The Visual Computer manuscript No.
(will be inserted by the editor)

A tool for teaching musical metrics based on computer vision

Rodrigo Schramm, Cláudio R. Jung

Unisinos - Universidade do Vale do Rio dos Sinos
Graduate School of Applied Computing
Av. Unisinos, 950. São Leopoldo, RS, Brazil, 93022-000
e-mail: rodrigo.schramm@gmail.com,crjung@unisinos.br

Abstract Computerized systems for e-learning and entertainment have been created in different areas of knowledge. This work presents a system designed to track and evaluate hand movements for conducting musical measures - binary, ternary and quaternary metrics - identifying for the user the maintenance of *tempo* and the correctness of hand patterns. The main goal of this work is to aid the study of musical rhythm for beginners, not focusing on conducting styles. The system was developed with computer vision algorithms to detect movements of the hand, and a finite-state machine was used to recognize the patterns. Feedback for the user is given through visual information on the screen. The accuracy of the results is verified by an external observer (a conducting specialist), with satisfactory results.

Key words computer vision, hand detection, object tracking, rythmic, music teaching

1 Introduction

This article presents a system capable of tracking and evaluating hand movements in conducting musical measures, aiming to support initial movements of rhythmic study. This tool helps the user in the maintenance of timing and accuracy of hand patterns in binary, ternary and quaternary metrics. The attention is focused only on metrics structure of the conducting measures, not considering conducting styles. The core of the proposed system consists of the detection and tracking of hands (based on background subtraction algorithms and segmentation of foreground object based on chromatic information), and the recognition of movement patterns, developed using a Transition System. The feedback for the user consists of a graphical interface shown on the computer screen, that indicates if the movement and/or tempo are correct or not. This article is structured in 5 sections. In the next section, a few connected works will be briefly described. In section 3, we present the proposed model. Section 4 illustrates some experimental results, and the conclusions are drawn in the last section.

2 Related Work

During the bibliographical review, no reference with the explicit intention of aiding the musical metric study using computer vision was found. However, there are some projects with similar purposes, that are briefly described next. In the work entitled *Personal Orchestra* [1], Borchers *et al.* developed an interactive system, that uses audio and video recordings of the Viena Philarmonica Orchestra. In this system, the user can control the volume and the timing of the music through simple gestures, such as the movement of the hand in the shape of a pendulum, and even with a rudimentary movement consisted of "to Up/to Down" gestures made with the hand.

Lee *et al.* [7] developed the *conga - A Framework for Adaptive Conducting Gesture Analysis*. This framework makes the conducting gesture analysis possible from a set of hand coordinates. The *conga* system was projected to be used through configurable blocks, where each block is responsible for a specific task. Several blocks may be interconnected through one acyclical graphic, making the detection of more complex movements possible.

Researchers of the Bell Labs [10] developed the *Visual Conducting Interface* (VCI) that uses a pair of special cameras, capable of capturing 60 frames per second. Through these cameras, the conductor's hand and baton are captured and followed in the 3D space. Using this system, the user can control the timing, the volume of each beat time and the global volume of the music. The cameras are located above the conductor, and are synchronized and calibrated by using a system of coordinates common to all. The movement of the baton on the vertical axis is evaluated to identify the *beat time* and the position of the hand is responsible for defining the

volume of different sections of the orchestra. The horizontal hand position defines the section of the orchestra (set of instruments) that will have the volume modified.

Ilmonen and Takala [3] developed a system using magnetic sensors to capture the conductor's movements, and neural networks to process them. The three-dimensional hands position, obtained by the sensors, are used as entry to the neural networks. The architecture of this system is modular, divided in three categories that work in collaboration. The first and the second are related to the detection of the movement patterns, and the third applies the alterations of dynamics and time detected in music.

An approach using *Hidden Markov Models* (HMMs) was developed by Kolesnik and Wanderley [6]. In their approach, the images are captured by two cameras, one providing a frontal view and the other a lateral view. Through these images, the hands are segmented and their coordinates are processed by HMMs, that recognize the movement patterns done by the user.

There are also other computer-based approaches for teaching/learning arts, such as the animation-based dance tutor [12] and the computer-based trainer for music conducting [9]. However, these methods do not employ computer vision.

According to the approaches described in the previous paragraphs, there are several proposals intending to recognize the conductor's gestures and, after interpreting them, apply alterations in the music. Some methods focus on capturing the movement, using different techniques for tracking and processing the images. Others concern about interpreting the movement obtained by the previous techniques. In the next section, the model proposed in this work is presented, dealing with the detection and tracking of the hands, as well as the interpretation and identification of the movements done by the user.

3 The Proposed Model

In the previous section, several human-computer interface systems were presented. These systems allow the user to control and modify several characteristics of musical electronic systems, such as *tempo*, volume and timbre. It is important to note that such parameters refer to the music itself, but they do not help the teaching of music. In fact, we could not find any work in the literature that was focused on aiding the teaching process for music students, regarding musical metrics.

The goal of this work is to aid the beginner music student in learning musical metrics through the movements of the hand. For that purpose, we developed a tool that captures hand movements using computer vision algorithm, being able to recognize binary, ternary and quaternary measures patterns, which are related to pre-defined hand movements. The key reference points for the hand in these movements, which are based on the book *The Grammar of Conducting* by Max Rudolf [8], are illustrated in Fig. 1.

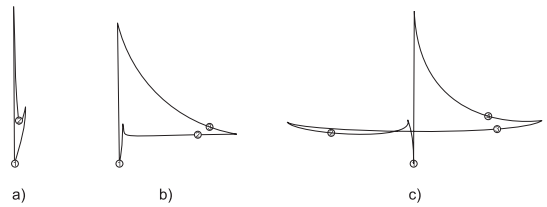

Fig. 1 Simple rythmic patterns adopted as models in this work. a) Simple binary. b) Simple ternary. c) Simple quaternary. Numbers indicate the positions of equal time intervals, that should be followed by the music student.

This section presents the proposed model for aiding the learning of musical metrics. The model is organized into two steps. In the first, computer vision algorithms are employed to track the center of the hand, generating a set of points. In the second step, this list of points is analyzed, and the correctness of the movement as well as the timing of the movement are evaluated. Fig. 2 illustrates the fluxogram containing the basic steps implemented in the proposed system.

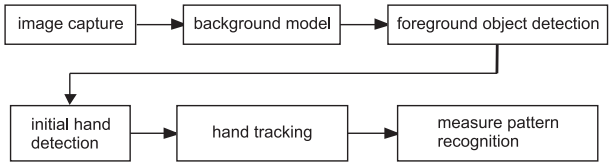

Fig. 2 Fluxogram of the model.

3.1 Hand Tracking

The first step of our system comprises hand tracking. For that purpose, two algorithms are applied sequentially to the input video sequence: background subtraction and hand segmentation based on chromatic information. It should be noticed that there are several techniques for hand segmentation based on skin color, using different approaches. A class of these techniques are based on statistics and/or neural networks [14,11,13]. Although such methods indeed produce nice segmentation results, they require a training period to learn the color distribution of hands. Another class of hand segmentation algorithms are based on image thresholding. Such methods do not require a previous training step and are generally faster, but also more sensitive to ambient illumination and to the camera sensor (in particular, webcams tend to have poor color sensitivity). Also, there is a compromise between false positives and false negatives

in thresholding methods: tighter limits present few false positives, but maybe large negatives; on the other hand, more relaxed thresholds reduce the number of false negatives, but increase false positives.

In this work, we combine a threshold-based hand segmentation based on chromatic information with a background subtraction algorithm. The background subtraction algorithm provides a mathematical model of the static background, and compares it with every new frame of the video sequence. Significant discrepancies are related to foreground objects, which are the search space for hand segmentation. To reduce the computational cost of our algorithm, we use only luminance information for background subtraction, and use color information only to analyze the remaining foreground blobs.

3.1.1 Background Subtraction

In this work, we adopted the background subtraction technique described in [4], which is based on W4 [2]. Initially, a training period is used to capture a set of frames, which are stored in a stack of k grayscale images. Let $V^k(i,j)$ represent the intensity of pixel (i,j) in the k^{th} image. To minimize the effect of moving objects during the training period, initially a pre-screening of stationary pixels is made. A pixel (i,j) at frame k is considered stationary if:

$$\left|V^k(i,j) - \lambda(i,j)\right| \leq 2\sigma(i,j), \tag{1}$$

where $\lambda(i,j)$ is the temporal median at pixel (i,j) and $\sigma(i,j)$ is the standard deviation. Within this set of stationary pixels (denoted by $V_s^z(i,j)$), the minimum value m, maximum value n and the maximum interframe difference d are computed for each pixel. The background model $\boldsymbol{B}$ is the given by:

$$\boldsymbol{B} = \begin{bmatrix} m(i,j) \\ n(i,j) \\ d(i,j) \end{bmatrix} = \begin{bmatrix} \min(V_s^z(i,j)) \\ \max(V_s^z(i,j)) \\ \max(V_s^z(i,j) - V_s^{z-1}(i,j)) \end{bmatrix} \tag{2}$$

Usually, better results are obtained when larger training periods are used. However, we used in this work only 60 frames in the training period, so that the user does not have to wait too long. Due to this small training period, it is advisable to train the model without moving objects in the background.

In the test stage, each new frame of the video sequence is compared with $\boldsymbol{B}$. A certain pixel (i,j) at frame t is considered background if:

$$m(i,j) - K\alpha \leq I^t(i,j) \leq n(i,j) + K\alpha, \tag{3}$$

where K is a discrimination threshold (default value is $K = 4$), and α =median$\{d(i,j)\}$) is the median of the interframe difference across all image pixels. The result of this process is a binary image F^t at each frame t, that returns 1 for foreground pixels and 0 for background pixels.

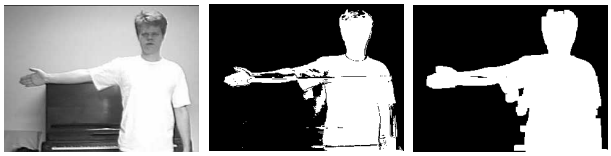

Fig. 3 a) Luminance of a captured image. b) Result of background subtraction. c) Post-processing.

For sakes of simplicity (and speed), we have not implemented any adaptation to the background model, meaning that it becomes susceptible to illumination changes. However, since the proposed application is aimed for indoor applications and relatively controlled environments, the lack of adaptation is not significant. Fig. 3 illustrates the process of background subtraction. As it can be observed in Fig. 3(b), some holes or spurious responses arise due to noise, particularly in regions where the foreground intensity is similar to the background intensity. To minimize the effect of noise, a morphological post-processing algorithm is applied. In this work, we employed morphological closing with a cross-shaped structuring element [5]. The result of background subtraction after post-processing is depicted in Fig. 3(c).

3.1.2 Hand Detection and Tracking

The output of the previous step is a binary image F_t for each frame t. Such binary image contains foreground objects (in general, the silhouette of the user), and the hand is not discriminated from the rest of the body. To isolate only the hand, we used a segmentation technique based on adaptive thresholding in the HSV color space, assuming that pixel colors associated with the hand must present colors sufficiently different than pixels related to the rest of the body. To ensure such hypothesis, the user can be wearing shirts with long sleeves (so that the hand is isolated from the rest of the arm), and that the hand must be sufficiently apart from the face (because the hand presents the same colors as the face). Another (more robust) solution is to employ a colored glove, to discriminate the hand from the arm and the face.

The first step of the segmentation algorithm is to learn the color pattern of the hand (either skin color or glove). The user must initially raise his/her right arm until it is parallel to the ground, as illustrated in Fig. 3. The background subtraction algorithm extracts the body of the user, and the summation along image columns produce a sharp peak in the row where the arm is placed. In fact, if $F_t(i,j)$ denotes the result of background subtraction at pixel (i,j), then the horizontal summation is given by:

$$A(i) = \sum_j F_t(i,j), \tag{4}$$

and the row where $A(i)$ achieves its maximum value is $i_{\max}$ given by:

$$i_{\max} = \underset{i}{\operatorname{argmax}}\, A(i). \tag{5}$$

A raster scan of $F_t(i,j)$ along the columns at row $i_{\max}$ is performed, and the tip of the hand is obtained at the end of the scan. The length of the arm is obtained by subtracting the value $A(i_{\max})$ (which corresponds to body plus arm) from the estimate of the body width, computed as the median value of $A(i)$. Finally, the length of the hand is assumed to be approximately 1/4 of the length of the arm, and a template is placed at the center of the hand. An example of the horizontal projection is shown in Fig. 4(a), the detection of the tip of the hand is shown in Fig. 4(b), and the placement of the hand template is shown in Fig. 4(c).

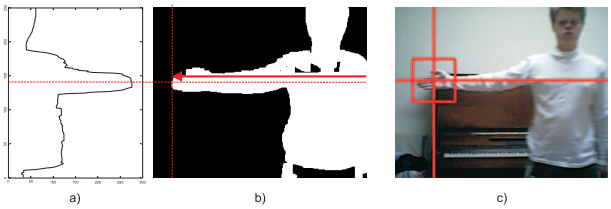

Fig. 4 Detection of the initial position of the hand. a) Horizontal projection of foreground blob. b) Raster scan to find the tip of the hand. c) Placement of the hand template.

The next step of the algorithm is to segment hand pixels using thresholds in the HSV color space. More precisely, pixels related to the hand are detected through:

$$H_{\min} < H < H_{\max} \text{ and } S_{\min} < S < S_{\max} \text{ and } V_{\min} < V < V_{\max}, \quad (6)$$

where $H_{\min}$, $H_{\max}$, $S_{\min}$, $S_{\max}$, $V_{\min}$, $V_{\max}$ are thresholds. Such thresholds are obtained adaptively, based on simple statistics computed in a small time period (approximately 3 seconds).[1] In this period, the mean μ_c and standard deviation σ_c is computed for each coordinate in the HSV color space, where $c \in \{H, S, V\}$ denotes the type of coordinate (Hue, Saturation or Value). The adaptive thresholds are given by:

$$\begin{aligned} H_{\min} &= \mu_H - 2\beta\sigma_H \\ H_{\max} &= \mu_H + 2\beta\sigma_H \\ S_{\min} &= \mu_S - \beta\sigma_S \\ S_{\max} &= \mu_S + \beta\sigma_S \\ V_{\min} &= \mu_V - \beta\sigma_V \\ V_{\max} &= \mu_V + \beta\sigma_V \end{aligned}, \quad (7)$$

and β is a threshold that controls the deviation from the mean, which allows the user to have some control over the segmentation results. It should be noticed that our experimental results showed that the Hue tends to be noisier than Saturation and Value, leading us to multiply β by 2 when defining the thresholds for H.

[1] It should be noticed that a very small training period is used in the proposed approach, opposed to traditional statistical methods.

After defining the adaptive thresholds for hand colors, equation (7) is applied to the last frame of the training period. The connected components of the resulting binary image are obtained, and the center of the bounding box that contains the connected component with the largest area is retrieved as the initial hand position.

The next step of the algorithm is the tracking of the hand. For video sequences acquired at 30 FPS, the position of the hand is not expected to suffer a large variation in subsequent frames. Let $(i(t), j(t))$ denote the hand position at frame t, and let us consider a square search region centered at $(i(t), j(t))$, 3 times larger than the size of the hand template. Within this search region at frame $t+1$, the background subtraction algorithm is computed, the RGB→HSV conversion is performed, and equation (7) is applied to foreground pixels. Again, the connected components are computed, and the center of the bounding box that contains the component with the largest area is defined as the new hand position. This procedure is applied at each new frame, resulting in a set of coordinates $P(t) = (i(t), j(t))$ that describe the trajectory of the center of the hand.

3.2 Detection of Metric Patterns

The binary, ternary and quaternary movements present individual trajectories in the plane orthogonal to the camera, and observed in Fig. 1. The first *beat* of each of these movements is always broader than the others (in the vertical direction), and it is oriented from top to bottom. After achieving the lowest point, the hand goes up a little bit, and then follows to the position that delimits the next *beat*, and so on. In fact, the pattern that defines the end of each *beat* is similar to a ball kicking: the hand goes down and then up, producing a local minima in the vertical direction, as illustrated in Fig. 1. The oscillating pattern of the hand, upwards and downwards, produces several local maxima and minima with equal distance in the time axis, representing the desired *tempo*. Furthermore, the first *beat* presents a wider amplitude when compared to the others, and can be used to detect the beginning of each movement.

In summary, the local extrema of the trajectory in vertical and horizontal directions can be used to detect the beginning of each *beat*, and they are strongly explored in this work. However, the hand tracking procedure is inherently noisy (the estimate of the center of the hand generally produces some small amplitude oscillations with high frequency), which causes several spurious local extrema in the estimated trajectory.

To produce a smooth trajectory in execution time, a causal low-pass filter can be applied for the temporal series $i(t)$ and $j(t)$. In this work, each coordinate of the trajectory is filtered using a second-order Chebyshev type II causal digital filter with cut-off frequency $w_c = 0.4\pi$ (which corresponds to a 6Hz if the video sequence is ac-

quired at 30 FPS), that showed to be efficient to remove spurious local extrema.

3.2.1 Events

Each local minimum (*beat time*) of the vertical movement generates an *event*. We call this event *toUp*, since it is related to an inversion of the vertical movement from descending to ascending. Analogously, local maxima are called *toDown* events.

Fig. 5 illustrates typical trajectories (vertical and horizontal movements) for binary, ternary and quaternary metrics. It can be observed that, in the binary metric, there is no significant variation on the horizontal coordinate; in the ternary movement, from the first to the second *beat*, the amplitude in the horizontal direction increases, while it decreases from the second to the third; in the quaternary metric, the horizontal amplitude decreases from the first to the second *beat* (opposed to the ternary metric), increases from the second to third, and decreases again from the third to fourth *beat*.

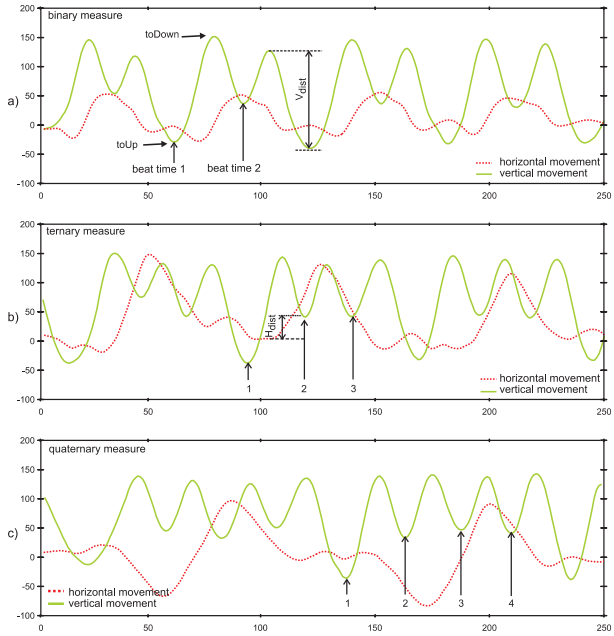

Fig. 5 Typical horizontal and vertical movements for (a) binary, (b) ternary and (c) quaternary metrics.

The combination of horizontal and vertical variations is explored in this work to identify the patterns that characterize each metric. In fact, the key events used to characterize the *beats* are *toDown* events (local maxima). When such event is detected, the horizontal and vertical coordinates of the local extrema are stored. Let $toDown_x^t$ e $toDown_y^t$ denote the horizontal and vertical coordinates of the last local maxima that was detected, and let $toDown_x^{t-1}$ e $toDown_y^{t-1}$ denote the same coordinates for the previous local maxima (just the last two local maxima are analyzed). Also, let $toUp_x^t$, $toUp_y^t$, $toUp_x^{t-1}$ and $toUp_y^{t-1}$ denote analogous variables related to local minima. These parameters are used to classify the movement performed by the user as binary, ternary or quaternary, and also to detect if the *tempo* are being timed correctly. For that purpose, when a *toUp* event is detected, the algorithm extracts the following measures:

1. $H_{dist}^t = |toUp_x^t - toDown_x^t|$: represents the horizontal displacement between the last local maxima and the last local minima;
2. H_s^t: represents the orientation of the horizontal movement between the last local maxima and the last local minima, being left-right if $toUp_x^t - toDown_x^t > 0$, or right-left otherwise.
3. $H_{dist}^{t-1} = |toUp_x^{t-1} - toDown_x^{t-1}|$: represents the horizontal displacement between the previous (one before the last) local maxima and the previous local minima;
4. H_s^{t-1}: represents the orientation of the horizontal movement between the previous (one before the last) local maxima and the previous local minima, being left-right if $toUp_x^{t-1} - toDown_x^{t-1} > 0$, or right-left otherwise.
5. $V_{dist}^t = |toUp_y^t - toDown_y^t|$: represents the vertical displacement between the last local maxima and the last local minima.

Table 1 Generation of input events for the transition system using geometric rules

Input event	Rules
$EBTQ_1$	$H_{dist}^t < 30$ $V_{dist}^t > 30$
EB_2	$H_{dist}^t < 30$ $V_{dist}^t > 30$ $V_{dist}^t \leq V_{dist}^{t-1}$
ET_2	$H_{dist}^t > 30$ $H_s^t =$ left-right
ET_3	$H_{dist}^t > 30$ $H_s^t =$ right-left
EQ_2	$H_{dist}^t > 30$ $H_s^t =$ right-left
EQ_3	$H_{dist}^t > 40$ $H_s^t =$ left-right
EQ_4	$H_{dist}^t > 20$ $H_s^t =$ right-left

These metrics are then evaluated through a function that applies several geometrical restrictions and returns a list of *input events*, that represents the possible gestures performed by the user. Table 1 defines the relationships between each geometrical rule and the corre-

sponding input event. This Table was built based on the expected trajectories for each metric (binary, ternary and quaternary), for a person standing at a distance of approximately two meters from the camera, and video sequence acquired with a resolution of 240×320 pixels.

3.2.2 Finite-State Machine

To improve the precision of the rules and to eliminate ambiguous movements in space (but distinguishable in time), a transition system that evaluates the temporal information of the movement was employed. The states in this system represent the *beats* of the binary, ternary and quaternary metrics, according to Table 2. The events that are returned through the evaluation of the proposed metrics (described above) are parameters fed to the finite-state machine. If an input event is accepted, a new state is assigned as the current state, representing the current *beat* of the movement. When an undefined input event is generated, the transition system defines *ini* as the current state.

Table 2 States of the transition system representing the *tempos* of each metric

State	Metric	*Tempo*
ini	undefined	undefined
btq_1	binary/ternary/quaternary	1
b_2	binary	2
t_2	ternary	2
t_3	ternary	3
q_2	quaternary	2
q_3	quaternary	3
q_4	quaternary	4

The transition system is formally represented as $A = (Q, \Sigma, \delta, ini)$, where:

1. Q is the set of possible states: $\{ini, btq_1, b_2, t_2, t_3, q_2, q_3, q_4\}$.
2. $\Sigma = \{EBTQ_1, EB_2, ET_2, ET_3, EQ_2, EQ_3, EQ_4\}$ is the finite set of possible events.
3. $ini \in Q$ is the initial state.
4. δ is the transition function ($\delta : Q \times \Sigma \rightarrow Q$) defined by the graph illustrated in Fig. 6.

At each *toUp* event, a subset of Σ, obtained from the rules shown in Table 1 and the current state, are fed to function δ. This function evaluates the input parameters according to the graph illustrated in Fig. 6, and returns as output the next state of the transition system, updating the current state of the metric performed by the user.

When *toUp* events are detected, not only the trajectory is analyzed, but also the duration of the movement is also computed. The system then evaluates if the *tempo* performed by the user is coherent with the metric. In fact, a Metronome is included in the system, producing beeps at the desired time steps. The movement (with respect to the timing) is considered correct when the temporal difference between the beep produced by the system and the detected event *toUp* is smaller than a threshold T_{time}. Such threshold can be adjusted by the user according to the desired tolerance.

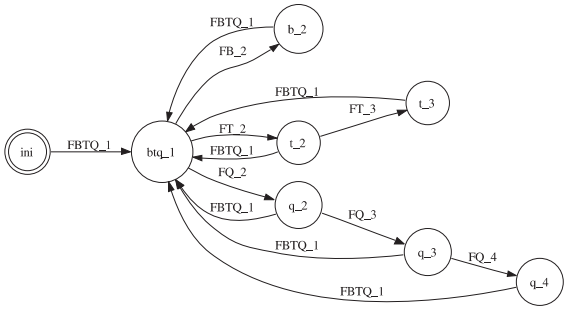

Fig. 6 Graphical representation of the Transition System.

4 Experimental Results

A collection of tests was done to validate the proposed system. In these tests, the movements of the user were captured by a webcam (*Genius - Slim 321C model*) from an approximate distance of two meters and under ambient illumination. To avoid problems during the image segmentation, the users wore a yellow glove, making the color of the region represented by the hand more homogeneous and, at the same time, more discrepant regarding to the rest of the body. Even in such conditions, it was sometimes necessary to do small manual adjustments in the color thresholds (equation (7) to obtain an appropriate tracking.

According to the basic conducting movement verified on Max Rudolf's diagrams, each kind of measure corresponds to a movement pattern, resulting in a drawing that the hand uninterruptedly persecutes in space. The precise time step at each new *beat*, along of this basic structure, is marked by the inversion of descending to ascending direction in the vertical movement of the hand. Moreover, such time steps must keep equitemporal between two consecutive inversions.

The performance of the system was verified by comparing its results (in terms of correctness of hand patterns and keeping of the *tempo*) with the evaluation of an external observer - an expert at the conducting area. In this case, it was significant to verify whether the system is capable to detect correspondence or not to the demanding level that the conventional evaluation, purely of the musical point of view, considers relevant. A requirement so that the system recognizes the movements and the timing, and that is also important to the musician, is that each *beat* would be effectively marked by

the inversion of the direction of the vertical movement, which needs to be clearly established for the user.

To test the system, three people with different levels of knowledge of musical metrics were used, namely a university graduated conductor P_1, a beginner student of music graduate course P_2, and a completely inexperienced person P_3. Each person performed a set of 8 measures to each one of the metrics (binary, tertiary and quaternary) in a timing of 60 *beats* per minute ($MM = 60$). The movements were recorded in video sequences, and further evaluated by the expert. Fig. 7 illustrates some frames of the movement for a quaternary metric, that was correctly classified by the system.

Fig. 7 Some frames depicting a quaternaty metric, correctly detected by the system.

A quantitative evaluation of the system was performed by measuring the number true positives (TP), true negatives (TN), false positives (FP), false negatives (FN) with respect only to the *movement* parameter (ground truth was obtained by the expert's opinion). In this work, TP are the results accepted by the system as well as the external evaluator and, in a similar way, TN are the results rejected by both. Oppositely, FP and FN are the events obtained by the system that are not in conformity with the evaluator's opinion. As it can be observed in Table 3, the system correctly identifies the majority of movements of the metric structures (100% of the binary movements, 86% of the ternary movements and 90% of the quaternary movements), not losing in the nuance of the course between one beat and the next, which would be of interest to the stylistic question, but can be considered irrelevant in an initial moment of the musical metric study.

Table 3 Quantitative evaluation of obtained results with respect to the correctness of the movement only (disregarding timing issues).

Metric	TP	FP	TN	FN
B	100.00%	0.00%	0.00%	0.00%
T	86.11%	5.56%	5.56%	2.78%
Q	90.63%	4.17%	2.08%	3.13%

Table 4 presents similar results, but regarding only the evaluation of the *timing* parameter (temporal equidistance between successive inversions), using $T_{\text{time}} = 300ms$ (a typical example of the difference between the time marked by the Metronome and the time marked by the user's hand is shown in Fig. 8). It can be noticed that a larger disagreement was detected in comparison to the *movement* parameter (Table 3). Part of this discrepancy can be explained by small failures during the tracking of the hand (due to residual noise after the low-pass filtering), which causes the inversion of the vertical movement direction at undesirable moments. Also, the increase of the number of false negatives (FN) happened because that the precision of the system may surpass the relevant precision according to the expert's opinion. In fact, a preliminar investigation indicated that movements were classified as *mistimed* by the expert when the temporal distance exceeded 400 milliseconds, while the system imposed a tighter threshold ($T_{\text{time}} = 300$ milliseconds). In order to confirm such hypothesis, however, additional studies are necessary.

Table 4 Quantitative evaluation of obtained results with respect to the correctness of the timing only.

Metric	TP	FP	TN	FN
B	81.25%	10.42%	2.08%	6.25%
T	73.61%	9.72%	8.33%	8.33%
Q	82.29%	4.17%	5.21%	8.33%

According to Table 3 and 4 the ternary movement obtained the highest rate of mistakes. This occured be-

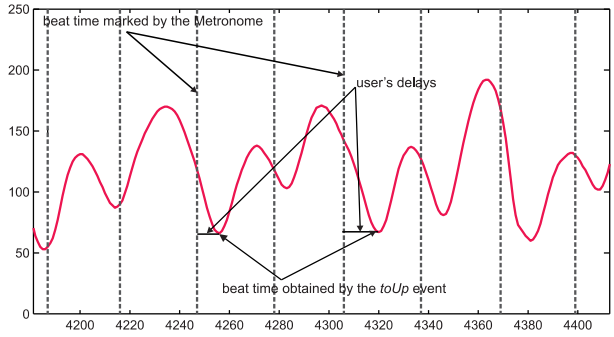

Fig. 8 Exemplification of the delay noted by the system during the execution of the binary metric.

cause the users made *toUp* movement of the third beat on the same horizontal level of the second beat and, so, did not satisfy the ET_2 condition. In this case, in order to enable the system verifying freer movements, the thresholds of Table 1 must be relaxed.

5 Conclusions

This work presented a method for evaluating the correctness of hand conducting movements for beginners in musical studies. The proposed system is based on computer vision algorithms to track the temporal evolution of the hand position captured with a frontal-view camera (possibly a webcam), and the evaluation of the trajectories to detect the accuracy of binary, ternary or quaternary metrics (with respect to movement and timing).

The experimental results indicated that the developed system is capable of identifying to the user the maintenance of the tracking and the correction of the drawings that result from the movements of his/her hands in the conducting measures, not considering the conducting stylistic questions. The obtained results indicated that the system has potential as an auxiliary tool in the musical rhythm learning. In fact, the quantitative evaluation performed in this work showed a low number of false positives and false negatives, using a regular webcam (with USB 2.0 output). The performance of the system is expected to increase if better quality cameras are used (in general, most webcams present a low definition of colors, that may compromise the tracking of the hand).

For future work, we intend to improve the hand detection scheme by exploring other segmentation techniques and other color spaces. In addition, we intend to do exhaustive tests with the system to identify and to solve cases in which the detection of the pattern might fail, as well as to use several different cameras for a better comparison.

References

1. Borchers, J.O., Samminger, W., Mühlhäuser, M.: Personal orchestra: Conducting audio/video music recordings. In: WEDELMUSIC 2002 International Conference On Web Delivering of Music (2002)
2. Haritaoglu, I., Harwood, D., Davis, L.S.: W4: Real-time surveillance of people and their activities. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8), 809–830 (2000)
3. Ilmonen, T., Takala, T.: Conductor following with artificial neural networks. In: Proc. Int. Computer Music Conf. (ICMC'99), pp. 367–370. Beijing, China (1999)
4. Jacques Jr., J.C.S., Jung, C.R., Musse, S.R.: Background subtraction and shadow detection in grayscale video sequences. In: SIBGRAPI '05: Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing, p. 189. IEEE Computer Society, Washington, DC, USA (2005)
5. Jain, A.K.: Fundamentals of digital image processing. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1989)
6. Kolesnik, P.: "Conducting gesture recognition, analysis and performance system". Master's thesis, McGill University, Montreal, Canada (2004)
7. Lee, E., Grüll, I., Kiel, H., Borchers, J.: conga: A framework for adaptive conducting gesture analysis. In: NIME 2006 International Conference on New Interfaces for Musical Expression, pp. 260–265. Paris, France (2006)
8. Rudolf, M.: The Grammar of Conducting, 2 edn. Schirmer Books, Inc., New York, USA (1980)
9. Schwaegler, D.G.: Acomputer-based trainer for music conducting: the effects of four feedbackmodes. Ph.D. thesis, University of Iowa (1984)
10. Segen, J., Kumar, S., Gluckman, J.: Visual interface for conducting virtual orchestra. In: Proceedings of the International Conference on Pattern Recognition, p. 1276. IEEE Computer Society, Washington, DC, USA (2000)
11. Seow, M.J., Valaparla, D., Asari, V.K.: Neural network based skin color model for face detection. In: Proceedings of the 32nd Workshop on Applied Imagery Pattern Recognition, p. 141. IEEE Computer Society, Los Alamitos, CA, USA (2003)
12. Sukel, K.E., Catrambone, R., Essa, I., Brostow, G.J.: Presenting movement in a computer dance tutor. International Journal of Human-Computer Interaction **15**(3), 433–452 (2003)
13. Yao, Y., Zhu, M.L.: Hand tracking in time-varying illumination. In: Third International Conference on Machine Learning and Cybernetics, vol. 7, pp. 4071–4075. IEEE Computer Society Press (2004)
14. Zarit, B.D., Super, B.J., Quek, F.K.H.: Comparison of five color models in skin pixel classification. In: RATFG-RTS '99: Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, p. 58. IEEE Computer Society, Washington, DC, USA (1999)