art algorithms can perform this task in an effective way from monophonic music recordings (i.e. one note played at a time), but the task becomes challenging when dealing with polyphonic music signals (i.e. when there are overlapping notes as usually happens in most music recordings) [13].

In this work, we address the task of automatic transcription from visual information, which is useful in situations where the audio modality is missing or in polyphonic music recordings where the effectiveness of audio-based methods decreases. We study the feasibility of the task in the monophonic case, and we focus on a particular instrument, the clarinet. The clarinet is a musical instrument belonging to the group known as *woodwind instruments*, and it carries a soloist or ensemble role in the orchestra. It has a single-reed mouthpiece, a straight cylindrical tube with an almost cylindrical bore, and a flared bell. The clarinet is played by blowing through its mouthpiece, opening/closing a set of holes and pressing a set of keys using fingers of both hands of the player. Nine fingers are used for that purpose, one in the back of the clarinet (a thumb) and the other eight in the front. The clarinet is held by the other thumb also in the back and is the only finger that it is not allowed to move freely.

A musical note in symbolic format is defined by three main characteristics: onset (beginning time), duration and pitch, which represents how high or low a note is. The pitch range of a clarinet spans nearly four octaves (around 44 different note pitches), and the combination of closed holes and keys determine the played pitch. There are standard fingering positions, some of them presented in Figure 6.

The clarinet is the focus of a previous related research on

the visual detection of note onsets [1], where a neural network architecture based on five 3D convolutional layers was presented and achieves a mean F-score of 25.7%. This result is significantly lower than the one obtained from audio-based algorithms, where state-of-the-art methods provides an F-measure around 77% for wind instruments as indicated at last MIREX evaluation campaign[1]. Visual onset detection then seems to be more challenging than its audio counterpart.

In our work we target the detection of the played pitch from visual analysis, which we hypothesize it is also a challenging task to perform visually and requires previous knowledge on how the clarinet is played. We exploit audio-based automatic transcription methods, in particular pitch estimation algorithms, for the automatic labelling of an existing video dataset recorded by Bazzica et al. [1]. This labelled dataset is used to train and evaluate a visual music transcription method.

Given the small size of the annotated dataset (see Section 2.1), we discard the use of deep learning strategies. Instead, we opt for a knowledge-driven feature-based approach which is described in Section 3. Experimental results are then presented in Section 4, and some conclusions and ideas for future work are provided in Section 5.

2. Music material

In our research, we take advantage of the *Clarinetists for Science* (C4S) dataset [1] built by recordings of 9 performers. Each of them performs six different songs, obtaining 54 audiovisual recordings with 4.5 hours in total. The videos have been recorded at 30 fps. The dataset includes annotations of note onsets that were first automatically labelled and then manually checked.

2.1. Ground-truth extraction

The dataset has labels of note onsets but lacks of pitch information. For training our system, we need the pitch of each note as the ground-truth. In order to obtain this information, we first estimate the fundamental frequency contour (f0) using the pYIN algorithm [16] available as a plug-in of Sonic Annotator [2]. Then, we associate each onset label with a note pitch, calculating the f0 median value in the interval between the onset and the next one and quantizing this value to the closest semitone. Finally, the labels are manually checked and corrected using Sonic Visualizer [3]. Figure 1 shows the histogram of video frames per pitch, which is coherent with the pitch range of the target instrument. In order to avoid possible errors on onsets or pitch labels and obviate fast transitions, notes with duration less than 200 ms (6 frames) are discarded. Furthermore, for longer notes, the two borders of 100 ms (3 frames) are also discarded.

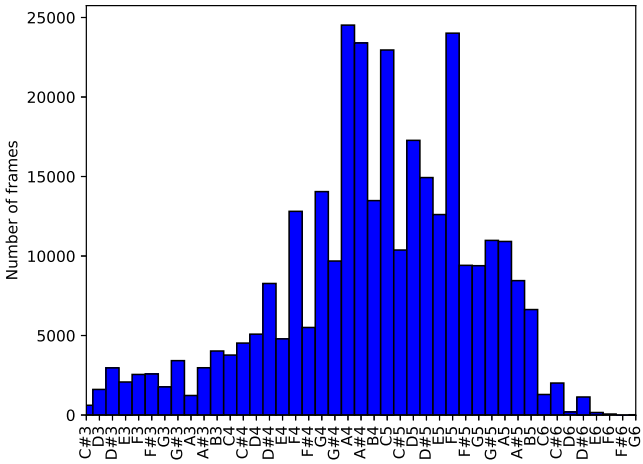


Figure 1: Histogram of frames per pitch.

3. Proposed approach

Note transcription from visual information requires the detection of fingertips together with the keys and holes of the clarinet and the identification of whether if every finger is pressing the corresponding hole/key or not. This last task is specially difficult when a single view is available since, depending on the relative position of the clarinet and the camera and the playing style of the performer, a finger that is not pressing the hole/key may occlude the hole/key in the image and then it makes difficult to distinguish if the hole/key is being pressed or not.

The proposed system uses image processing algorithms for feature extraction and a machine learning algorithm for classification. Since we are using a single frontal view of the clarinet we only use the eight fingers placed in its frontal part. The features we extract are the relative position (in the image domain) of fingertips with respect to the corresponding holes or keys of the clarinet, resulting in a 16 dimensional vector of scalar values. These features are denoted as (x_i^r, y_i^r), for $i = 0, \ldots, 7$, where the subindex i denotes each finger and it is associated to each one of the holes or keys.

The feature extraction is performed as shown in Figure 2. The upper branch of the diagram illustrates the extraction process of the absolute positions of fingertips (x_i^a, y_i^a). This process consists in (1) extraction of the region of interest (ROI), (2) skin segmentation, (3) fingertip detection and (4) Kalman filter. The lower branch shows the process of extracting the positions of holes and keys $\left(x_i^h, y_i^h\right)$ using a (5) matching algorithm that calculates key points matches between every frame and a reference frame and robustly finds

[1] http://nema.lis.illinois.edu/nema_out/mirex2016/results/aod/resultsperclass.html

a planar homography that characterizes these matches.

3.1. Region of interest

We extract the regions of interest (ROIs) used in [1] and provided in the dataset. These ROIs are: mouth, left/right hands, and clarinet tip. Figure 3 left shows an example of these ROIs. Since we want to extract fingering information, we only use the left and right hands ROIs. Figure 3 right shows the subimage extracted from these two ROIs, these kind of images are the ones we use in our algorithm.

3.2. Skin segmentation

Several works have addressed the skin segmentation problem, such as Bayesian methods [11], Gaussian models [20] and threshold-based methods [5, 4, 21]. Saxen and Al-Hamadi published a complete review and evaluation of the state of the art [22].

We use a threshold-based method in the YCrCb color space [4] that achieves good results in the literature [23, 17, 18]. In this space the Y value is related to luminance and the Cr and Cb channels encode the chrominance. After transforming the input image from the RGB to the YCrCb colorspace, a pixel is classified as skin if the following conditions hold simultaneously:

$$Y > 130, \qquad 143 < Cb < 200, \qquad 92 < Cr < 140.$$

We define a skin mask $M(x, y)$ which is equal to 1 if the pixel (x, y) is classified as skin, and 0 otherwise.

3.3. Fingertip detection

We detect fingertips by searching for blobs of a particular size in the skin mask image. A blob is a region that is either brighter or darker than the background and surrounded by a smoothly curved edge [14, 6]. This approach works because the highest response to a blob detector applied to the skin mask is located at the fingertips. In other words: they are the most blob-like structure on the skin mask (for a specific blob size).

A common method for blob detection is to apply the Laplacian of Gaussian (LoG) and find its local extrema [9]. The first step is to apply a Gaussian filter g_σ with scale σ to the skin mask, $I = g_\sigma * M$. After this, the discrete Laplacian $\nabla^2 I$ is computed using standard finite differences. Figure 4b shows the result of $\nabla^2 I$. The blobs candidates are the local minima of $\nabla^2 I$. We further refine these candidates by a threshold and check if the eigenvalues, $\lambda_0 \leqslant \lambda_1$, of the Hessian matrix of $\nabla^2 I$ satisfy the following conditions

$$\lambda_0 > 0, \qquad \lambda_1 < 5\lambda_0.$$

The first condition ensures that both eigenvalues are positive, thus it is indeed a strict minimum and the second bounds the aspect ratio of the level sets of the $\nabla^2 I$. This filters out local minima originated by elongated structures in M.

Note that after the ROI extraction, the fingertips are always approximately of the same size (in the dataset we assume the distance between the camera and the clarinet is nearly constant). Thus instead of using a scale space (as is commonly done in computer vision) we use a fix scale parameter σ.

Once the blob candidates are filtered, it is necessary to associate them to each finger. Therefore, for the finger i, we find the nearest blob candidate. If the Mahalanobis distance[2] between the candidate and the position of the hole/key i is lower than a threshold, the nearest candidate is associated to that finger. Note that the positions of the holes and keys $\left(x_i^h, y_i^h\right)$ are calculated as explained in Section 3.5. The output of this stage for frame k is a vector $\hat{\mathbf{x}}_k = [\hat{x}_{1,k}, \hat{y}_{1,k}, ..., \hat{x}_{8,k}, \hat{y}_{8,k}]^T$ of 16 components with the estimated 2D coordinates for each finger. If no blob candidate has been associated to a hole/key, we set the corresponding coordinates to -1, meaning that the coordinates of the corresponding fingertip are unknown.

3.4. Kalman filter

We use a Kalman filter [12] to smooth the trajectories of the estimated fingertip positions. This also allows us to interpolate the unknown positions of the fingertips that were not detected. We use a constant velocity filter with 16 observations (the vector $\hat{\mathbf{x}}_k$) and 32 state variables $\mathbf{z}_k = [\mathbf{x}_k^T, \mathbf{v}_k^T]^T$, where $\mathbf{x}_k \in \mathbb{R}^{16}$ are the positions and $\mathbf{v}_k \in \mathbb{R}^{16}$ their velocities. This results in the following Gaussian linear model:

$$\begin{cases} \mathbf{z}_k = F\mathbf{z}_{k-1} \\ \hat{\mathbf{x}}_k = H\mathbf{z}_k + \mathbf{r}_k \end{cases}$$

where the state transition and observation matrices are:

$$F = \begin{bmatrix} I_{16} & I_{16} \\ 0_{16} & I_{16} \end{bmatrix}, \qquad H = \begin{bmatrix} I_{16} & 0_{16} \end{bmatrix}.$$

Here I_{16} is the identity 16×16 matrix and 0_{16} is a 16×16 matrix of zeros. The observation matrix H maps the state into the observed positions by selecting only the first 16 values (x and y coordinates). In our model the state transition is deterministic (there is no state transition noise). For the detected fingertips the noise of the observation $\mathbf{r}_k$ is modeled with zero mean and variance of 25. For the fingertips that have not been found in the fingertip detection step, we set the variance to ∞. In such cases the output of the Kalman filter corresponds to the Kalman prediction, i.e. the position is given by $(x_{i,k-1}, y_{i,k-1}) + (v^x_{i,k-1}, v^y_{i,k-1})$ and the speed is kept at $(v^x_{i,k-1}, v^y_{i,k-1})$.

[2]The Mahalanobis distance is calculated with a covariance matrix defined as $\Sigma = \left[\begin{smallmatrix} 3 & 0 \\ 0 & 10 \end{smallmatrix}\right]$ for weighting the distance on y axis more than x axis.

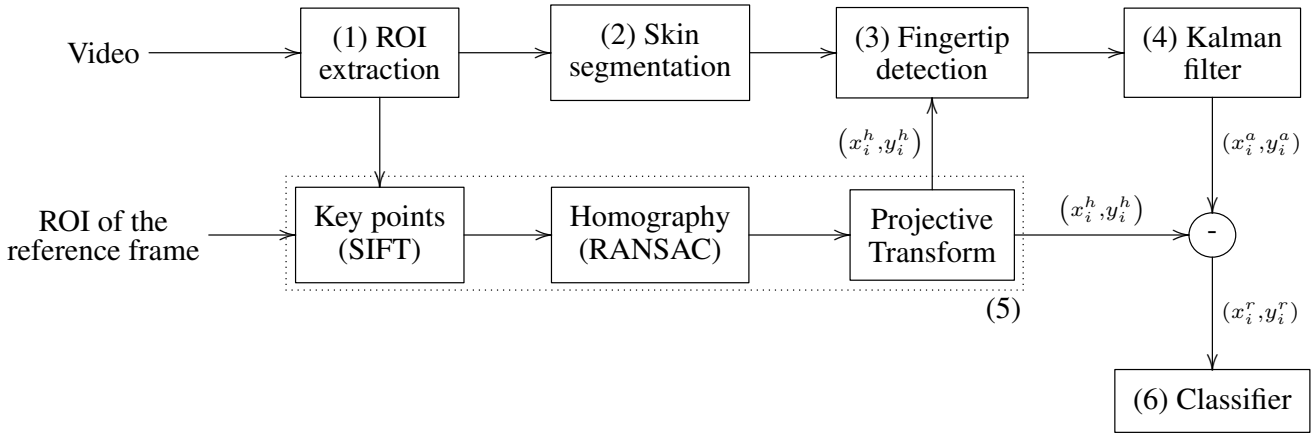
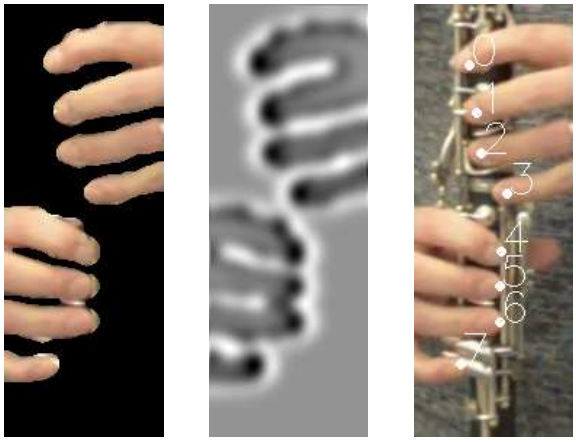


Figure 2: System block diagram.

Figure 3: Example of ROIs used in [1]. Left: Original frame and the four different ROIs marked in red. Right: Subimage extracted from the bounding boxes of ROIs corresponding to left and right hands.



(a) Mask $M(x, y)$ applied to the frame.

(b) Laplacian of Gaussian $\nabla^2 I(x, y)$ applied to $M(x, y)$

(c) Points (x_i^a, y_i^a) obtained after Kalman filter.

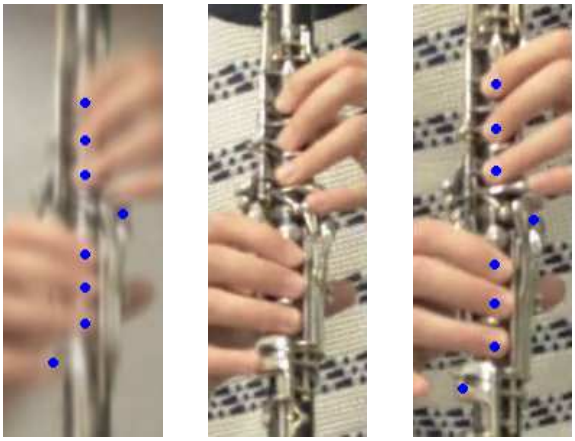
Figure 4: Fingertips detection. Illustration of the process of the upper branch in the diagram of Fig. 2.

We will denote the output of the Kalman filter using the superscript "a", to stress that these are *absolute* coordinates and differentiate them from the relative coordinates with respect to the ith hole/key. Thus $(x_{i,k}^a, y_{i,k}^a)$ are absolute coordinates for the ith finger in the kth frame.

3.5. Tracking the clarinet

The musician moves while playing the clarinet and some performers move more than others. As explained in Section 3.1 we use the tracked ROIs corresponding to the left and righ hands and which are already provided by the dataset [1]. As a result, there is only a slight motion of the clarinet among the different frames extracted from the ROIs regardless of the playing style of the musician. Due to this remaining slight motion in the images, the position of the clarinet is not constant along time.

Feature extraction can be improved knowing the position of holes and keys of the clarinet at every instant of time. This is carried out calculating the mean image of the videos corresponding to the ROIs of interest of each performer and manually marking the holes and keys positions. Let us define these points as $(\bar{x}_i, \bar{y}_i)$, an example is illustrated in Fig. 5a where these points are marked as blue dots. As noticed in the example, this mean image is always blurred because the clarinet is not completely aligned in the different ROI frames. Then, for each video a reference frame is found. This frame is *similar* to the mean image but not affected by blur (see Figure 5b and compare it to Figure 5a). For finding the reference frame, we first determine the fingertips positions as explained in previous subsections. As the holes/keys positions (x_i^h, y_i^h) we use the points $(\bar{x}_i, \bar{y}_i)$



(a) Mean image and points manually marked $(\bar{x}_i, \bar{y}_i)$.

(b) Reference frame.

(c) Points $\left(x_i^h, y_i^h\right)$ after applying the homography.

Figure 5: Tracking the holes and keys. Illustration of the process of the lower branch in the diagram of Fig. 2.

manually marked on the mean image. Then, we compute the sum of Euclidean distances to the points marked in the mean image:

$$D_k = \sum_i \text{dist}\left((\bar{x}_i, \bar{y}_i), \left(x_{i,k}^a, y_{i,k}^a\right)\right).$$

The reference frame corresponds to the time:

$$k_{ref} = \text{argmin}\left\{D_k\right\}.$$

For each frame, we calculate key points and their matches using Scale-invariant feature transform (SIFT) [15] and find a planar homography that relates the current frame to the reference frame using the DLT algorithm [10] and the Random Sample Consensus (RANSAC) procedure [8] in order to deal with outlier matches. Then, a projective transformation to the points $(\bar{x}_i, \bar{y}_i)$ is applied, using the homography previously found, in order to get the positions of holes and keys at each frame $\left(x_i^h, y_i^h\right)$. Figure 5c exemplifies this result. By using a 2D homography (3×3 non singular matrix) we are assuming that the front part of the clarinet can be approximated by a planar surface; although it is a rough approximation it gives good results in practice. Let us also remind that the purpose of the homography is to correct the residual weak motions in the images extracted from the tracked ROIs.

The last step of feature extraction is just to calculate the relative position of each fingertip as:

$$\left(x_i^r, y_i^r\right) = \left(x_i^a, y_i^a\right) - \left(x_i^h, y_i^h\right).$$

3.6. Classifier

Having defined specific features for our problem, we consider a simple classification algorithm. Thus, in order to train and validate our system, we use a Random Forest classifier with 100 trees implemented on *scikit-learn* library [19].

4. Experiments and results

For testing the performance of the system we use the *Clarinetists for Science* (C4S) dataset [1] and apply a 9-fold cross validation scheme. Each fold corresponds to one performer. Therefore, for each fold, a model is trained with the other 8 folds and validated with itself. Doing this, we have the most exigent results and we can evaluate the generalization capability of this work. For selecting the note classes, three criteria are used: (1) there is only one fingering position; (2) only the holes are considered; (3) try to keep a good balance inter-classes. Figure 6 shows the fingering of the 8 selected classes. As explained in Section 2, the dataset provides note onsets ground truth which was firstly obtained with the automatic method proposed in [1] and then manually checked. We automatically extracted the note pitch ground truth by means of audio-based techniques (more details in Section 2.1) and was manually validated afterwards.

Fold	Accuracy (%)
Performer 1	44.1
Performer 2	50.2
Performer 3	49.6
Performer 4	23.2
Performer 5	41.7
Performer 6	55.4
Performer 7	44.6
Performer 8	55.7
Performer 9	60.0
mean	**47.2**

Table 1: System performance for each fold.

Table 1 shows the accuracy for each fold and the system achieves a mean accuracy of 47.2 %. These results confirm the difficulty of visual transcription, as pointed out in previous studies [1], with respect to audio-based monophonic pitch estimation. The borderline cases can be analyzed by the way the performers play. In the best case, the performer does not move much, the frames are all in focus and thus the homography can be calculated correctly. The fingertips positions are also correctly found. In the worst case, the performer moves fast and then, a lot of frames are blurred. Although SIFT descriptors present some invariances and partial invariances they are not robust to blur and not enough correct matches are produced in order to estimate a good homography in these cases.

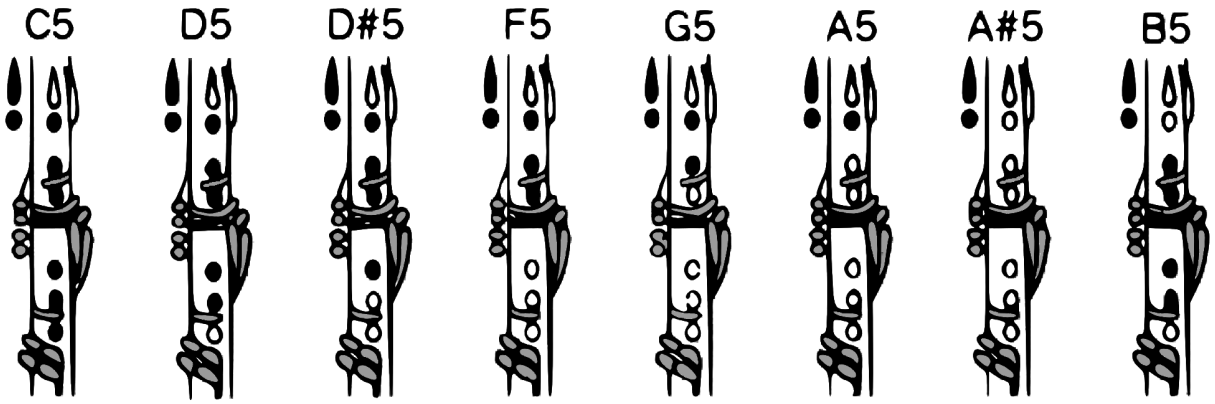


Figure 6: Fingering of selected classes. The two shapes on the left of every clarinet represent the key (top) and the hole (bottom) located in the back part of the instrument. The keys are colored in grey (they are not pressed in none of the eight selected classes) and the holes in black or white, indicating, respectively, if they are closed or not. Figure adapted from http://www.clarinetcloset.com/clarinet-fingeringchart.html

Figure 7 shows the confusion matrix. The main number of confusions are between adjacent classes and this is because they differ in just one hole. The other main confusion is between D5 and B5 that differ in the first hole from top to bottom (see Figure 6). We present in the next section some of the most challenging examples.

For a better comprehension of the problem challenges and the performance of the proposed method, we have prepared a video that shows the main steps of the image processing proposed pipeline, together with some failure cases. The video and source code are available at https://www.upf.edu/web/mdm-dtic/clarinet-iccv17.

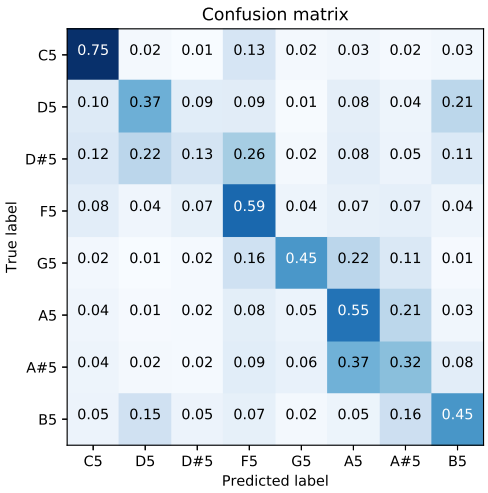


Figure 7: Confusion Matrix.

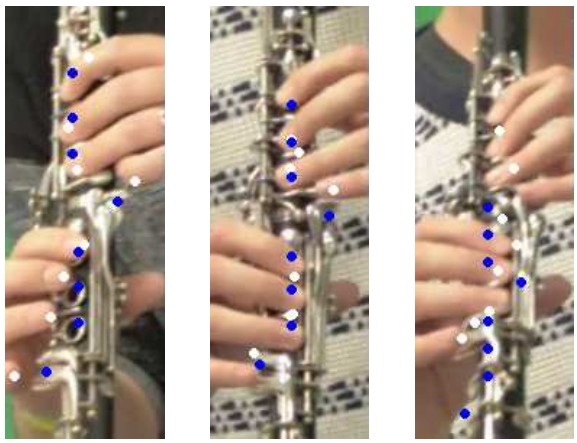
4.1. Analysis of errors

Figure 8 shows three typical failure cases. The first example is when a finger is not covering the hole, but it is very close to it and actually is occluding it. This is a very challenging problem, because we only measure the position of the fingers in the (x, y) plane. In this example the algorithm confuses a F5 note as a D#5 (see Figure 6). Indeed, the use of additional viewpoints or depth information would help in these critical cases. The second example shows how the fingers were wrongly detected, in particular, confusing with parts of the clarinet. The third example shows how the homography could be badly estimated and the positions of holes and keys are wrong. In this last example the fingertips detection also fails.

5. Conclusion and future work

A system for automatic transcription of clarinet using video information is presented in this work. The relative fingertips positions are extracted in order to train a machine learning algorithm. For training the system, the ground-truth notes were automatically extracted and manually corrected.

The studied problem is very challenging, and the system achieves a mean accuracy of 47.2 % using 8 classes. From the obtained results, we see that, although automatic transcription of monophonic recordings is an easy task in the auditory domain, it is very challenging in the visual domain, given the difficulty of estimating small changes in finger positions. In this respect, audio pitch estimation can be exploited to assist visual models in case of missing audio excerpts.



(a) Confusion F5 as D# 5 (b) Detection errors of fingertips (c) Detection errors of holes/keys

Figure 8: Three cases of errors. Blue points indicate holes and keys positions and white points indicate fingertip detection.

Since the approach of this work is tailored to the particular structure of the clarinet, as future work we want to investigate the possibility of extending and generalizing this technique to different instruments. We are also interested on the design of a neural network that could extract more robust features.

6. Acknowledgements

We are grateful to Alessio Bazzica and co-authors of the Clarinetists for Science (C4S) project [1] for sharing their dataset and to Olga Slizovskaia for the initial idea of this research task. This work is partly supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502); the Spanish research projects CASAS (TIN2015-70816-R, MINECO/FEDER, UE) and MOTECAVI (TIN2015-70410-C2-1-R, MINECO/FEDER, UE); BPIFrance and Région Île de France in the framework of the FUI 18 Plein Phare project; the Office of Naval research by grant N00014-17-1-2552, ANR-DGA project ANR-12-ASTR-0035; and ANR-14-CE27-001 (MIRIAM).

References

[1] A. Bazzica, J. C. van Gemert, C. C. Liem, and A. Hanjalic. Vision-based detection of acoustic timed events: a case study on clarinet note onsets. In *International Workshop on Deep Learning for Music (DLM) in conjunction with the International Joint Conference on Neural Networks (IJCNN)*, Anchorage, Alaska (USA), 2017. 2, 3, 4, 5, 7

[2] C. Cannam, M. O. Jewell, C. Rhodes, M. Sandler, and M. d'Inverno. Linked data and you: Bringing music research software into the semantic web. *Journal of New Music Research*, 39(4):313–325, 2010. 2

[3] C. Cannam, C. Landone, and M. Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the ACM Multimedia 2010 International Conference*, pages 1467–1468, Firenze, Italy, October 2010. 2

[4] D. Chai and K. N. Ngan. Face segmentation using skin-color map in videophone applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:551–564, January 1999. 3

[5] A. Cheddad, J. Condell, K. Curran, and P. M. Kevitt. Statistical color models with application to skin detection. In *16th IEEE International Conference on Image Processing (ICIP)*, pages 497–500, 2009. 3

[6] A. J. Danker and A. Rosenfeld. Blob detection by relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3:79–92, January 1981. 3

[7] S. Essid and G. Richard. *Multimodal Music Processing*, chapter Fusion of Multimodal Information in Music Content Analysis, pages 37–52. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012. 1

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981. 5

[9] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001. 3

[10] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5

[11] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 274–280, 1999. 3

[12] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45, 1960. 3

[13] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*. Springer, New York, 2006. 1

[14] H. Kong, H. C. Akakin, and S. E. Sarma. A generalized laplacian of gaussian filter for blob detection and its applications. *IEEE Transactions on Cybernetics*, 43:1719–1733, December 2013. 3

[15] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, 1999. 5

[16] M. Mauch and S. Dixon. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, 2014. in press. 2

[17] J. Montenegro, W. Gomez, and P. Sanchez-Orellana. A comparative study of color spaces in skin-based face segmentation. In *10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pages 313–317, 2013. 3

[18] H. Nisar, Y. K. Ch'ng, T. Y. Chew, V. V. Yap, K. H. Yeap, and J. J. Tang. A color space study for skin lesion segmentation.

In *IEEE International Conference on Circuits and Systems (ICCAS)*, pages 172–176, 2013. 3

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 5

[20] S. L. Phung, A. Bouzerdoum, and D. Chai. Skin segmentation using color pixel classification: analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:148–154, January 2005. 3

[21] R. F. Rahmat, T. Chairunnisa, D. Gunawan, and O. S. Sitompul. Skin color segmentation using multi-color space threshold. In *3rd International Conference on Computer and Information Sciences (ICCOINS)*, pages 391–396, 2016. 3

[22] F. Saxen and A. Al-Hamadi. Color-based skin segmentation: An evaluation of the state of the art. In *IEEE International Conference on Image Processing (ICIP)*, pages 4467–4471, 2014. 3

[23] A.-S. Ungureanu, H. Javidnia, C. Costache, and P. Corcoran. A review and comparative study of skin segmentation techniques for handheld imaging devices. In *IEEE International Conference on Consumer Electronics (ICCE)*, pages 530–531, 2016. 3