Detection and Tracking of Pianist Hands and Fingers

Gorodnichy, Dimitry; Yogeswaran, A.

Third Canadian Conference on Computer and Robot Vision (CRV 2006) [Proceedings], 2006

National Research
Council Canada

Institute for
Information Technology

Conseil national
de recherches Canada

Institut de technologie
de l'information

NRC · CNRC

*Detection and Tracking of Pianist Hands and Fingers **

Gorodnichy, D., and Yogeswaran, A.
June 2006

* published at the Third Canadian Conference on Computer and Robot Vision (CRV 2006). June 7-9, 2006. Québec, Québec, Canada. NRC 48492.

Canada

Detection and tracking of pianist hands and fingers

Dmitry O. Gorodnichy[1] and Arjun Yogeswaran[2]
[1] Computational Video Group, IIT-ITI, NRC-CNRC
[2] Computer Engineering Department, University of Ottawa
http://synapse.vit.iit.nrc.ca/piano

Abstract

Current MIDI recording and transmitting technology allows teachers to teach piano playing remotely (or off-line): a teacher plays a MIDI-keyboard at one place and a student observes the played piano keys on another MIDI-keyboard at another place. What this technology does not allow is to see how the piano keys are played, namely: which hand and finger was used to play a key. In this paper we present a video recognition tool that makes it possible to provide this information. A video-camera is mounted on top of the piano keyboard and video recognition techniques are then used to calibrate piano image with MIDI sound, then to detect and track pianist hands and then to annotate the fingers that play the piano. The result of the obtained video annotation of piano playing can then be shown on a computer screen for further perusal by a piano teacher or a student.

1. Introduction

1.1. Video recognition for piano playing: new application

Current music recording and transmitting technology allows teachers to teach piano remotely. This is in many cases the only way to teach music, especially in rural or distant areas where the ratio of piano teachers to piano students is extremely low [4]. MIDI recording technology allows a teacher to play a piano at one place and to see a piano played by itself, as by an "invisible teacher", at another place (see Figure 1): the piano keys are pressed exactly at the same place, velocity and duration on a remote piano [1]. However, to know *how* these keys were played by a teacher remains unknown. This includes the knowledge of which hand played a key, which finger was used, and who (in case of a four hand musical piece) was playing. With the current advances in computer vision and video recognition, some of this knowledge can now be also transmitted.

This paper describes a video recognition tool called *C-MIDI* that allows one to detect pianist hands and fingers using a video camera mounted on top of the piano keyboard (see Figure 2). By synchronizing video data with MIDI data, the program is able to annotate MIDI events according to the visual labels $Hand = \{left,\ right\}$ and *Finger*=$\{1,2,3,4,5\}$. These video-annotated MIDI events can then be stored or sent to a remote server where they can be played at the same time with playing an annotated video, as shown in Figure 1.

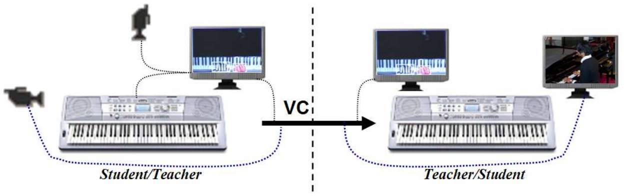


Figure 1. Video-conferencing (VC) for distant piano learning. A conventional session includes the transmission of a video image only (thick line). Video recognition technology allows one to transmit also the annotated video image (thin line).

In addition to distant and offline learning, video annotation of pianist hands and fingers has also a few other applications important for piano teaching. For example, it can be used for storing detailed information regarding music pieces for a searchable database (such as in [4]). It can facilitate producing music sheets. It can also be used for score driven synthetic hand/finger motion generation (as in [9]).

1.2. Piano playing for video recognition: new testbed

While the utility of the pianist hand detection problem for piano performers and teachers is clear, we also would like to demonstrate here that this problem is also of great utility for the computer vision community.

Recognition of hands and fingers using video, which is a very challenging video recognition problem, has been considered so far in the context of such applications as computer-human interaction [14, 15, 11], automatic sign language recognition [15, 12], robotic hand posture learning [10], and multimedia [13]. In all of these applications, the motion of the hand and fingers is limited to a predefined number of states, which often constitute a hand/finger gesture vocabulary that a computer vision system attempts to identify.

Furthermore, in all of these applications hands and fingers are manipulated by humans *in order to be detected*, i.e. they are used to send *visual* commands or signs to either a computer or a human. Because of that the set of possible hand and finger configurations is such that it makes them easier to be *visually* distinguished from one another. In particular, fingers would be normally well visible to a viewer (or a camera), well protruded from the center of the palm when possible, which would normally be made use of by conventional finger detection algorithms [14, 15, 11, 15, 12, 10, 13].

In the case of detecting pianist fingers playing piano, the situation is very different. Pianists use hands to play music and therefore put all their attention on the *acoustic quality* that the motion of their hands produce, rather than on how they visually appear to a viewer. Therefore, pianist hand/finger motion can be considered as an example of non-collaborative and unbiased visual data, which can be used as a unique testbed for hand/finger recognition algorithms. At the same time, pianist hand/finger motion presents a wide range of challenging computer vision problems, the most challenging of which are tracking of highly deformable and flexible 3D objects (since pianist hands and fingers are extremely flexible and fast) and multiple object tracking (since hands and fingers may occlude each other and disappear).

This paper addresses three video-recognition problems which need to be resolved in the context of piano playing annotation and which are a) piano keyboard recognition, b) hand recognition, and c) finger recognition. The organization of the paper is the following. First, we describe the setup developed for the project and outline the challenges we are facing when using this setup (Section 2). The we provide a general outlook of video recognition approaches to be uses (Section 3) and propose solutions to each mentioned video-recognition problem (Sections 4–6). The results of live video annotation of professional pianist playing are demonstrated in Section 7. Future work concludes the paper.

2. System overview and the issues

The setup that was developed for tracking pianist hands and fingers using a video camera is shown in Figure 2. A video camera is mounted on a tripod above the piano keyboard with its field of view covering four octaves where the music piece will be played. When a pianist plays, the camera observes his/her hands and sends the video data to a computer to be processed in real-time and displayed back on a computer screen along with the annotation.

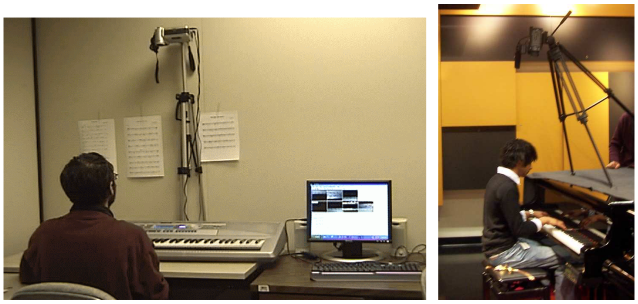


Figure 2. Setup for tracking pianist hands and fingers using a video camera: in home environment with Yamaha MIDI-keyboard (left image) and in a professional piano studio environment with MIDI-equipped grand piano (right image).

While this setup has been found most suitable and convenient for the task, it still poses several challenges to deal with. First, because of the real-time system performance requirement, video is set to the minimum resolution within which fingers are visible, which is the resolution 160x120 pixels. While indeed clearly visible at this resolution for a human, fingers however are only a few pixels wide, which makes it difficult for a computer to detect them. Second, there are a lot of specular reflections present on the keyboard image and there are shadows often cast on the keyboard from hands, which also creates a problem for hand location. Finally, the main challenge is the complexity of the hand and finger motion, as they constantly change their shape and location, becoming often self-occluded and poorly visible. This can be well seen in Figures 3-6 and the provided video recording.

Figure 3 illustrates the main stages of video-recognition we have to perform. First, the piano keyboard has to be detected and its image rectified, i.e. transformed to a canonical horizontal position. In doing that, the piano keys (such as ”C” etc) have to be recognized and video-sound calibration performed. This constitutes the initialization stage. Then goes the hand detection stage, within which hands are first detected as foreground objects in front of the piano keyboard and then tracked over time using deformable hand templates. The final stage is finger position localization in which the fingers are detected and matched to the piano keys underneath them. Because of low image resolution and a very complex, from a vision perspective, finger positioning – they are never protruded, mostly bent towards the keyboard (i.e. away from camera), often touching and occluding each other, a new finger detection technique is developed. This technique is based on a new edge detection approach called the *crevice detection*. *Crevices* are defined as locations in the image where two convex shapes meet. Since pianist fingers appear convex to the camera, their edges can thus be detected as crevices.

Details on the proposed solutions to each of the described tasks will be given later, but prior to that let us summarize the techniques that one should follow when design-

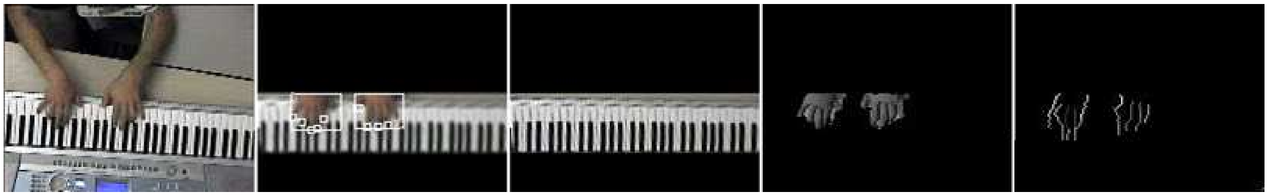


Figure 3. The stages of pianist hand and finger detection: 1) keyboard image rectification, 2) hand tracking, 3) finger location estimation.

ing a computerized video-recognition system.

3. General video recognition rules

Humans easily identify objects and their interrelationships in video. For a computer however video data is nothing but a changing in time matrix of three-dimensional numbers (RGB pixel values). To give a meaning to these data is what makes research in the area of video recognition. While many results have been obtained in this emerging, interdisciplinary and highly demanded area, below we list some of them that we adhere to when designing the solutions to the video recognition tasks of this project.

Colour space. When analyzing colour video images, it is generally a good idea to perform image analysis in a non-linearly transformed colour space (such as HSV or YCrCb) that separates brightness values of the signal from the values related to the signal colour. We process images in the UCS (perceptually uniform colour space), which is a non-linear version of the YCrCb space obtained from the empirical study on psychophysical thresholds of human colour perception [6].

Local importance. While global normalization techniques. such as histogram-based intensity and colour normalization, generally improve video recognition, it is even a better idea to perform local (fovea-driven) image analysis where possible. This not only enhances video information, but also provides a way to filter out noise. The local processing techniques include local intensity normalization, local structure transforms [5] and such popular techniques as median filter and morphological dilation and closing. For the same reason, it is often preferable to analyze gradient images rather than absolute-value images.

Collective decision. It is also preferable to use collective-based decisions rather than pixel-based ones. This includes using support areas for change/motion detection and also using higher-order parametrically represented objects such as lines and rectangles for detection and tracking instead of raw pixels that comprise the objects.

Accumulation over time. Finally, the temporal advantage of video, which allows one to observe the same scenery or object over a period of time, has to be made use of for better detection and tracking of objects.

4. Initialization stage

Prior to the hand and finger detection, the position of the piano keyboard in the image has to be detected. Only the part of the image containing the piano keyboard will be used in further processing[1]. For the purpose of vision-based annotation of the played sounds (MIDI events), the localization of the octaves and the middle C is also required.

4.1. Keyboard image detection and rectification

The keyboard is detected based on the fact that it contains repetitive black keys surrounded by non-black areas[2]. To detect these keys, only the pixels that have both low luminance and chromaticity (less than 70 in UCS space) are highlighted. The obtained binary image is postprocessed by a median filter and morphological dilation and all blobs that do not satisfy the black key proportions ($5<\text{height/width}<30$) are removed from the image. Then the lower and upper tips of each blob are used to find the lines of the best fit. These two lines pass through the vertices of the black keys and their rotation define the rotation of the keyboard with respect to the camera axis. By rotating the video image in the opposite direction by that angle, the position of the keyboard is rectified.

The top and bottom of the keyboard are computed based on the assumption that white keys are surrounded by non-white areas. The rotated video image is traced, starting from the detected black keys in four directions (left, right, up, down) until a drastic intensity change is detected (represented by the luminance and cromaticity). The majority of the detected boundary pixels define the boundary of the keyboard.

Under the assumption that the camera is mounted exactly above the keyboard, which is made the case,[3] the de-

[1] The future work includes using a part of the video image located between the keyboard and a pianist for better tracking of pianist hands.

[2] The assumption is made that the colour of the piano is not black, which was the case in our experiments. If this is not the case, then automatic detection of the keyboard can be replaced by the manual region of interest selection.

[3] For a general case when a video camera is not positioned exactly above the keyboard, the rotation of the image is not sufficient to rectify the piano keyboard image. Instead, a homography operation which transforms four corners of the black keys to a four corners of a desired rectangle has to be performed.

scribed piano rectification approach is almost 100% accurate, except for the situations when the keyboard is rotated more than 30 degrees with respect to the camera horizontal axis. Figure 6 shows the result. The detected black keys are shown in the lower right image circumscribed by white rectangles.

4.2. Detection of the “C" key

After the image has been rectified so that the piano keyboard is shown horizontally positioned on the image (as in Figure 3, middle image), the patterns of the black keys are analyzed to determine the position of the C key. Similarly to the way humans recognize the key, the binarized image containing the blobs corresponding to black keys is examined and two groups consisting of three and two black keys are detected by scanning image from left to right. The left boundary of the two-key group indicates the position of the C key.

This technique alone does not guarantee the detection of the middle C. Therefore, to complete the calibration, a pianist is required to show the middle C key with a finger. If the note s/he plays coincides with the note detected by video, the calibration of the system is considered successful, meaning that the system is ready for video annotation of MIDI played data.

It has to be mentioned at this point that, while having high resolution is not critical for the described piano image rectification and video-MIDI calibration to work, the quality of the video camera is. In particular, it is imperative to make sure that mapping from the observable space (containing the keyboard) to the image space (showing the image of the keyboard) is linear. This implies that the distance between the keys in the middle of the image is the same as on the image boundary and that the keyboard contour is a non-warped rectangle. To achieve this we use a high quality video camera with the zoom functionality [4].

5. Hand detection and tracking

5.1. General object detection rules

Detection of objects in video can be generally performed by one of the two methods: by isolation from a background and by recognizing the object dominant features.

The first method is usually employed when the object features are not known in advance (e.g. when the objects is of unknown colour or shape) and/or when the camera is stationary with respect to the background. This is the method that is most commonly used for surveillance applications [7]. The preferable techniques for this method include non-linear change detection techniques [6], which consider a pixel changed based on the patch around the pixel rather than the pixel itself, and statistical background modeling (such as the Mixtures of Gaussians technique) that learns the values of the background pixels over time.

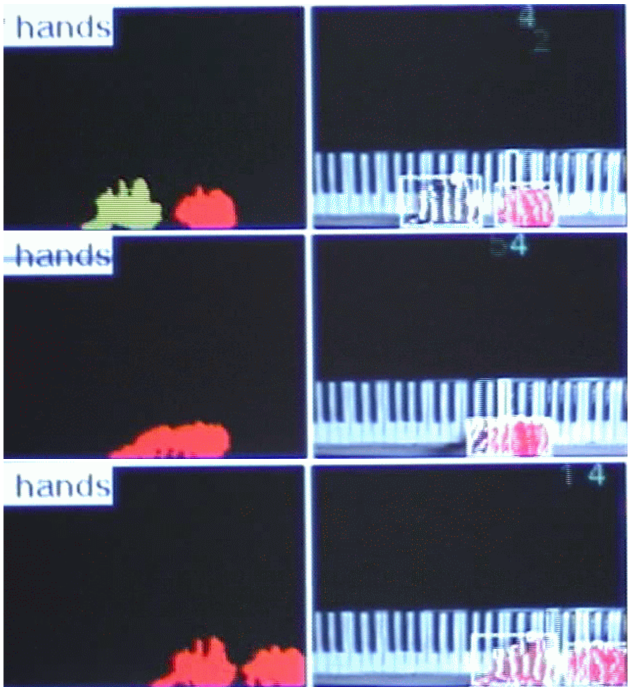


Figure 4. Foreground detection extracts blobs corresponding to the hand images (left column), while hand template tracking allows one to detect partially occluded hands (right column).

The second method is used when object features, the most descriptive of which are colour, shape, and texture, are known or when the camera-scenery setup is such that makes background/foreground computation impossible. Face and skin detection is most commonly performed by this method [5]. With respect to the skin detection, which can be used for detecting pianist hands, the following techniques are worth mentioning: skin colour detection with edge merging techniques [15, 13] and the recent work [2] based on hysteresis-like colour segmentation of skin colour.

After an object has been detected in a video frame, there is usually no need to scan the entire video image when searching for the object in the consecutive video frames. Instead, the past information about the object location and size is used to narrow the area of search and to detect the object. This is what defines object tracking. Besides speeding up the detection of objects, tracking makes it also possible to detect occluded and partially disappearing objects.

5.2. Hand detection for hand template initialization

For the detection of pianist hands using the setup described above, either background or colour-based detection

[4] If eye-fish lens cameras such as off-the-shelf web-cams are used, then the second-order dewarping operation has to be performed prior to image processing.

method can be used. We have chosen the fastest of them, which is the background subtraction method.

A simplified faster version of the Mixtures of Gaussians technique is used to compute the background image of the piano. As the playing session starts, the statistical information about the background is computed in terms of its running average I_{BG} and deviation D_{BG}. The background image is then constantly updated in all pixels where no motion is observed. The motion is observed where the change image dI, computed as the diluted sum of the differences between the last three consecutive frames, is larger than a threshold. Since pianist hands are barely static, this provides a good way of not counting the hands as part of the background. The foreground image I_{FG}, which is the image of the hands, is then computed as the part of the image where the difference from the background image is at least twice as high as the background deviation: $I_{FG} = |I - I_{BG}| > 2 * D_{BG}$.

When foreground containing hands is detected, its colour can be learnt by updating the 2D histogram that counts the values of the Cr and Cb components of the foreground in the YCrCb colour space. This histogram can then be backprojected to the image at any time to find the areas containing hands. Within our setup however there is no need to usie this technique, since foreground detection alone is sufficient for the purpose. Typical hand detection result is shown in Figure 4.

In order to commence tracking of the hands, we need first to initialize the hand templates. In the context of our application, hand template is defined as a box circumscribing the hand. The size (width, height) and location of the box define the template parameters. When a hand is localized, the hand template parameters are adjusted to reflect the hand deformation.

In order to find the number of hands to be tracked, which can range from one to four, the pianist(s) is/are required to show non-occluded hands on top of the keyboard at the beginning of the play (as on the top image in Figure 4). The adaptive K-means (with K=1,2,3,and 4) is then applied to the foreground image in order to find the number of hand blobs present.

5.3. Hand tracking using deformable templates

When hand templates are initialized, hand tracking continues as follows. The foreground image obtained as described above serves as a guide to detect hands by means of finding the best fit between the updated hand templates and the blobs in the foreground image [5]. Only gradual change in the hand template is allowed between the consecutive frames. In particular, hand box parameters such as velocity, location and size are allowed to change not more than by 5%. Such a gradual transition of the hand template not only makes it possible to track highly deformable objects such as hands, but also allows one to track hands that momentarily overlap, which is a frequent case in piano playing.

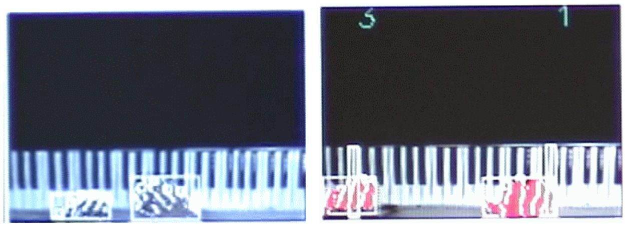


Figure 5. Finger edges are detected in the hand regions using the crevice detection operator. When a MIDI signal is received, meaning that a piano key was pressed, the hand and finger which are believed to press the piano key are shown: hand is highlighted in red, the finger number is shown on top of the image. See also Figure 4.

Figures 3-6 show the results of hand tracking (as white boxes circumscribed around the hands). The experiments show that the described technique is sufficient for not losing the overlapping hands, provided that after the occlusion the shapes and the location of the hands do not change much. Another assumption about the hand motion is that the hands are assumed not to crossover each other. Under these assumptions, the described hand detection technique proves sufficient for the next stage, which is finger detection.

6. Finger detection

Present solutions to the problem of finger detection, which is closely tied to the problem of hand detection and tracking, include cylinder-based 3D model fitting [3], correlation with predefined templates [10, 14], image-division-based decision tree recognition [12], skin colour detection with edge merging techniques [15, 13], and recent work [2] based on the hysteresis-type colour segmentation of skin colour. As already mentioned earlier, the main feature of the referred works is that they deal with hands with well visible fingers, which in most cases very well protrude from the center of the palm, and which serves as the main cue in detecting the fingers.

In the case of detecting fingers of a piano player, the situation is very different. As already mentioned, the pianist fingers are *never* protruded. Furthermore, they are mostly bend towards the keyboard, i.e. away from camera, often touching and occluding each other. For this reason, we propose another technique for detection which can deal with non-protruding fingers, which are possibly tightly grouped together.

What makes our technique preferable to other techniques for the current application is that it makes use of the a-priory knowledge about the configuration of pianist hands with re-

[5] Another approach to guide tracking is to backproject histogram containing the skin colour information.

spect to the camera view. It does not require high-resolution images, which is often a requirement for other techniques and is robust to illumination and skin colour changes. The program developed using the described hand-finger detection runs in real-time, which makes it possible to use it to annotate piano playing live.

After the rectangular areas corresponding to the detected hands are detected in the hand tracking stage, they are examined for the presence of what we call *crevices*, which are defined as the locations in the image where two convex shapes meet. Most pixels detected as crevices are disconnected. Therefore the post-processing techniques follow that try either to connect them into continuining lines of finger edges or segment the blobs surrounded by these pixels as corresponding to different fingers. Thus detected finger blobs and finger edges are traced to the upmost point in the image, which corresponds to a potential point of contact with a keyboard.

6.1. Crevice detection operator

The problem with conventional gradient change based edge detection techniques, such as Sobel's gradient change based, Canny's hysterethis based, or Harris gradient orientation based techniques, is that they either detect to too many pixels in the hand area, or a too small number of them. To circumvent this problem, we make use of the observation that finger edges are, in fact, of a very specific type. They are the edges of convex objects – fingers. Furthermore, in the case of piano playing, fingers are most frequently touching each other or are on top of each other, which makes them look like crevices on the landscape of the hand surface.

We also note that, as opposed to a piano keyboard which shows a lot of specular reflection, hand fingers look very much like Lambertian surfaces with constant albedo. Therefore (e.g. see [8]) finger edges can be detected by an operator which searches the part in the image, where intensity goes down (becomes consistently darker) and then up again (becomes consistently lighter).

With respect to the fingers playing a piano, fingers have a dominant vertical orientation. This allows us to implement the crevice detection operation using the following procedure. The algorithm goes horizontally from one starting point to another and marks all locations where the intensity gradually darkens until it lightens again. In doing so the algorithm allows the intensity values to change within a certain threshold level, until they consistently decrease. When after the decrease, the intensity values start showing a consistent increase, aa pixel is marked as a point on a crevice.

This method is ideal for detecting the end of one convex object and the beginning of the next convex object, as in the case of fingers. It does not limit the exact pixel width of the edge. It allows the edge to be as wide as necessary, and will only detect the breaks between fingers instead of the edges of the finger itself.

6.2. Postprocessing

Since the crevice detection algorithm is applied separately on each successive horizontal row, it is possible that the marked pixels may not all match up and be perfect. When plotted, thus detected edges will be seen as discontinued lines. Therefore, the detected pixels have to be further processed to be recognized. Two methods for this are considered.

The first method is based on connecting the existing line fragments, thereby detecting the edges of the fingers. Morphology skinning techniques based on dilation and erosion caused problems when the pixels were in close proximity to each other. The Hough Transform required that the lines be piece-wise straight. Therefore another method is proposed that connects the lines that fell within a certain closeness to being on the line. All connected pixels are stored as probable edges. There are usually more lines than fingers, due to errors in finger detection. These extra lines must be connected together to generate the most probable lines of the fingers. The position and slope of each line fragment is matched to all other line within proximity that would fall along a similar slope If the match score is high enough, within a certain error threshold margin, the lines are connected into one. After applying this method to all detected lines, newly created finger lines are obtained, refining the image with better detection of the fingers. The result of this method is shown Figure 3.

The second method connects two similar successive lines to form an interpolation of a finger shape. Based on the raw crevice detection image, and using the left-most crevice pixels in each row as a starting point, we proceed to the right until we find the next crevice pixel. The distance between the left-most pixel and the next pixel to the right, is stored for each row. It is sorted to find the median, and that is used to limit the different widths. Then the best possible blob is created by using the left edge, and creating a blob with a width of the median. That blob is then removed from the original image, and this process repeats until there are no more lines to expand. The result of this method is shown Figures 4-6.

A set of experiments has been conducted to compare edge growing crevice detection to finger blob segmentation crevice detection. Different complexity piano pieces were played, in different lighting conditions and using different hands. The results showed that edge growing works well at finding where one finger ends and another begins, but it is not good at identifying the entire finger; it is used primarily to separate each finger, and locate the boundaries. At the same time, finger blob growing finds well the estimate of the best possible fingers, but the results of detecting the top and bottom boundaries of fingers are worse. Therefore, for the current demonstration the second of these methods is used.

7. Video annotation of piano playing

In order to apply the developed video recognition techniques to piano pedagogy, they have been integrated with the MIDI events reading software. The video capture is synchronized with the MIDI capture and the video-MIDI calibration described in Section 4 is performed. Thus created MIDI annotation program is named *C-MIDI* to signify the idea that the program can see("C") the MIDI events which are otherwise "blind". To test its performance, a professional pianist is asked to play several piano pieces of different levels of speed and complexity, while the results of playing hands and fingers detection are visualized in front of him on a computer monitor (as shown in Figure 2).

Live video recordings of these experiments are provided at our website (see e.g. http://synapse.vit.iit.nrc.ca/piano/demo/SeePianistPlay.wmv.), while the snapshots from these recordings are shown in Figures 4-6. Figure 6 shows the final output of the *C-MIDI* program. The subwindows on a screen show the following (in a clockwise order from the top): the image captured by camera; the computed background image of the keyboard which is used to detect hands as the foreground; the binarized image used to detect the black keys of the piano keyboard which is also used for video-MIDI calibration; the automatically detected piano keys (highlighted as white rectangles on the bottom right image), the segmented blobs in the foreground images (coloured by the number of blobs detected in the bottom left image); and the final finger and hand detection results shown upside down, as camera sees it (on the top left), and in a vertically flipped for a convenient viewing by a pianist (bottom middle), where the label of the finger that played a key is shown on the top of the image. The result of the vision-based MIDI annotation is also in a separate window at the bottom right: each received MIDI event receives a visual label for the hand (either 1 or 2, i.e. left or right) and finger (either 1,2,3,4, or 5, counted from right to left) that played it. When the finger can not be determined, the annotation is omitted.

8. Conclusion

In this paper we have shown how to design a video recognition system for detection and tracking of pianist hands and fingers. Solutions to three video recognition tasks have been proposed: piano keyboard recognition, hand recognition and finger recognition.

The experiments conducted with professional pianists demonstrate well the potential of the proposed techniques and, in particular, their applicability for piano playing annotation. More specifically, unless a music piece is very complex and involves many overlapping hands, hands are tracked very well, which makes robust annotation of played MIDI-events using hand labels possible. This result alone can significantly advance distant and off-line piano learning, if, for example, a filter program is written on top of a MIDI sending server that sends only those MIDI events that are played by a requested hand.

As for finger detection, the experiments show that our method to detect pianist fingers using the proposed in the paper crevice detection operator is very powerful. The professional pianists, as they play and see at the same time on a computer screen the visual annotation of their playing, are content acknowledging the correct finger annotation in about a half of cases. Of the other half, the fingers are either left unmarked or can provide a set of possibilities to choose from. While these results may not appear outstanding, they do show enough promise of our approach as well as encourage further development of the video-recognition tools for music teaching and performance visualization. One of such tools, called *C-MIDI* and described in this paper, can now be used by piano players as a visual aid in piano teaching and performing.

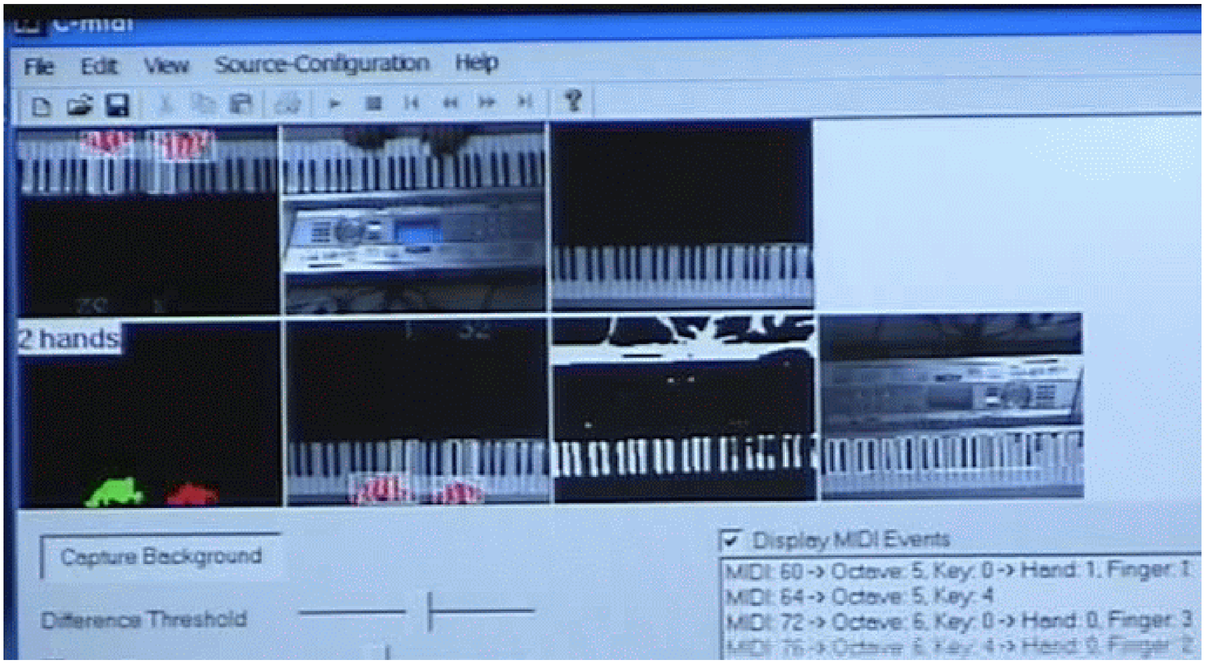
The promising directions for future work are seen in both of the addressed in the paper domains. In the domain of music pedagogy, it appears very promising to perform vision-based annotations of MIDI playing for other musical instruments. In particular, annotation of guitar and violin playing will be very useful, because the recorded MIDI data of these instruments are practically useless for pedagogical needs, since they do not provide information on *how* and *where* on a fret- or finger- board these data are played. In the domain of video recognition, the work on improving finger detection has to be continued. In particular, imposing additional constraints on the finger inter-relationship and their temporal coherency, which is not done at the present work, is believed to improve the recognition results.

Acknowledgements

This work is partially funded by the Piano Pedagogy Laboratory of the Department of Music of the University of Ottawa. The authors thank the chair of the laboratory, Gilles Comeau, for his initiative to bring advances in computational video to a new application area – piano pedagogy, and also for participating in the experiments. We also thanks Mihir Sharma for the help in writing code for the MIDI events reader, and Emond Bruno and Martin Brooks for stimulation discussions.

References

[1] International MIDI association, 1988. Standard MIDI Files 1.0. Los Angeles, CA. 1



```
MIDI: 60 -> Octave: 5, Key: 0 -> Hand: 1, Finger: 1
MIDI: 64 -> Octave: 5, Key: 4
MIDI: 72 -> Octave: 6, Key: 0 -> Hand: 0, Finger: 3
MIDI: 76 -> Octave: 6, Key: 4 -> Hand: 0, Finger: 2
```

Figure 6. The output of *C-MIDI* piano playing video annotation program (a snapshot from a video recording of a live annotation).

[2] A. Argyros and M. Lourakis. Real time tracking of multiple skin-colored objects with a possibly moving camera, 2004. Proceedings of the European Conference on Computer Vision (ECCV04), vol. 3, pp. 368-379. 4, 5

[3] J. Davis and M. Shah. Toward 3-d gesture recognition, 1999. International Journal of Pattern Recognition and Artificial Intelligence, Volume 13 Number 3, pp. 381-393. 5

[4] B. Emond and M. Brooks. The private video sharing and annotation server: A broadband application for teacher training and music education, 2003. International Lisp Conference, New York, NY. 1

[5] D. Gorodnichy. Seeing faces in video by computers. Editorial for special issue on face processing in video sequences. *Image and Video Computing*, 24(5):1–6, 2006. 3, 4

[6] D. O. Gorodnichy. Facial recognition in video. In *Proc. Int. Conf. on Audio- and Video-Based Biometric Person Authentication (AVBPA'03), LNCS 2688, pp. 505-514, Guildford, UK*, 2003. 3, 4

[7] D. O. Gorodnichy and L. Yin. Introduction to the first internation workshop on video processing for security (vp4s-06). In *Proceedings of the Canadian conference Computer And Robot Vision (CRV'06), June 7-9, Quebec City, Canada*, 2006. 4

[8] B. K. P. Horn. Understanding image intensities. *Artificial Intelligence, Vol. 8, pp 201-231*, 1977. 6

[9] H.Sekiguchi and S. Eiho. Generating the human piano performance in virtual space, 2005. International Conference on Pattern Recognition (ICPR'00), pp. 4477-4481. 1

[10] I. Infantino, A. Chella, H. Dindo, and I. Macaluso. A cognitive architecture for robotic hand posture learning, 2005. pp. 42- 52, Volume: 35, Issue: 1. 1, 2, 5

[11] J. Letessier and F. J. Brard. Visual tracking of bare fingers for interactive surfaces, 2004. ACM Symposium on User Interface Software and Technology (UIST), Santa Fe, New Mexico, USA. 1, 2

[12] J. Mackie and B.McCane. Finger detection with decision trees, 2004. Proceedings of Image and Vision Computing New Zealand 2004, pp 399-403. 1, 2, 5

[13] C. Nolker and H. Ritter. Visual recognition of continuous hand postures, 2002. IEEE Transactions on Neural Networks, Volume: 13, Issue: 4, pp. 983- 994. 1, 2, 4, 5

[14] K. Oka, Y. Sato, and H. Koike. Real-time fingertip tracking and gesture recognition, 2002. Computer Graphics and Applications, IEEE, Volume 22, Issue 6, Nov.-Dec. 2002 Page(s):64 - 71. 1, 2, 5

[15] J.-C. Terrillon, A. Piplr, Y. Niwa, and K. Yamamoto. Robust face detection and japanese sign language hand posture recognition for human-computer interaction in an "intelligent" room, 2002. Proceedings of Intern. Conf. on Vision Interface (VI 2002), online at www.cipprs.org/vi2002. 1, 2, 4, 5