INF573

–

IMAGE ANALYSIS AND COMPUTER VISION

–

DETECTION OF PIANO KEYS PRESSED IN A VIDEO

November - December 2022

Aude Bouillé - Long Vân Tran Ha

CONTENTS

INTRODUCTION

The goal of this project is to detect the notes played on the piano in a video, and their concordance with the sheet music followed by the pianist. We are working with videos taken from above the keyboard, well centered and of good spatial and temporal resolutions. The video we have finally taken as our main material, which shows the sheet music on its upper part and the keyboard on the lower part, is in 720P with nearly 30 frames per second.

We can imagine many applications to a more advanced development of this project. For example, we could offer learning assistance to piano beginners, detecting their mistakes and providing them with statistics, such as the percentage of errors on a certain passage. It could also be the automatic transcription on sheet music of an improvisation. The reasonable framing and quality requirements for videos could allow almost everyone to take their own and thus benefit from such applications.

Apart from these practical aspects, we are both music and especially piano enthusiasts, and appreciate the enhancement computer sciences can bring to art, as well as the help to better appropriate it, and this was an additional motivation for the subject of this project.

1 SOURCES OF INSPIRATION

The paper [1] gave us a good basis : we reused several image processing techniques that include Hough transform, morphological dilation and erosion, and region labeling, and we were warned about one of the main difficulties encountered when it comes to key pressed detection, namely the imprecise determination of the hands' shape, as the shade has the same color as the hands themselves.

The suggestions of our teaching coordinators and our own ideas prevented us from having to look for other papers, especially since we wanted to do as much as possible on our own. We used exclusively OpenCV, NumPy and Matplotlib libraries, and created various scripts corresponding to the successive steps we found.

Regarding the content of the project itself, we initially thought only of detecting notes on a keyboard. However, the videos we were using contained the sheet music, and that's when the idea of playing the sheet music as well was born.

2 METHOD

Since we aim at detecting the piano keys pressed by a pianist following a music score, we need to:

1. Determine and label the keys regions:
   We have to know precisely which pixels of the images correspond to which note of the keyboard,
2. Read the sheet music:
   We should be able to follow the sheet music, that is, knowing the notes and in the right order,
3. Detect the keys pressed:
   We need to determine what are the regions of the keyboard the pianist is pressing on.
4. Combine everything:
   We produce an output video similar to the input one, but enhanced by our detection.

We will explain in greater detail each of those steps in the following sections.

2.1 Piano keys

In order to determine the areas corresponding to the white keys and black keys, and to associate them to their respective names, we successively:

1. Apply Sobel to detect edges, using $\sqrt{\text{sobel}_x^2 + \text{solbel}_y^2}$ with sobel_x and sobel_y the gradients following the two axis, and then changing pixels for complete white or complete black, following whether they are above the threshold or not,
2. Get the connected components of the previously obtained gray image, and remove the ones that are either too small or too big, so that each component corresponds to a key,
3. Reassign the labels of the connected components in the right order using the x coordinates of the centroids of the connected components,
4. Erase the noise in the black keys, by finding the convex hull of each black key and filling it with white, as the black keys are convex,
5. Detect the groups of black keys according to their spacing, to know where the music scales begin on the keyboard,
6. Finally name the notes, assuming that the 4^{th} octave is in the center of the keyboard.

2.2 Sheet music

We tackle two tasks linked to the sheet music:

1. Read the notes,
2. Highlight in tempo the notes obtained.

2.2.1 • Read notes on staff

To determine the notes on the music score, we have to:

1. Find the staff,
2. Get the circles corresponding to the notes,
3. Attribute the correct names.

Find the staff To find the staff, we:

1. Erode strongly horizontally, in order to remove all the notes, vertical lines etc, while keeping the horizontal lines of the staff,
2. Calculate the mean of the color channels for each ordinate, and select the ordinates with very low means, that correspond to the black staff contrasting with the white empty space,
3. Group the lines in groups of five, with some calculations, to separate the different musical staffs of the sheet music.

Get the circles corresponding to the notes We proceed slightly differently for the black notes than for the white notes, but in both cases, we need to:

1. Find at least all the circles corresponding to the notes, as we can remove false positives, but should not miss any note,
2. Filter out the wrong ones that might have been detected because of vertical lines or edges of notes creating shapes that can be mistaken for circles,
3. Refine the center, which is useful to determine the name of the note, and thus should not be closer to a wrong line or interline of the staff than the actual one.

For the black notes, we:

1. Erode the image, in order to remove the thin lines of the staff and the white notes (interestingly, we need to erode more to make it work at the best for the computer than what seems more appropriate for our human eyes),
2. Dilate, to make what remains of the black notes bigger, and ease its detection,
3. Get the circles with HoughCircles, and keep only the ones after the actual start of the music score, as we do not want the treble clef or the bass clef to be wrongly considered as black notes.
4. To refine the center, we:
   (a) Calculate the mean of the pixels on a small window around the first obtained center,
   (b) Select for the new center the pixel with the maximum mean.

For the white notes, we:

1. Inverse the gray image (black becomes white and conversely), as HoughCircles detect the white circles better,
2. Get the circles with HoughCircles, and keep only the ones after the actual start of the music score,
3. Keep only the ones not to close to black notes (this removes the false white notes detected between a vertical line of the score and a black note).
4. To refine the center, we:
   (a) Get the connected connected components of the image, and keep only the small enough ones, in order to eliminate the components linked for example to the space between the lines of the staff,
   (b) Group the very close ones (to solve the case of a white note separated by a line of the staff),
   (c) Choose the center, with right weights in case of two components.

Attribute the correct names We name each note found with the name of the note known from the staff which center is the closest to the center of the note.

2.2.2 • Highlight the notes on tempo

In order to highlight the notes in tempo, on the music score as well as on the keyboard, we:

1. Find the bar lines by eroding strongly vertically and then applying HoughLines,
2. Do some calculation to determine to which beat of the measure each note belongs,
3. Color successively each beat of the music score, and the corresponding notes on the keyboard.

2.3 Hands and keys pressed

In order to determine the keys pressed by the hands, we:

1. Get the hands region, by thresholding the standard deviation on the RGB channels, which is much higher for the beige of the hands and their shade than for the black and for the white of the keyboard,
2. Calculate for each frame for each note the standard deviation of the region above the hands region (and above the white keys for the black keys),
3. Mark as pressed the keys that have a mean of standard deviation over a small number of frames above some threshold.

We has also tried some other approaches for the finger detection (see A).

2.4 Combination and final output

We combine the reading of the sheet music with the keys detected on the keyboard. We highlight successively each beat on the sheet music and some keys on the keyboard: these keys appear in green when the keys pressed detected and the awaited keys concur, in blue when keys are awaited from the sheet music but not detected as pressed, and in red when keys pressed are detected but not awaited from the sheet music. Note the following details:

- We do not recognize which frame corresponded to which beat based on visual input, we simply divide equally the video from the start of the music to its end according to the number of beats in the sheet music,
- We apply a kind of green color extension rule: if a key that should initially have been colored in red or blue is separated by only some frames of the same key colored in green, we color it in green (this avoids having for a few fractions of a second a flash of color with little meaning, as a pianist is not necessarily precise to the nearest tenth of a second on the pressing of a key),
- We do not color the part of the key that correspond to the slightly dilated mask of the hands: indeed, the result is otherwise disturbing, as our human brains know very well that the keys are located behind the hands. Not coloring above the hands seems thus more natural, as if the key itself was colored in real life.

3 RESULTS

We have obtained very good results for most of the duration of the small piece we worked on, "Scarborough Fair". Only a tiny minority of notes is not detected, and there are only two passages where too many notes are detected. In both passages corresponding to rests, the pianist moves his right hand higher above the keyboard, which causes the detection of the notes played to run riot.

CONCLUSION AND FUTURE WORK

We reached our goal of detecting in a video the piano keys pressed by a pianist following a music score, by using a variety of computer vision concepts, such as applying filters, converting an image to gray-scale or applying Sobel transform, eroding, dilating, detecting lines and circles, getting connected components and convex hulls, working on temporal or spatial variations, and so on.

As for future work, we could try to obtain cleaner masks for the hands, and better shapes and fillings for the keys. Furthermore, a lot of parameters with numerical values in our code are tuned for the video we worked on, and we could now try to make the whole thing more flexible and adjustable to new inputs. Additional work would also be needed for tackling more complicated pieces, containing other types of notes such as eighth notes for example, or involving a lot of hand gestures. Besides, as our algorithm are quite sensible to noise and to video quality, we would probably face some problems if we were to take very amateur videos as inputs.

REFERENCE

[1] Potcharapol Suteparuk. 2006. *Detection of Piano Keys Pressed in Video*. stacks.stanford.edu https://www.semanticscholar.org/paper/Detection-of-Piano-Keys-Pressed-in-Video-Suteparuk/75d75a3fe7796fe9b3003ea73252de58d93d60dc

A EXTRA EXPLANATIONS

On finger detection

To get the hand regions, we had first tried to use color conversion to see the difference between the hands and the piano, but this did not give good enough results. Then, we tried to use a lower bound and an upper bound for the ratio between R and G, but our third method, using the standard deviation of the color, since the piano is black or white while the hands are beige (different values of color channels) different, performed better, so we chose this last method.
The main problem concerning the hands is that the shade of the hands on the keyboard is almost of the same color as the hands themselves. It creates imprecise and rough hands mask, with fingers almost not distinguishable. To solve this problem, we had tried to detect fingertips, by using the HoughCircles function, and keeping only the circles whose center was in the rough hand region and not too far from the rest of the space. However, the result was extremely unstable: we had sometimes twelve fingers, or only seven, and we could not have imposed ten circles, since some fingers are really hidden at time by other parts of the hands.

B FIGURES

B.1 Piano keyboard

Figure 1: Original keyboard

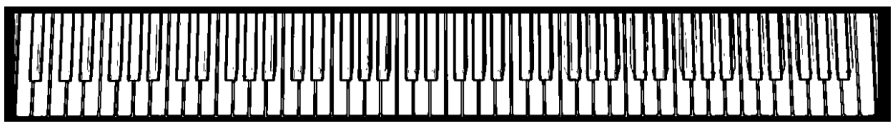

Figure 2: Sobel transform

Figure 3: Detection of key regions

Figure 4: Reassignment of region labels

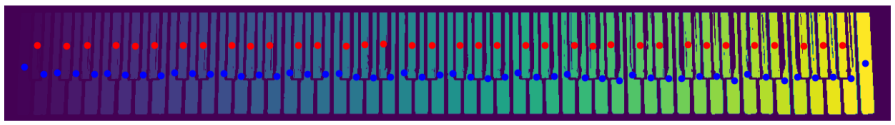

Figure 5: Determination of centroids

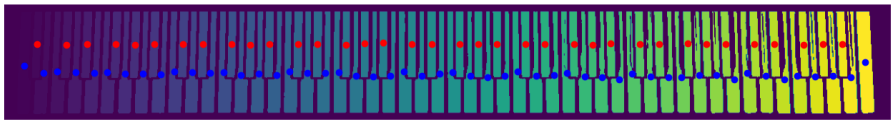

Figure 6: Filling of convex hulls of black keys

Figure 7: Detection of groups of black keys

Figure 8: Keys named

B.2 Sheet music

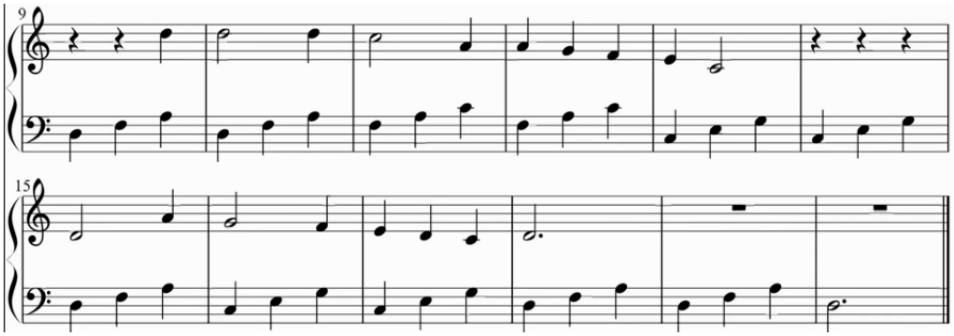

Figure 9: Original sheet

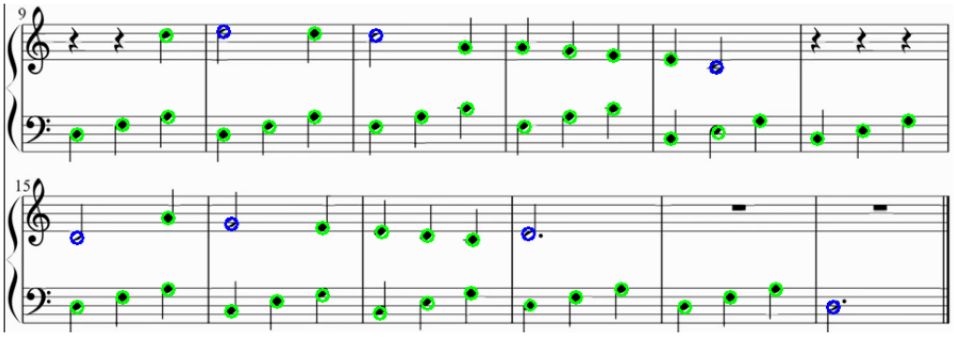

Figure 10: Notes obtained

B.2.1 • Staff

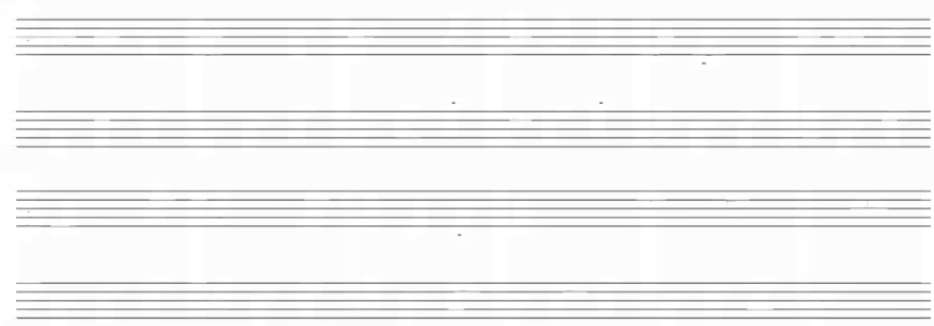

Figure 11: Strong horizontal erosion

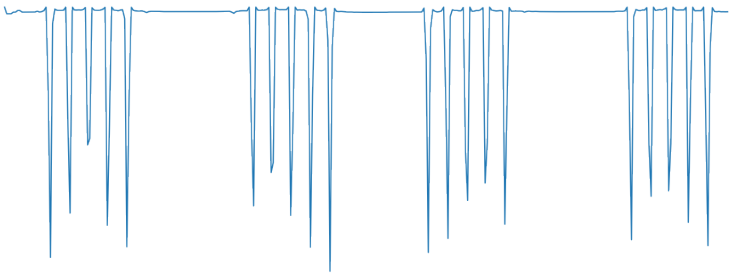

Figure 12: Mean depending on the ordinates

B.2.2 • Black notes

Figure 13: Strong erosion

Figure 14: Dilation

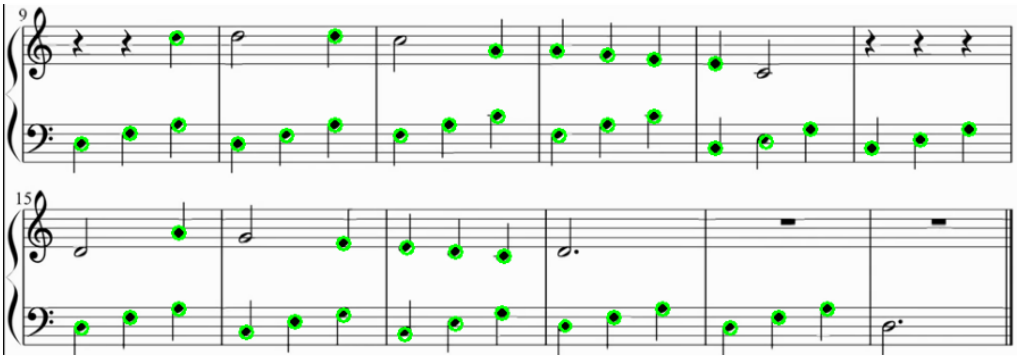

Figure 15: Circles obtained

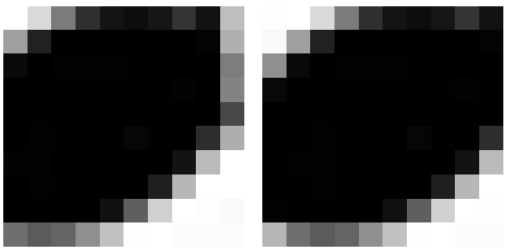

Figure 16: Refinement of the center (old at left, new at right)

B.2.3 • White notes

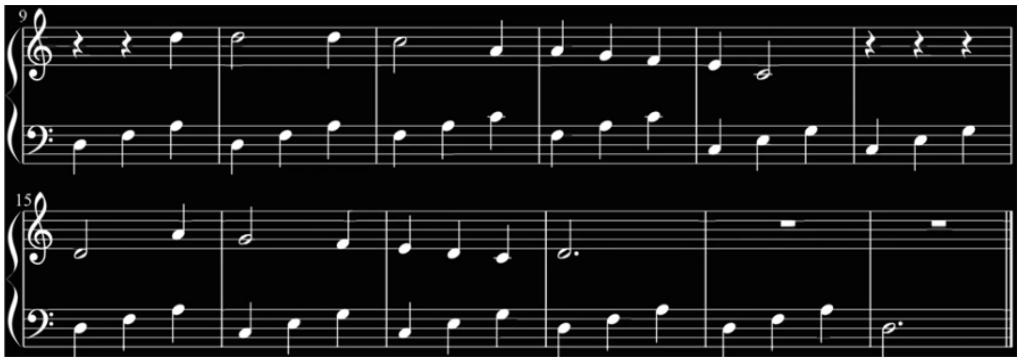

Figure 17: Inversion of gray image

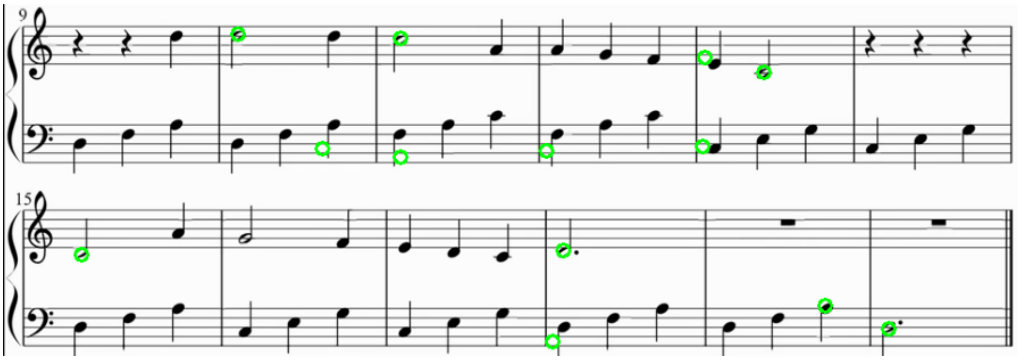

Figure 18: Circles obtained

Figure 19: Determination of small components

B.2.4 • Highlight on tempo

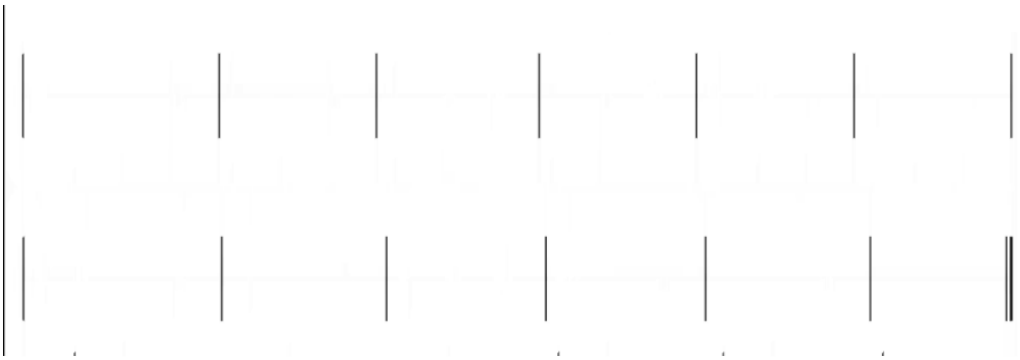

Figure 20: Strong vertical erosion

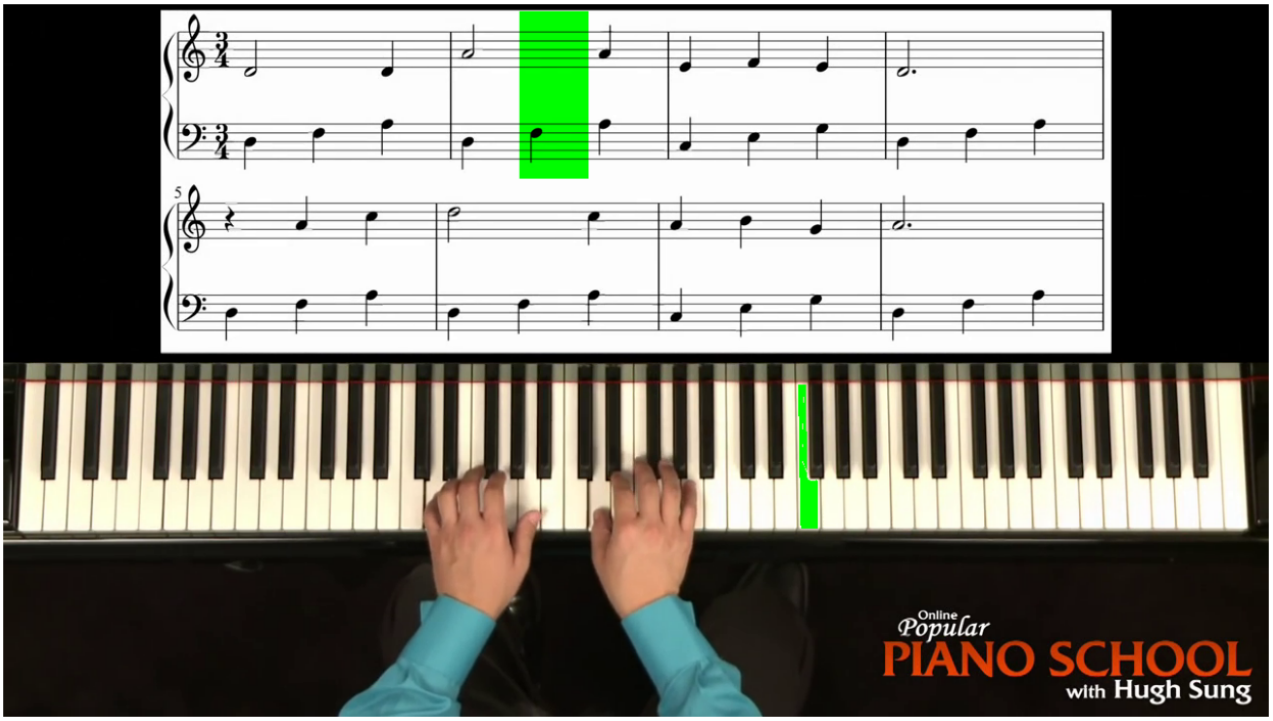

Figure 21: Highlight of a beat on the sheet and on the keyboard

B.3 Hands and keys pressed

Figure 22: Hands region

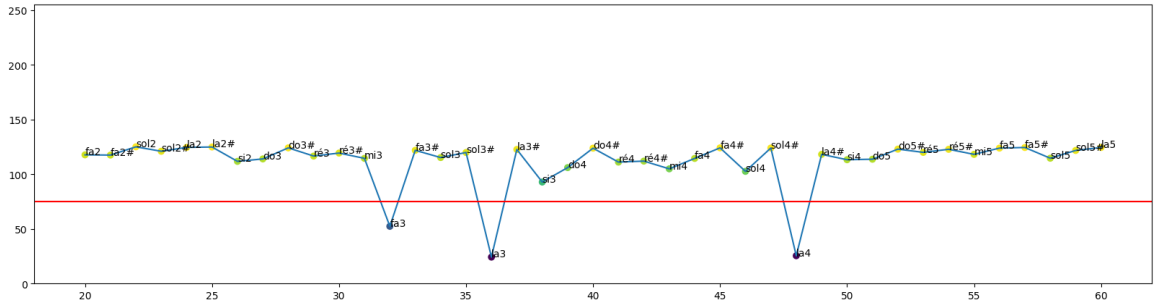

Figure 23: Study of the standard deviation

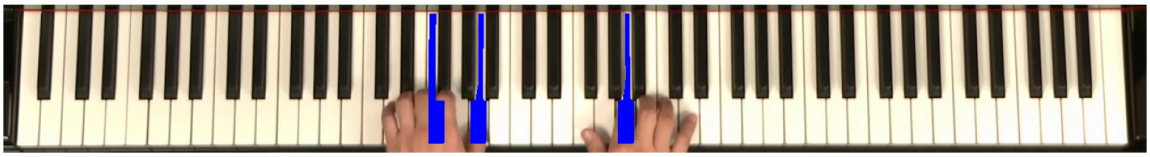

Figure 24: Corresponding keys pressed

B.4 Combination

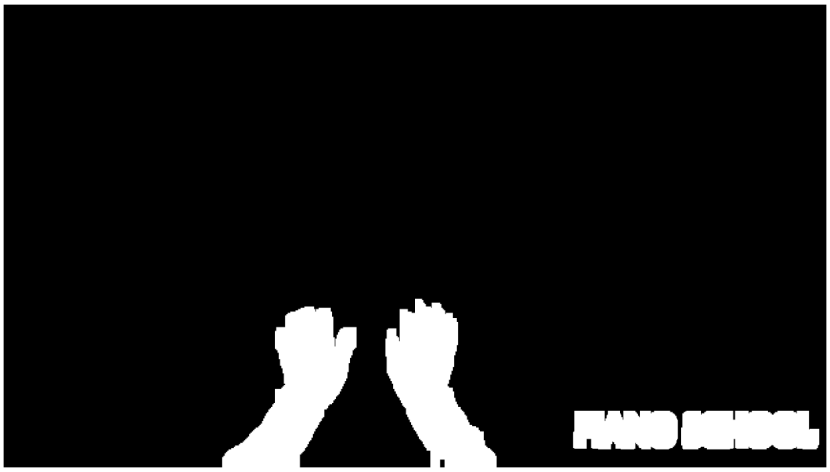

Figure 25: Dilated mask of the hands

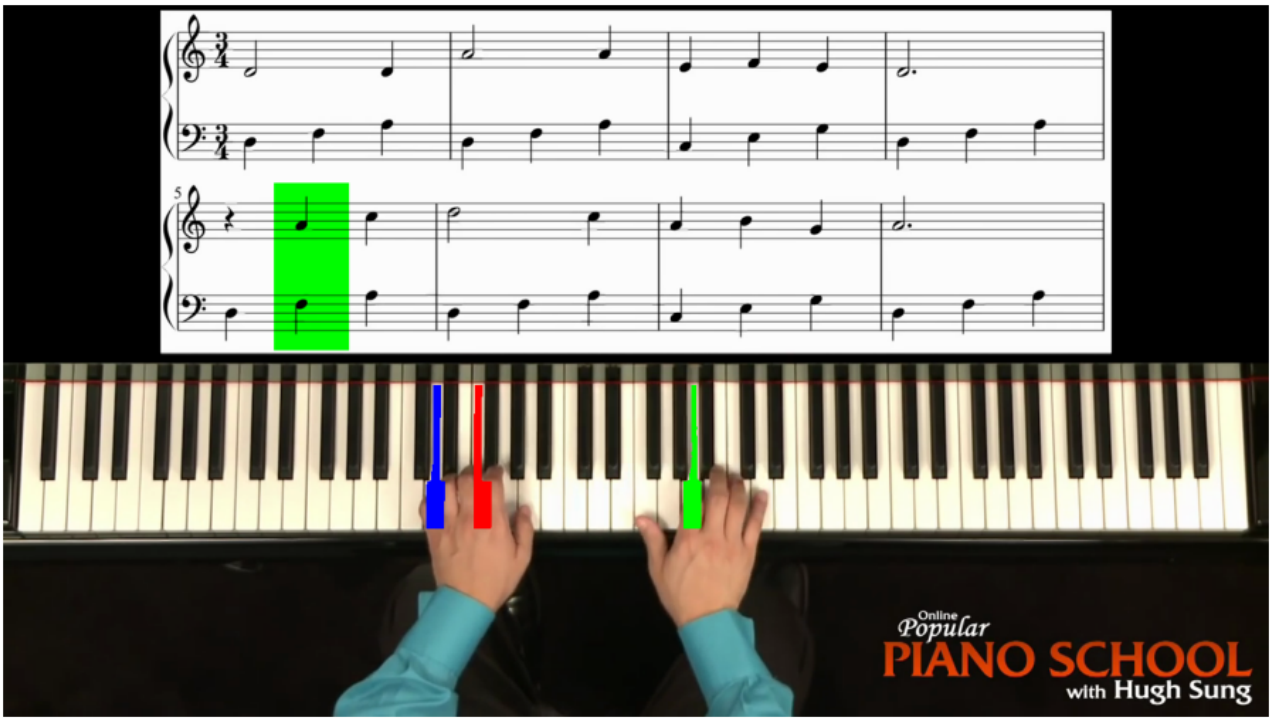

Figure 26: Combination with overlap

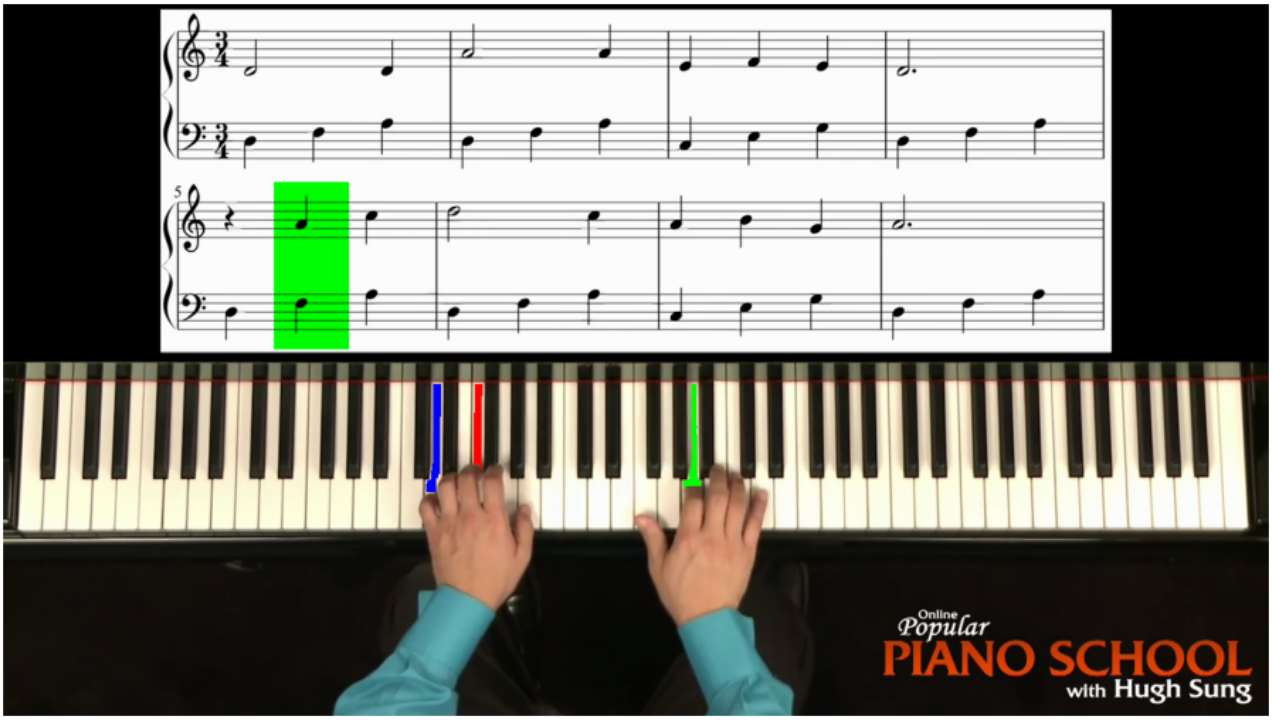

Figure 27: Final combination