Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x Vol. x, No. x, month year, xx–xx

Article Type (Research/Review)

Dance to the Beat: Enhancing Dancing Performance in Video

Rachele Bellini[1], Yanir Kleiman[1], and Daniel Cohen-Or[1]

Abstract In this paper we introduce a video post-processing method that enhances the rhythm of a dancing performance, in the sense that the dancing movements are more cohesive with the beat of the music. The dancing performance as observed in a video is analyzed and segmented into motion intervals delimited by motion-beats. We present an image-space method to extract the motion-beats of a video by detecting frames at which there is a significant change in direction or motion stops. The motion-beats are then synchronized with the music-beats such that as many beats as possible are matched with as little as possible time-warping distortion to the video. We show two applications for this cross-media synchronization; one where a given dance performance is enhanced to be better synchronized with its original music, and one where a given dance video is automatically adapted to be synchronized with different music.

Keywords video processing, synchronization, motion segmentation, video analysis.

1 Introduction

Dancing is a universal phenomenon, which crosses cultures, gender and age. Dancing is even observed in some animals in the wild. We all appreciate and enjoy good dancing, however an interesting question is what makes a dancing performance look good, or can we enhance a dancing performance as observed in a video? One critical aspect of a good dance performance is moving in-rhythm, that is, a good synchronization between the dancing movement and the music [15]. In this paper, we introduce a video post-processing technique that enhances the rhythm of a dancing performance so that the dancing movements are more cohesive with the music.

Studies dealing with dance analysis such as Kim et al. [8], Shiratori et al. [17] and most recently Chu and Tsai [2] agree that dance key poses occur when there are stops or turns in the performer's movement trajectories. Furthermore, dance movements or generally human movements can be segmented into primitive motions [5]. These key poses in the dance motion are said to be the motion-beats. In the presence of music, the *motion-beats* should synchronize with the rhythm of the music as defined by the *music-beats* [7, 8, 17].

In the last decades interest in techniques that post process images and videos has increased, particularly in such techniques that attempt to enhance a subject captured in an image or a video, such as the works of Levyand et al. [9] and Zhou et al. [25]. Enhancing dance performance in our work is applied at the frame level without manipulating the context of the frames. The dancing performance as observed in a video is analyzed and segmented into motion intervals delimited by the motion-beats. The rhythm of the dance is enhanced by manipulating the temporal domain of each motion interval using a dynamic time warping technique which resembles the work of Zhou et al. [23, 24]. The challenge is twofold: (i) extracting the motion-beats from the video sequence; and (ii) defining and optimizing an objective function for synchronizing the motion-beats with the music-beats.

To extract the motion-beats from the video sequence we analyze the pixel-level optical flow of the frames in the video; unlike the work of Chu and [2] where motion-beats are tracked by an object-space analysis, here the analysis is applied in image space, bypassing the difficulties incurred by tracking objects in videos. We analyze the optical flow across a range of frames to detect significant changes in directions, which indicate where the key poses in the video are. Our method is

[1] Tel Aviv University, Tel Aviv 6997801, Israel. E-mail: rachele.bellini@gmail.com, yanirk@gmail.com, dcor@tau.ac.il

Manuscript received: 2014-12-31; accepted: 2015-01-30.

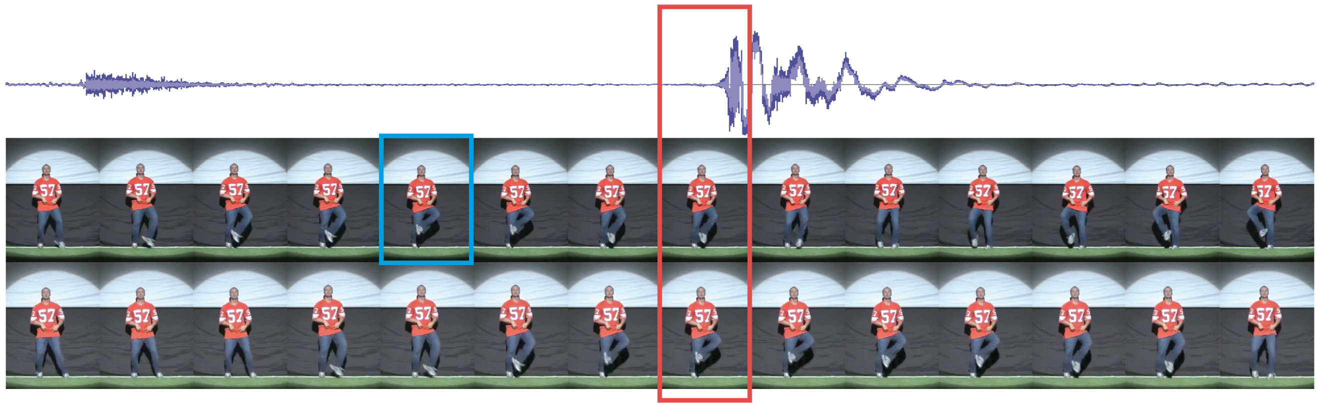

Fig. 1 A sequence of frames is displayed from both the input video (first row) and output video (second row). The music signal is displayed above, and the music-beat is marked by a red frame. The output video is synchronized, while in the input video the motion-beat, marked by a blue frame, occurs 3 frames before the music-beat.

capable of detecting fast changes in direction, as well as gradual changes and long pauses in the dance which are followed by a change in direction.

Given the music and motion beats streams, we find a mapping between motion-beats and music-beats that would minimize the time-warping distortion of the video. We introduce an objective function that aims at maximizing the number of good matches between motion-beats and music-beats. This optimization is constrained in the sense that matches are not enforced; only good matches are allowed, which may leave some motion-beats unmatched. Using the mapping between motion-beats and music-beats, we adapt the video to the music using a time warping technique. That is, stretching or contracting each segment of the video sequence such that the motion-beats are aligned with the music-beats they are mapped to.

We show two applications for this cross-media synchronization; one where a given dance performance is enhanced to be better synchronized with its original music, and one where a given dance video is automatically adapted to be synchronized with different music. A blind user study was conducted, in which users were requested to compare the input and output videos, and decide which of them is better synchronized with the music. The results show improvement for most of the tested videos. The user study is explained in detail in the Experimental Results section. All of our input and output videos can be found on the supplemental material or the online project page of this paper, at `http://sites.google.com/site/dancetothebeatresults`.

2 Related work

Synchronizing Video and Music. Recent years have seen a vast increase in the availability of digital video and audio media. This has led to a rising interest in video editing techniques for researchers and end users alike. Several recent works have been dedicated to the concept of synchronizing video files with audio files to create music videos or enhance the entertainment value of a video. Yoon et al. [21] create a music video by editing together matching pairs of segmented video clips and music segments in a sequence. The matching is performed based on features such as brightness and tempo which are extracted from the video and audio signal. A previous work [22] matches a given video with generated MIDI music that matches the motion features of the video. The motion features are tracked with the help of user interaction. Jehan et al. [6] presented an application that creates a music video for a given music by using pre-processed video clips with global speed adjustment as required to match the tempo of the music.

Recent years has seen a vast increase in the availability of digital video and audio media. This has lead to a rising interest in video editing techniques for researchers and end users alike. Several recent works are dedicated to the concept of synchronizing video files with audio files to create music videos or enhance the entertainment value of a video. Yoon et al. [21] create a music video by editing together matching pairs of segmented video clips and music segments in a sequence. The matching is performed based on features such as brightness and tempo which are extracted from the video and audio signal. A previous work [22] matches a given video with generated MIDI music that

matches the motion features of the video. The motion features are tracked with the help of user interaction. Jehan et al.[6] presented an application that creates a music video for a given music by using pre-processed video clips with global speed adjustment as required to match the tempo of the music.

The more recent work of Chu and Tsai [2] extracts the rhythm of audio and video segments in order to replace the background music or create a matching music video. After the rhythm is extracted from the audio and video segments, each audio segment is matched with a video segment shifted by a constant time offset that produces the best match for the music. This method works well for the creation of a music video where a large set of video clips or audio clips is available. However, if only a single video clip and a single audio clip are available the matching is inexact. In contrast, our method processes a single video segment by locally warping its speed along the video, to produce a natural looking video that matches the given music beat by beat.

In a very recent work, Suwajanakorn et al. [19] present a method to morph Obama's facial movements, matching them with an audio track. A neural network is trained to find the best matching between mouth shapes and phonemes, while the head movements are warped using dynamic programming. While the results they achieved are impressive, their goal and end results are quite different than the work presented here, which is focused on matching a given video to the soundtrack without altering its content.

Other works involve synchronization of a pair of video sequences or motion capture data [1, 18, 20, 23, 24]. They extract a feature vector from each frame and align the dense signals, as opposed to a sparse stream of motion-beats like in our work. Zhou et al. [23] allow aligning feature sequences of different modalities, such as video and motion capture; however their technique assumes a strong correlation between the sequences, such as a pair of sequences describing the same actions in video and motion capture streams. To synchronize the two streams, these methods allow altering the speed of both sequences. Humans tend to be less tolerant towards irregularities in music rhythm than in video appearance, therefore in our method the music stream is kept intact to avoid an irregular and unnatural sounding audio. Furthermore, we also enforce additional constraints on the video to make sure we do not allow noticeable changes. Wang et al. [20] find the optimal matching among two or more videos based on SIFT features, warping segments of the videos if necessary. While this kind of features produce good results to align videos with similar content, it cannot be used in video and music synchronization, since it does not capture temporal events.

Lu et al. [10] present a method to temporally move objects in a video. Objects of interest are segmented in the video, and their new temporal location is found optimizing the user's input with some visual contraints. This kind of approach, however, is focused on the overall movement of a single object object, approximating it to a rigid body. In our method, instead, we aim to process complex movements, such as objects where different parts have different motions (e.g. a person dancing).

Beats Extraction. Our method is based on synchronizing motion beats to music beats. First we detect the motion beats from the video and music beats from the audio. There is some body of work regarding motion beats detection from motion capture data; works such as Kim et al. [8] and Shiratori et al. [16, 17] detect motion beats from motion capture data to analyze dance motion structures, in order to synthesize a novel dance or create an archive of primitive dance moves.

Detecting motion beats from a video is a more challenging task, especially for amateur or home-made videos which typically are contaminated with significant noise, moving background objects, camera movement or incomplete view of the character. Chu and Tsai [2] track feature points in each frame and build trajectories of feature points over time. Motion beats are then detected as stops or changes in direction in the trajectories. Denman et al. [3] find motion clusters and use them to detect local minimas of the amount of motion between frames.

Extraction of music-beats, or *beat tracking*, is a well researched problem in the field of audio and signal processing. The aim of beat tracking is to identify instants of the music in which a human listener would tap his foot or clap his hands. We only reference a few representative works here. McKinney et al. [11] provide an overview and evaluation of several state of the art methods for beat tracking. These methods usually discover the tempo of the music, and use it to extract music beats that are well matched to the tempo.

The work of Ellis [4] provides a simple implementation that uses dynamic programming after the tempo is discovered to produce a series of beats which best match the accent of the music. We use the code provided by the author for our music beat tracking.

3 Motion-beat Extraction

The motion-beats of a dance are defined as the frames where a significant change in direction occurs. Often, there is a motion stop for a few frames in between a significant change in direction. Therefore, to detect a motion-beat we consider a non-infinitesimal time range that consists of a number of frames. A *direction-change score* is defined for every frame, and computed over multiple time ranges to detect direction changes of various speeds. A low score relates to a strong change in direction. Motion-beats are then detected as local temporal minima of the frame-level score function, ensuring the detection of the most prominent changes in direction over time.

3.1 Super-pixel Optical Flow

Extracting the motion-beats of a large variety of videos is a challenging task. The video can be of low quality and low resolution, such as one captured by a smartphone. We assume the input video consists of fast motions of an object over a natural background. However, the motions of the object do not necessarily follow a fixed rhythm. Since the moving object is also not necessarily human, we do not operate in object-space, but in image space.

Our technique is based on analyzing the optical flow of the video. In particular, we use the Matlab Horn-Schunck optical flow technique. Since the video may be noisy, we first apply a median filter and consolidate the optical flow by considering spatio-temporal super-pixels. Each super-pixel is a 5×5 block of pixels over five frames. We assign each pixel with the mean of pixel-level optical flow over the spatio-temporal super-pixel centered around it. The super-pixels optical flow values are considerably less noisy, yet still fine enough to faithfully measure the local motion direction. It should be noted that the motion analysis is in fact meant to identify the motion-beats at the frame level rather than at the pixel-level.

The super-pixel optical flow of four frames is illustrated in Figure 2(b), where each color depicts a different direction; blue for left and down, cyan for left and up, red for right and down, and yellow for right and up.

3.2 Frame-level Motion-beats

The frame-level motion-beat extraction is applied at the window-level. The frames are subdivided into a grid of 8×8 windows, assuming that in each window only one object is moving. Larger windows, or the entire frame for that matter, may contain multiple objects with opposing directions, for example a symmetrical movement of the hands. Such motion would nullify the average direction of movement. Within a small window only one object is moving and the average direction has a meaningful value. The average directions of the windows are shown in Figure 2(c), with the same color encoding as Figure 2(b).

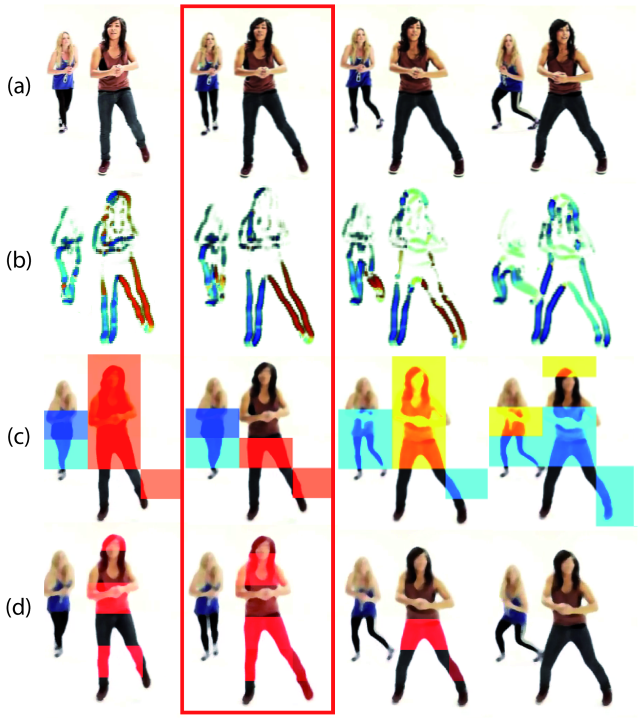

Fig. 2 Motion-beat Extraction. The original video (a) is displayed along with (b) its super-pixel optical flow, (c) the average motion direction of each window and (d) the average direction change. The detected motion-beat is marked with a red outline.

We define a direction-change score function as the dot product between the normalized average motion vectors of two frames; the dot product is maximal where the motion vectors are of the same direction and minimal when they are of exact opposite directions. We compute

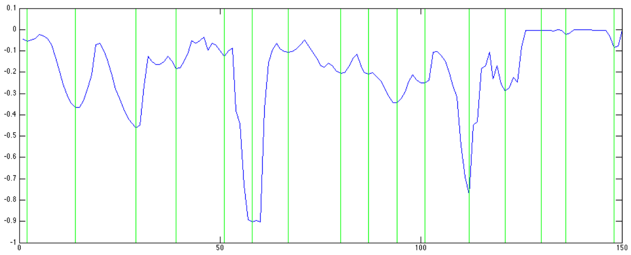

Fig. 3 Frame-level direction-change score for every frame in the video, along with the detected local minima of the function, marked by the green vertical lines.

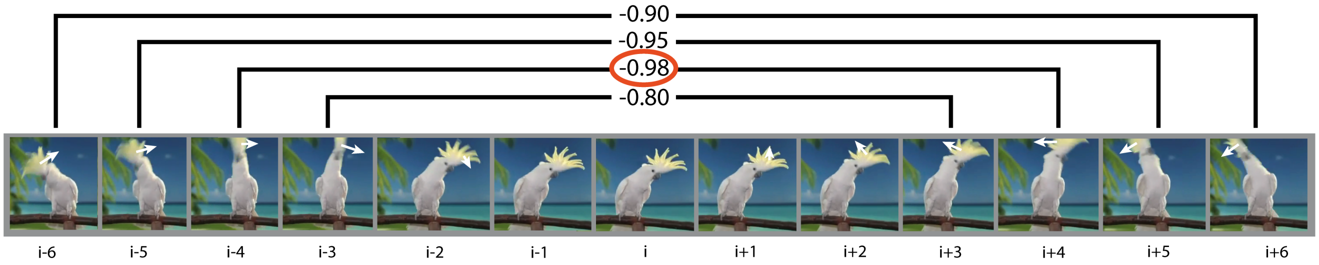

Fig. 4 An illustration of the temporal window in which pairs of frames are compared. The white arrows mark the approximate direction in which the parrot's head is moving. The dot product is computed for each pair of frames and the minimum is chosen as the direction-change score of that frame. Note that the dot product is actually computed separately for each window in the 8×8 grid.

the score function at the window level. To reduce temporal noise in the direction of motion, the average direction is first smoothed over time using a Gaussian filter. Since a change in direction in natural movement is often gradual, we identify changes in direction that occur at varying speeds. We compare pairs of frames at various intervals centered around each frame in the video. The score function for every frame is the minimum score among all intervals that are centered around that frame. Figure 4 shows a temporal window around frame i. Frame $(i-6)$ is compared with frame $(i+6)$, frame $(i-5)$ with frame $(i+5)$ and so on. The dot products for each pair of frames is shown above the video frames. The change of direction score for frame i will be the minimum between all pairs, which is reached in this case for the pair $\{i-4, i+4\}$.

The measurements described above provide a score for the change of direction of every spatial window, where a low value suggests a strong change. The frame-level score simply takes the minimal score among all windows, as means of detecting local changes of direction over all areas of the frame. The global score for every frame in the video is displayed in Figure 3, along with the detected local minimas of the function, marked by the green vertical lines. The local minimas are computed by finding the strongest negative peaks of the function that are at least five frames apart from each other. Figure 2(d) shows the color-coded score of the spatial windows in four frames. The negative values are shown in red, where a stronger color marks a lower score, closer to -1. A motion-beat is detected in the second frame from the left, which is the local temporal minimum of the frame-level change of direction score. Note that these scores are computed using a larger temporal window which is not shown in the figure.

4 Synchronization

The detected motion-beats effectively divide the video into a series of *motion intervals*, each starting one frame after a motion-beat and ending at the next motion-beat. *Music intervals* can be defined similarly using the music-beats. When a motion-beat occurs on the same frame as a music-beat, the dancing object appears to be in sync with the music. Our goal is thus to adjust the speed of each interval such that as many motion-beats as possible occur on the same frame as music-beats, with as little distortion as possible for each interval.

The output of the audio and video analysis is two sequences: the music-beat times A_i and the motion-beat times V_j. When two beats are matched, the relevant motion interval is stretched or contracted such that the motion-beat occurs simultaneously with the music-beat. The music signal however is fixed and does not change; if the movements of the performer become slightly faster or slower, the final video can still be perceived as natural, while even the smallest modification to the music may be perceived as unpleasant. Respecting these asymmetric properties of the video and music streams, we differ from previous solutions such as Zhou et al. [23, 24] and solve an optimization problem which better fits the task at hand.

To compute the optimal mapping between motion-beats and music-beats, a time-warping distortion of every matched pair is measured. Naturally, not every motion-beat has a matching music-beat and vice versa. However, the mapping is monotonic, i.e. every matched motion-beat is at a later time than the previous matched motion-beat, and likewise for music-beats. Therefore, to compute the time-warping distortion incurred by a matched pair, it is necessary to accumulate all the motion-intervals and music-intervals, respectively, between the previously matched beats and the currently matched beats. We compute a *compatibility score* function of the accumulated intervals, which is high when the time-warping distortion between the accumulated intervals is low. For further details see section 4.1.

The global optimization problem is then to find a mapping which maximizes the total compatibility score of all the pairs it consists of. Such maximization encourages as many matched beats as possible, as long as they do not force a large time-warping distortion over other matched pairs. The local time-warping distortions between each two pairs are also subject to user-defined constraints. In our setting, the user has control and can define separate constraints for speeding up the video and for slowing it down, since different videos tolerate different amounts of speed changes, while still retaining their natural appearance, according to their original content. Once the beats are analyzed and matched, the video is processed by contracting or stretching motion intervals in order to have the motion-beats occur simultaneously with the music beats they are matched with. The output is the processed video which is a time-warped version of the original video that fits the given music.

4.1 Compatibility Score

The compatibility score function is defined for every possible monotonic mapping between the motion-beats and music-beats. It measures the total amount of distortion over all pairs of matched beats. A high compatibility score means low distortion, so maximization of the compatibility score encourages as many matching beats as possible, as discussed above.

Formally, for every music-beat A_i and motion-beat V_j we define the length in frames of their corresponding intervals:

$$l_a(i) = A_i - A_{i-1}, \qquad l_v(j) = V_j - V_{j-1},$$

where i is the index of the current aufio beat and j is the index of the current video beat. We will use i and j similarly throughout this section. For the first intervals:

$$l_a(1) = A_1, \qquad l_v(1) = V_1.$$

Every mapping is a sequence of pairs $\{i^{(m)}, j^{(m)}\}$, $1 \le m \le k$, where for every m the music-beat $A_{i^{(m)}}$ is matched with the motion-beat $V_{j^{(m)}}$.

For clarity, we define the aggregations of several intervals between two beats as:

$$L_a(i_1, i_2) = \sum_{i_1 < i \le i_2} l_a(i), \qquad L_v(j_1, j_2) = \sum_{j_1 < j \le j_2} l_v(j).$$

Note that i_1 and j_1 can be zero, in which case the aggregation starts from the first interval. The compatibility score C of an arbitrary pair of sequences starting from $\{i_1, j_1\}$ and ending at $\{i_2, j_2\}$ is then:

$$C(i_1, i_2, j_1, j_2) = \frac{\min\{L_a(i_1, i_2), L_v(j_1, j_2)\}}{\max\{L_a(i_1, i_2), L_v(j_1, j_2)\}}.$$

The score $C(i_1, i_2, j_1, j_2)$ of of two pairs of beats $\{i_1, j_1\}, \{i_2, j_2\}$, indicates the distortion incurred by this sequence of intervals which includes no other matching pair in between. The score function values are within the range $[0, 1]$. A high value indicates low distortion while a perfect match where the video is neither stretched or contracted yields a value of one. We can now solve the global optimization problem which is the maximization of the total score of all pairs associated with a mapping.

4.2 Dynamic Programming Solution

The sequence with a maximal score up to any pair of beats $\{A_i, V_j\}$ can be computed regardless of any matched pairs that are selected afterwards. We can thus solve the optimization by using dynamic programming.

We define a global score function $F(i, j)$ that aggregates the values of the compatibility score function for sequences up to pair $\{Ai, Vj\}$. The maximization step for each pair is then

$$F(i, j) = \max_{\substack{1 \le k \le i \\ 1 \le l \le j}} \{F(i-k, j-l) + C(i-k, i, j-l, j)\},$$

with initial values

$$F(0, j) = 0, \qquad F(i, 0) = 0.$$

During the computation of each pair the distortion constraints are tested, and pairs that violate the constraints are discarded from the computation. Since the maximization of F at each step is dependant on $i \cdot j$ values, the theoretic complexity of the solution is $O({n_a}^2 \cdot {n_v}^2)$, where n_a is the number of music-beats and n_v is the number of motion-beats, or $O(n^4)$ if n_a and n_v are both about n.

However, many of the possible values can be ignored in practice. Between two consecutive matched pairs, it is unlikely that several beats in a row will be left unmatched for both music-beats and motion-beats. It is possible that a corresponding pair of motion-beat and music-beat are both left unmatched if their distortion score is violating the constraints. However, due to the regularity of the music beats and the minimum distance between motion-beats, it is not probable that the difference between consecutive matched pairs will be several motion-beats *and* several music-beats. It is far more likely that some motion-beats are left unmatched since they are between two consecutive music-beats

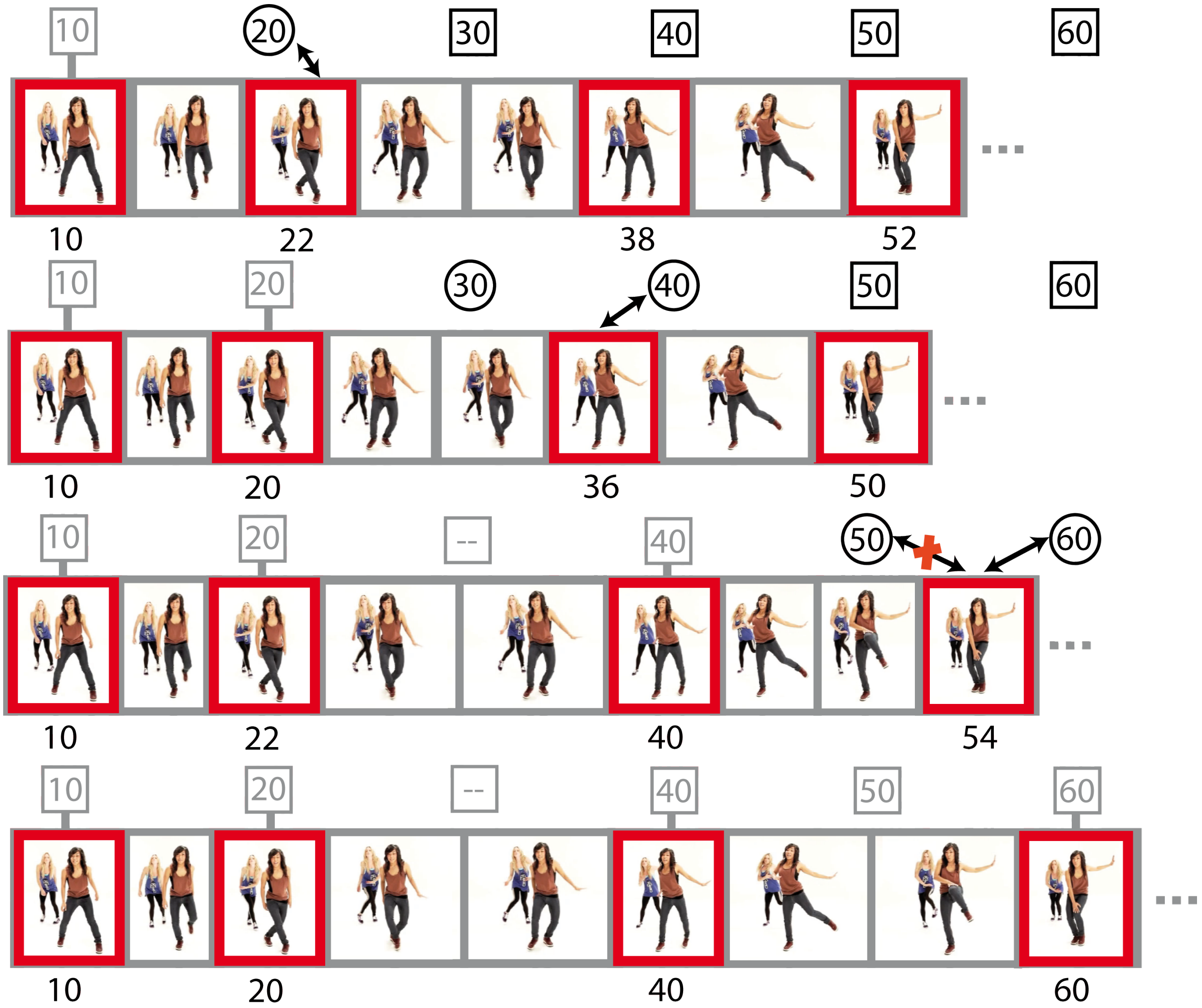

Fig. 5 An example of the synchronization process. See section 4.2 for details.

or vice versa. Hence, it is sufficient to maximize $F(i,j)$ over the narrow band: $\{i',j-1\},\{i',j-2\},\{i-1,j'\},\{i-2,j'\}$, where $1 \le i' < i, 1 \le j' < j$. This requires a time complexity of $O(n_a+n_v)$ for each pair $\{i,j\}$. The total complexity of such a solution is $O(n_a \cdot n_v \cdot (n_a+n_v))$, or $O(n^3)$ if n_a and n_v are both about n.

An example of the synchronization process is shown in Figure 5. The synchronization is viewed as a sequential process for clarity. The top row shows the original music-beats and motion-beats of the video sequence. The music-beats are marked in grey squares for beats that are already matched, and black squares if they are yet unmatched. Motion-beats in the video are marked by red highlights. In the second row, the second music-beat at frame 20 is matched with the second motion-beat at frame 22. The video is thus pushed backwards by two frames and the position of the next motionbeats is modified accordingly. Note that the score of each matched pair is based on the distance between two consecutive beats and not on the absolute position of the beat. The fourth motion-beat is now at frame 54 which is closer to 50 than to 60. The score of matching the beat to frame 50 is also a bit higher. However, matching the motion-beat to frame 50 would require shortening the video segment by 40%. The user selected to constrain the speed ratio of the video to no more than 25% increase of speed, therefore this possible match is discarded. On the other hand, the constraint for stretching the video is more loose, since slowing down the video by a certain ratio looks more natural to the viewer than speeding it up by the same ratio. Therefore, the beat is matched to the music-beat at frame 60 and the video is stretched accordingly. If the user would select to constrain the stretching of the video to a lower ratio as well, this motion-beat would not be matched to a music-beat, and the next motion-beat would have been considered.

5 Experimental Results

The performance of the presented method was evaluated on a number of videos, which represent a variety of dancing types, motions, rhythms, environments, and subjects. The videos are also varied in quality: some videos are of commercial production levels, while other videos are of home video quality, captured with a hand held camera or even a smartphone, and contain a lot of noise. Figure 6 shows a few frames from each video to capture its essence. For further evaluation, we refer the reader to the accompanying videos and the online project page mentioned above, where all of the examples can be found and evaluated. We experimented with two types of applications as described in the following.

Performance Enhancement. Here we consider a video containing a dance or rhythmical movements, which do not match with the background music exactly. The original video is then edited by our method, and enhances the dancing to better fit the beats of the original music, that is, the motion-beats match with the music-beats.

First, we present two rather simple examples, where the enhancement can be clearly appreciated. The performing subject in the first example is a well-known parrot named Snowball, a Sulphur-crested Cockatoo whose renowned dancing skills have been studied and described in several academic papers [12–14]. Although the parrot has an extraordinary ability to move according to the music beat, its performance is still imprecise. Snowball is famous enough to have been cast for a Taco Bell commercial in 2009, where he dances along with the song ”Escape (the Pia Colada song)” by Rupert Holmes. Some of the movements of the parrot in the video are irregular, so some of the motion-beats are not synchronized with the music-beats. As can be observed, our method modifies the video so that the movements become more rhythmical and better synchronized with the given song.

The subject of the second video is another type of parrot, Bare-Eyed Cockatoo, which dances according to the song ”Shake a Tail Feather” by Ray Charles. The parrot does not have a constant rhythm, and misses the music-beats quite often. Our method greatly enhances its performance.

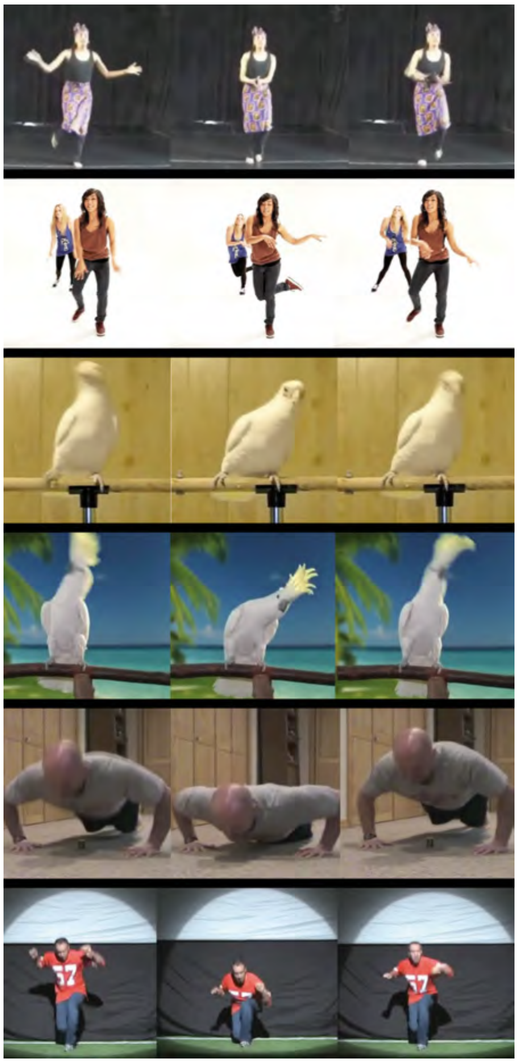

Fig. 6 Typical frames from each input videos. The center frame of each video is taken from a motion-beat. Note that we handle videos with moving camera (such as the first video) and significant motion blur (such as the parrots).

The third example is of a human dancer performing in front of a live audience. The segment is a part of a performance by Judson Laipply called ”The Evolution of the Touchdown Dance”, which includes memorable NFL touchdown dances. The dancer keeps the beat pretty well in most parts, however in the second part of the video segment the contribution of our method is clear. It should be noted that our method does not interfere with the original video where it is well synchronized. Therefore, in this particular example, the first half of the output video looks almost the same as the input, while in the second half there is a noticeable difference.

Dance Transfer. The second set of experiments transfer a dance to new background music. The input dance is then modified to match the new rhythm and audio-beats of the new music. Again, we begin with a few simple and clear examples. The first one is a video of a man doing push-ups. After a while he gets tired and does not keep a constant rhythm in the original video. This is particularly evident around the 15th second. We synchronized it with the song ”Where is the love” by The Black Eyed Peas. The song has a slow rhythm, so the beginning is slowed down: the input video shows ten push-ups in the first ten seconds, and the output shows only eight. However the same rhythm is kept until the end on the video, and the man does not seem to get tired as in the original video. When synchronizing the same input video with the song ”Billie Jean” by Michael Jackson, the results are significantly different. The rhythm of the song is a bit faster, and thus the push ups happen at a faster rate of ten pushups in the first ten seconds. The rhythm of the push-ups is again more regular than in the input video. Note that since the rate of the push-ups is slower as the video progresses, a music-beat is skipped once in a while, and a motion-beat is matched with the next music-beat, while still maintaining the constant rhythm.

Another clear example is given by the same Taco Bell commercial used for the first type of experiments. This time the background song is changed to the song ”Trashin’ the Camp” by Phil Collins. In the original version the movements of the parrot do not fit with the rhythm of the music, since they are shifted in time and their rhythm is quite faster than the music rhythm. In the output video the movements are slowed down a little and time-shifted to match with the new music.

We also show a few experiments with videos of professional human dancers. The first experiment with a human dancer is a video of Caren Calder, a dance instructor, performing a traditional West African dance. We synchronize her dance to ”Oh! Darling” by The Beatles, which is a famous song of a very different culture. In the original video the dancer’s hand clapping and music-beats are not matched, while in the new version her movements and hand clapping occur on the beat of the music.

Another dance experiment is conducted again with footage from the performance of Judson Laipply, ”The Evolution of the Touchdown Dance”. We took two different segments of this performance. One has been synchronized to ”Billie Jean” by Michael Jackson, and the other to ”Pantala Naga Pampa” by Dave Matthews Band. With the first song the changes are very clear, since the original video does not match the music. With ”Pantala Naga Pampa”, similar to the experiment of the same video segment with the original music, the dancer’s movements match the music quite well, and the change in some parts of the dance is not easily noticeable. However there are a few changes: in the very beginning, in the original video the first jump occurs within the silence before the beginning of the song, while in the new version the first jump occurs on the first beat of the song. In the second part of the video the dancer jumps a few times with irregular rhythm. This part is significantly changed to better fit the new music.

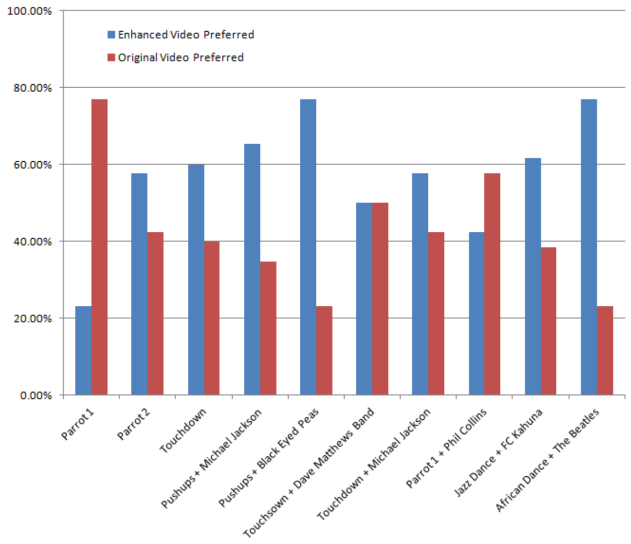

Fig. 7 User preferences when comparing the original version to the enhanced version of each video. The videos that were presented to the users are provided as supplemental material and appear online on the project page.

The last video we present here is a jazz-funk dance online video of the Dr Pepper Cherry YouTube Dance Studio. The original background music is replaced and the dance is synchronized with the song ”Hayling” by FC Kahuna. The original dance is quite fast, while the music we chose has a very low BPM (beats per minute). In this case, since the dancers keep a rather constant rhythm, the original performance fits almost perfectly to the new music. Hence the changes are minute and hard to perceive without careful observation. However, when the input and output videos are viewed side by side, the difference can be observed. The user study shows most people believe the output video is better synchronized with the music, even though the difference between them is small.

We refer the reader again to the supplemental material or the project page to view the example videos described above.

User Study. A blind user study was conducted for each of the videos we describe above, to which 26 users replied. Users were presented with pairs of videos, and were requested to rate which of the videos better matches the music, without knowing which video is the enhanced version and which is the original. The results of the user study are displayed in Figure 7. The first three videos, labeled *Parrot 1*, *Parrot 2*, and *Touchdown*, are examples of performance enhancement of two different parrots and a segment from ”The Evolution of the Touchdown Dance”. In these videos the background music remains the same as the original video. The rest of the videos are examples of music replacement.

As can be seen by the figure, most of the enhanced videos were preferred over the input videos. Our most successful examples were ones in which the dance performance deviates significantly from the rhythm of the song. The african dance video have a slow rhythm, and so does the song we matched it to. Therefore users can easily identify where a music-beat is being missed in the original performance, and it is easier to evaluate the differences between the videos. In the push-ups videos, which also received positive reviews, the person working out is getting tired and does not keep a constant rhythm, which again enables the user to clearly see where a music-beat is missed.

Conversely, for video segments which maintain a fast fixed rhythm, it is harder for the human mind to distinguish between different motion-beats and separate them from the rhythm of the attached music. It may require several viewings even for a professional video editor to determine whether a certain motion-beat occurs simultaneously with a music-beat. However, it can often be felt subconsciously whether a certain video is in sync with the music or not. An interesting example is the jazz dance performance; the rhythm of the dance is very fast, and the replaced music matches the original video quite well. Only minor corrections are applied by our synchronization algorithm, and the output video looks almost identical to the the original video, such that the differences between videos are only noticeable when the videos are viewed simultaneously side by side. Yet, in our user study more than 60% of the users preferred the enhanced video over the original.

A noticeable failure case shows that sometimes keeping the beat is not enough to present a good dance performance. In the video of the parrot Snowball, the head motions does not follow the beat particularly well. However, the parrot changes his dancing style noticeably as the chorus of the song begins. In our enhanced version, the individual motions of the parrot’s head are better synchronized with the music, but the change of dancing style occurs a few music-beats later than the beginning of the chorus. The users did not react well to that change and preferred the original performance of the parrot. A possible improvement of our method for such cases is to synchronize motion-beats to music-beats only within segmented non-overlapping sections of the video.

Implementation. Our algorithm is implemented in Matlab. We ran all the experiments on a Macintosh MacBook Pro platform with 2.4 GHz Intel Core 2 Duo processor and 4 GB of RAM, and the Mac OS X 10.6.8 operating system. The experiments were mostly run on videos of about 500 frames, or 20 seconds, with 640 × 360 resolution. The bottle neck of our implementation is the video editing itself, i.e. reading and writing frames from the video and the compression of the video. This could be vastly improved by using other video compressions (for example an image sequence) and faster editing methods. We estimate that computation of the algorithm without the video editing, i.e. detecting the motion-beats and music-beats and finding an optimal mapping between them, requires about two minutes for the videos described above.

Our algorithm is implemented in Matlab. We ran all the experiments on a Macintosh MacBook Pro platform with 2.4 GHz Intel Core 2 Duo processor and 4 GB of RAM, and the Mac OS X 10.6.8 operating system. The experiments were mostly run on videos of about 500 frames, or 20 seconds, with 640 × 360 resolution. The bottle neck of our implementation is the video editing itself, i.e. reading and writing frames from the video and video compression. This could be vastly improved by using other video compressions (for example an image sequence) and faster editing methods. We estimate that computation of the algorithm without the video editing, i.e. detecting the motion-beats and music-beats and finding an optimal mapping between them, requires about 2 minutes for the videos described above.

6 Conclusions and Future Work

We have presented a method to enhance of a dancing performance in a video. The key idea is to synchronize between the motion-beats and the music-beats so the dance rhythm correlates better with the given music. The detected motion-beats are used to delimit motion intervals. However, while music-beats are well understood, the notion of motion-beats is not yet established; there are other possible definitions for motion-beats, such as the center, rather than the end, of motion segments. How to segment motion in a video remains an open interesting problem. Clearly, segmenting motion of articulated bodies can be performed better in object-space. However, this requires identifying and tracking the skeleton of the dancing body, which is known to be a difficult task. In our work, we analyze the video frames directly, detecting motion-beats in image-space. The main advantage of our approach is that it is not limited to human bodies, or tailored to a particular subject known a priory. However, in cases of extreme camera movements some motion beats may not be detected. As mentioned in Section 3.1, we assume that in the input videos the foreground objects are moving in contrast to the background in higher speed. Yet, we consider integrating object-space analysis and learning whether it can significantly improve the segmentation of the dancing subject.

Another interesting direction for future work is intra-frame enhancement of videos containing more than a single dancing performer. Here we can use the extracted motion-beats, and synchronize two or more motions in the video with each other as well as with the background music. In this case, the correct synchronization will be more evident than in a cross-media synchronization.

The motion-beats we detect can also be helpful outside the scope of synchronization; one such possible application is global time remapping of a video. When stretching or contracting a video, deformation can be concentrated around the motion-beats. Video segments where continuous motion occurs would retain their natural speed, while motion stops, or segments where there is a change in directions, would be prolonged or shortened. Since the speed of motion is low in such segments, the change of speed would be less noticeable and more natural looking.

References

[1] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1409–1424, 2002.

[2] W.-T. Chu and S.-Y. Tsai. Rhythm of motion extraction and rhythm-based cross-media alignment for dance videos. *IEEE Transactions on Multimedia*, 14(1):129–141, 2012.

[3] H. Denman, E. Doyle, A. Kokaram, D. Lennon, R. Dahyot, and R. Fuller. Exploiting temporal discontinuities for event detection and manipulation in video streams. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 183–192. ACM, 2005.

[4] D. P. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.

[5] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985.

[6] T. Jehan, M. Lew, and C. Vaucelle. Cati dance: self-edited, self-synchronized music video. In *ACM SIGGRAPH 2003 Sketches & Applications*, pages 1–1. ACM, 2003.

[7] M. R. Jones and M. Boltz. Dynamic attending and responses to time. *Psychological review*, 96(3):459, 1989.

[8] T.-h. Kim, S. I. Park, and S. Y. Shin. Rhythmic-motion synthesis based on motion-beat analysis. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 392–401. ACM, 2003.

[9] T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski. Digital face beautification. In *ACM Siggraph 2006 Sketches*, page 169. ACM, 2006.

[10] S.-P. Lu, S.-H. Zhang, J. Wei, S.-M. Hu, and R. R. Martin. Timeline editing of objects in video. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1218–1227, 2013.

[11] M. F. McKinney, D. Moelants, M. E. Davies, and A. Klapuri. Evaluation of audio beat tracking and music tempo extraction algorithms. *Journal of New Music Research*, 36(1):1–16, 2007.

[12] A. D. Patel, J. R. Iversen, M. R. Bregman, and I. Schulz. Experimental evidence for synchronization to a musical beat in a nonhuman animal. *Current biology*, 19(10):827–830, 2009.

[13] A. D. Patel, J. R. Iversen, M. R. Bregman, and I. Schulz. Studying synchronization to a musical beat in nonhuman animals. *Annals of the New York Academy of Sciences*, 1169(1):459–469, 2009.

[14] A. D. Patel, J. R. Iversen, M. R. Bregman, I. Schulz, C. Schulz, and C. San. Investigating the human-specificity of synchronization to music. In *Proceedings of the 10th International Conference on Music and Cognition. Sapporo, Japan*, pages 100–104, 2008.

[15] B. H. Repp. Musical synchronization. *Music, motor control and the brain*, pages 55–76, 2006.

[16] T. Shiratori, A. Nakazawa, and K. Ikeuchi. Detecting dance motion structure through music analysis. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 857–862. IEEE, 2004.

[17] T. Shiratori, A. Nakazawa, and K. Ikeuchi. Dancing-to-music character animation. In *Computer Graphics Forum*, volume 25, pages 449–458. Wiley Online Library, 2006.

[18] M. Slaney and M. Covell. Facesync: A linear operator for measuring synchronization of video facial images and audio tracks. In *Advances in Neural Information Processing Systems*, pages 814–820, 2001.

[19] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017.

[20] O. Wang, C. Schroers, H. Zimmer, M. Gross, and A. Sorkine-Hornung. Videosnapping: Interactive synchronization of multiple videos. *ACM Transactions on Graphics (TOG)*, 33(4):77, 2014.

[21] J.-C. Yoon, I.-K. Lee, and S. Byun. Automated music video generation using multi-level feature-based segmentation. In *Handbook of Multimedia for Digital Entertainment and Arts*, pages 385–401. Springer, 2009.

[22] J.-C. Yoon, I.-K. Lee, and H.-C. Lee. Feature-based synchronization of video and background music. *Advances in Machine Vision, Image Processing, and Pattern Analysis*, pages 205–214, 2006.

[23] F. Zhou and F. De la Torre. Generalized time warping for multi-modal alignment of human motion. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1282–1289. IEEE, 2012.

[24] F. Zhou and F. Torre. Canonical time warping for alignment of human behavior. In *Advances in neural information processing systems*, pages 2286–2294, 2009.

[25] S. Zhou, H. Fu, L. Liu, D. Cohen-Or, and X. Han. Parametric reshaping of human bodies in images. In *ACM Transactions on Graphics (TOG)*, volume 29, page 126. ACM, 2010.

Rachele Bellini Rachele Bellini received the BSc degree cum laude in Digital Communication (Computer Science dept) in 2012 and the MSc degree cum laude in Computer Science in 2015, both from the University of Milan. Since 2014 she is collaborating with Cohen-Or's group at the Tel Aviv University on texturing, image processing and video analysis. She is currently working as a VFX Developer at Pixomondo LLC (Los Angeles).

Yanir Kleiman obtained his PhD from Tel Aviv University in 2016, under the supervision of Prof. Daniel Cohen-Or. He was a post-doc at cole Polytechnique in France in 2016-2017. He is currently a graphics software developer at Double Negative Visual Effects. His research focused on shape analysis, including shape similarity, correspondence and segmentation, and includes some work in other domains such as image synthesis, crowd sourcing and deep learning. Prior to his PhD, Yanir had more than 10 years of professional experience as a software developer and a visual effects artist.

Daniel Cohen-Or is a professor at the School of Computer Science, Tel-Aviv University. He received the BSc (cum laude) degree in mathematics and computer Science and the MSc (cum laude) degree in computer science, both from Ben-Gurion University, in 1985 and 1986, respectively. He received the PhD from the Department of Computer Science at State University of New York at Stony Brook in 1991. He received the 2005 Eurographics Outstanding Technical Contributions Award. In 2015, he was named a Thomson Reuters Highly Cited Researcher. Currently, his main interests are in few areas: image synthesis, analysis and reconstruction, motion and transformations, shapes and surfaces.