This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAFFC.2015.2445335, IEEE Transactions on Affective Computing

A Method for Automatic Detection of Psychomotor Entrainment

Tariq Iqbal and Laurel D. Riek, *Senior Member, IEEE*

Abstract—Group interaction is an important aspect of human social behavior. During some group events, the activities performed by each group member continually influence the activities of others. This process of influence can lead to synchronized group activity, or the entrainment of the group. Understanding entrainment is important, because it can be a critical behavioral indicator of group cohesiveness, and can provide context for accurately understanding a group's affective behavior. In this paper, we present a novel method to automatically detect group psychomotor entrainment, which takes multiple types of discrete, task-level events into consideration. We experimentally validated the method on two synchronous rhythmic activities, "the cup game" and a marching task. We also compared its accuracy against two alternate synchrony detection methods in the literature. The results suggest our method can successfully measure group psychomotor entrainment, and is more accurate compared to other methods. This method will be useful to researchers interested in quantitatively and automatically measuring entrainment, and can also provide insight into understanding how groups interact socially.

Index Terms—Social signal processing, synchrony, entrainment, joint action, computational group modeling

1 Introduction

GRoup interaction is an important aspect of human social interaction, and is an active area of research in social psychology, communication, and cognitive science [1]–[6]. In group interaction, the individual activity of each member continually influences the activity of other group members. Most groups create a state of interdependence, where each member's outcomes and actions are determined in part by other members of the group [6]. This process of influence can result in coordinated group activity over time, which can be described as the synchronization, or entrainment, of the group.

Entrainment is described as the spatiotemporal coordination of signals, which results from rhythmic responsiveness to a perceived rhythmic signal [7]. It can occur in many group contexts, such as dance, music, and games, and can be a result of both intentional and unintentional motor coupling during social interaction. *Intentional entrainment* occurs in cooperative group tasks, such as rowing a boat with a team. In contrast, an example of *unintentional entrainment* would be a group of friends spontaneously coordinating their movements when walking together [8].

Clayton et al. [9] describe entrainment as "the process whereby two rhythmic processes interact with each other in such a way that they adjust towards and eventually 'lock in' to a common phase and/or periodicity", and suggest that two basic components must be involved. First, there must be two or more autonomous rhythmic processes or oscillators. And second, the rhythmic processes or oscillators must interact. According to Clayton et al., the interaction between the rhythmic processes can be described as entrainment if the system re-establishes the synchronization after any perturbation.

In a group setting, the entrainment phenomena can be observed not only based on the coordinated movements of the group members, but also on how the group members perform activities. If a group task is cooperative and rhythmic (e.g., synchronous swimming), then entrainment can also be found in the activity performed by an individual member of the group. Therefore, it is important to take a systemic approach to modeling group entrainment, at both the group-level, individual-level, and task-level.

This leads us to explore several research questions. First, can we automatically measure the overall entrainment of a group while taking multiple types of task-level events into account simultaneously? Second, will such an automatic entrainment measure be comparable to group members' own perceptions of entrainment? Third, can such a measure also be used to estimate asynchronous behavior?

Addressing these research questions is important, because synchrony[1] is a crucial parameter in any social interaction. Group level synchrony detection may be an important behavioral indicator of group level cohesiveness, as well as accurately understanding the affective behavior of a group [8], [10]. Understanding how individuals contribute to synchrony can help us perceive group dynamics more effectively.

• *The authors are in the Computer Science and Engineering Dept., University of Notre Dame, USA. E-mail: tiqbal@nd.edu, lriek@nd.edu.*
The authors thank James Delaney, Giovanna Varni, Moreno Coco, Maria O'Connor, and Lucinda Krahl.

1. *We will use the term synchrony to represent entrainment.*

To address these questions, we have developed a new method to automatically detect and model psychomotor entrainment in groups. In contrast to other methods, our method takes multiple types of events into account, and is able to detect both asynchronous conditions and also model "non-events" in time. Furthermore, it is able to work with non-periodic time series data as it estimates entrainment.

We validated the method in four ways. First, in Section 4.1, we validated the method by applying it to a multiple event-based rhythmic game, where each player performs a periodic psychomotor activity within the group, and each player's movements influence all others' movements. Our method was able to model both synchronous and asynchronous activity, and was well-matched to the players' perceptions of entrainment. Second, in Section 4.2, we compared our results to another method from the literature that uses single event types, and found our multiple event-based method to be more accurate in estimating entrainment. Third, in Section 4.3, we compared our results to the cross-recurrence quantification analysis (CRQA) method. The results again suggested that our multiple event-based method is more accurate in estimating entrainment.

Finally, in Section 4.4, we validated our method by using it to study people engaging in synchronous and asynchonous marching tasks while being followed by autonomous mobile robots. We again found our method accurate in measuring entrainment.

In Section 5, we discuss how our approach can benefit the affective computing community and others who design intelligent interactive systems. This work will enable the community to more accurately understand group behavior and estimate its cohesion, a critical behavioral indicator of group affect [11].

2 Methods for Measuring Synchrony in Dynamic Systems

Many disciplines have approached the problem of how to assess synchrony in a system. These include: physics, neuroscience, psychology, dance, and music [9], [12]–[24]. Recently, researchers in computational fields such as social signal processing and robotics have also become interested in this problem [25]–[27]. Across all fields, two types of measurement methods exist. The first is best suited for continuous time series data, e.g., recurrence analysis, correlation-based methods, and phase-based approaches. The second type is well-suited for discrete events in time series data, e.g., event-based methods. We will now describe each of these methods.

2.1 Measuring synchrony in continuous time series data

Recurrence analysis is one of the most widely used methods to measure synchrony [20], [28], [29]. This approach is based on trajectory similarities in phase space. First proposed in physics, this analysis method was inspired by coupled dynamic systems. Recurrence analysis assesses how many times the state of a system visits close to a previous state [30].

The graphical representation of recurrence in dynamical systems is called a recurrence plot (RP) [31]. Its central idea is to plot a dot when a state is sufficiently close to any of its previous states. It is also a useful diagnostic tool to quantitatively analyze recurrences in a dynamic system. Structures and patterns found in an RP are closely linked to the dynamics of the underlying system [30]. The cross-recurrence plot (CRP) is the non-linear bivariate extension of the recurrence plot [30], [32]. Recurrence quantification analysis (RQA) is used to assess and diagnosis complex dynamic systems using RP and CRP [33], [34].

Varni et al. [10] used the RP and RQA measures, and presented a system for real-time analysis of nonverbal affective social interaction in a small group. In their experiments, several pairs of violin players were asked to perform while conveying four different emotions. RQA measured the synchronization of the performers' affective behavior. Konvalinka et al. [35] also used RQA for measuring the synchronous arousal between performers and observers during a Spanish fire-walking ritual. This synchronous arousal was derived from heart rate dynamics of the active participants and the audience.

Correlation is another approach used to evaluate synchrony among continuous time series data [36]. Typically a time-lagged cross-correlation is applied between two time series using a time-window. For example, Boker et al. [37] used the cross-correlation method to determine the symmetry building and breaking of body movements in synchronized dancing, and Quian Quiroga et al. [38] used it to measure the synchronization between left and right hemisphere rat electroencephalographic (EEG) channels.

Phase differences are another well-explored approach in the literature. Typically, phases are defined by Hilbert or wavelet transforms. For example, Richardson et al. [8] proposed a method to assess group synchrony by analyzing the phase synchronization of rocking chair movements. A group of six participants rocked in their chairs with their eyes either open or closed, and they used a cluster-phase method to quantify phase synchronization. Néda et al. [39] investigated synchronized clapping in a naturalistic environment. They quantitatively described the phenomena of how asynchronous group applause starts suddenly, and transforms into synchronized clapping.

2.2 Measuring synchrony in categorical time series data

While the aforementioned methods work well for measuring synchrony in continuous data, there are instances where it may be useful to have methods that work across categorical time series data, which may define discrete events.

Quian Quiroga et al. [40] proposed an event synchronization (ES) method for discrete events which can be used for any time series where events can be defined. This method is simple and fast, and the notion of phase is not required. As ES is based on the relative timing of events, it can also determine a leader-follower relationship between two time series, if one exists. The authors applied their method across two sets of time series data - EEG signals from a rat, and intracranial human EEG recordings taken during an epileptic seizure. In both cases, only singular types of events (i.e., local maxima of input signals) were taken into consideration.

Varni et al. [41] proposed an extension of this work to measure group synchronization. They authors described a system called I-DJ, which is capable of retrieving music content based on the interaction patterns (i.e., synchronous motion) in a group of dancers. The synchronization of the group, as well as a possible dominant person or a clique in that group, were measured using their proposed method from the dancers' body motion. However, they too only consider a single type of event when measuring group synchronization.

Dale et al. [42] presented a cross-recurrence analysis type method for quantifying the relationship between two time series of categorical data (CRQA), and Coco et al. [43] recently released an R package which provides an implementation of this and similar methods.

CRQA works adequately for consecutive, categorical data. However, when the data are sparsely distributed across time (i.e. non-periodic), there are often instances when no synchronous activity occurs. CRQA actually includes instances of these "non-events" as actual events while measuring synchrony, which can artificially inflate the true synchrony. ES-type methods do not have this problem; however, they are only able to incorporate a single type of event while assessing synchrony, which is limiting considering the inherent multimodality of events within human social interaction [44].

Thus, there is a significant gap in this space: there are cases where it may be important to detect synchrony across multiple types of events, and those events may be sparsely distributed. Our work addresses this gap, by use of an event-based method which can successfully take multiple types of discrete, task-level events into consideration, and successfully ignore the "non-events" while measuring the synchrony of the system.

3 PROPOSED METHOD FOR MEASURING SYNCHRONIZATION

During a group activity, multiple discrete, task-level events occur, and the outcome and timing of each event depends on the events preceding it. The overall synchronization of the system depends on all of these events. In our work, we are interested in modelling the whole group as a system, thus, our method for measuring synchrony incorporates multiple types of events together. To achieve this goal, we extend the ES method proposed by Quian Quiroga et al. [40], as well as the follow-on work by Varni et al. [41].

Below, we describe the method to compute each group member's synchronization to the group as well as the group's overall synchronization, while taking multiple types of events into consideration. The events associated with individual participants over time can be expressed by a time series. First, we will describe the event synchronization method between two time series for one event, and later we will extend this method for more than two time series and multiple events.

As described by Quian Quiroga et al. [40], suppose x_n and y_n are two time series, where $n = 1, \dots, N$, and each time series has N samples. Suppose m_x and m_y are the number of events occurring in time series x and y respectively, and E is the set of all events.

The events of both series are denoted by $e^x(i) \in E$ and $e^y(j) \in E$, where, $i = 1, \dots, m_x$, $j = 1, \dots, m_y$. The event timings on both time series are t_i^x and t_j^y ($i = 1, \dots, m_x$, $j = 1, \dots, m_y$) respectively. If the events are synchronous in both time series, then the same event will appear in both more or less simultaneously. Two events are synchronous if the same event appears within a time lag ($\pm\tau$) on both time series.

3.1 Measuring synchronization of a single type of event across two time series

Now, suppose $c^\tau(x|y)$ denotes the number of times a single type of event $e \in E$ appears in time series x shortly after (within a time lag τ) is appears in time series y. Here,

$$c^\tau(x|y) = \sum_{i}^{m_x} \sum_{j}^{m_y} J_{ij}^\tau \tag{1}$$

Where,

$$J_{ij}^\tau = \begin{cases} 1 & \text{if } 0 < t_i^x - t_j^y < \tau \\ \frac{1}{2} & \text{if } t_i^x = t_j^y \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Similarly, $c^\tau(y|x)$ denotes the number of times a single type of event $e \in E$ appears in time series y shortly after it appears (within a time lag τ) in time series x. And,

$$c^\tau(y|x) = \sum_{j}^{m_y} \sum_{i}^{m_x} J_{ji}^\tau \tag{3}$$

Where,

$$J_{ji}^\tau = \begin{cases} 1 & \text{if } 0 < t_j^y - t_i^x < \tau \\ \frac{1}{2} & \text{if } t_j^y = t_i^x \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

From $c^\tau(x|y)$ and $c^\tau(y|x)$, we can calculate the synchronization of events in two time series as,

$$Q_\tau(e) = \frac{c^\tau(x|y) + c^\tau(y|x)}{\sqrt{m_x m_y}} \tag{5}$$

$Q_\tau(e)$ represents the synchronization of events in two time series, where we are only considering a single type of event e in both time series. We normalized the value of $Q_\tau(e)$ by the number of events in both time series to get a value in between 0 and 1. Thus the value of $Q_\tau(e)$ should be $0 \leq Q_\tau(e) \leq 1$. $Q_\tau(e) = 1$ means that all the events of both time series are fully synchronized. On the other hand, $Q_\tau(e) = 0$ means that the events are not synchronized at all.

$c^\tau(x|y)$ and $c^\tau(y|x)$ values also give us the leader-follower pattern in two time series, if there exists any [40]. This relationship can be incorporated during the calculation of $Q_\tau(e)$ for situations where this pattern might be important.

3.2 Measuring synchronization of multiple types of events across two time series

$Q_\tau(e)$ gives us the synchronization of events in two time series when only one type of event is considered. In this section, we extend the notion of synchronization of events in two time series for more than one type of event.

Suppose we have n types of events $\{e_1, e_2, \ldots, e_n\} \in E(n)$, where $E(n)$ is the set of all types of events. First, we calculate $Q_\tau(e_i)$ for each event type $e_i \in E(n)$. While calculating $Q_\tau(e_i)$, we will not consider any other event types, except e_i. Now, let $m_x(e_i)$ be the number of events of type e_i occurring in time series x, and $m_y(e_i)$ is the number of events of type e_i occurring in time series y. To measure synchronization of multiple types of events between two time series, we take the average of $Q_\tau(e_i)$, weighted by the number of events of that type. We will call it the synchronization index of that pair. So, the overall synchronization of all events in time series x and time series y is:

$$\forall e_i \in E(n) : Q_\tau^{xy} = \frac{\sum [Q_\tau(e_i) \times [m_x(e_i) + m_y(e_i)]]}{\sum [m_x(e_i) + m_y(e_i)]} \tag{6}$$

3.3 Measuring the individual and overall synchronization index of the group

Using the described method, we will calculate the pair-wise synchronization index for each pair. Suppose we have H number of time series. The time series data are represented as $s_1, s_2, \ldots, s_H$. First, we calculate the pair-wise event synchronization index for each pair. So, we have the value of $Q_\tau^{s_1 s_2}, Q_\tau^{s_1 s_3}, \ldots, Q_\tau^{s_{(H-1)} s_H}$.

Building on the work of Varni et al. [41], after calculating the pair-wise synchronization index, we build an undirected weighted graph from these indices, where each time series is represented by a vertex. So, if the time series are $s_1, s_2, \ldots, s_H$, then there is a vertex in the graph which will correspond to a time series. We connect a pair of vertices with a weighted edge, based on their synchronization index value.

There exists an edge connecting two vertices in the graph if the pair-wise synchronization index of the corresponding time series is greater than or equal to a threshold value Q_{thresh}. Otherwise, there will be no edge connecting that pair of vertices in the graph. The value of Q_{thresh} should be chosen based on the group task as well as the physical configuration of the group. The weight of that edge will be this pair-wise synchronization index of that pair of vertices. We will refer to this graph as the *group topology graph (GTG)*.

The individual synchronization index depends on both the group composition as well as the size of the group. The size of the group influences the nature of the group in many ways [6]. In some group tasks, an individual may be influenced only by his/her neighbors, whereas in other tasks an individual can be influenced by any member of the group regardless of the group's configuration.

Moreover, the amount of influence may vary based on the size of the group and orientation of the setup. In the case of a very large group, the possibility for each member to be connected with all other members of the group becomes very small [6]. Other group members may have some direct or indirect influences in developing synchrony.

If the group size is small (e.g., four people), we can assume that an individual is influenced directly by all other group members. The individual synchronization index of an individual is measured as the average of the weight of the edges connected to the corresponding vertex in the topology graph. So, the individual synchronization index of series s_i is:

$$I_\tau(s_i) = \frac{\sum_{j=1,\ldots,H,\ j\neq i} Q_\tau^{s_i s_j} \times f(s_i, s_j)}{\sum_{j=1,\ldots,H,\ j\neq i} f(s_i, s_j)} \tag{7}$$

Where,

$$f(s_i, s_j) = \begin{cases} 1 & \text{iff } edge(s_i, s_j) \in GTG \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

After calculating the individual synchronization index for each member, the overall group synchronization index is calculated. We take both the individual synchronization index as well as the member's connectivity to the group into consideration while calculating the overall group synchronization index. In a small group, we also consider that each individual is supposed to connect to all other group members in the topology graph when the group is well-synchronized.

For a given vertex in the GTG, the ratio of the number of edges connecting to it, and the number of maximum possible edges in a very synchronized

Fig. 1. Game phases during one iteration of the cup game[2]. Game phases are: c - clapping, t - tapping, h - holding, m - moving, and p - passing the cup. The game is in sequence from left to right, and top to bottom.

condition for that vertex, is called *the connectivity value* (CV). Thus we can define CV of series s_i as:

$$CV(s_i) = \frac{\sum_{j=1,\ldots,H,\, j\neq i} f(s_i, s_j)}{H-1} \tag{9}$$

The CV represents how well an individual is synchronized with the rest of the group. If an individual is well-synchronized with all other members of the group, then their CV value will be 1. On the other hand, if they are not synchronized with any other group members, then their CV value will be 0.

While calculating the overall group synchronization index, both the individual synchronization index and the CV are taken into account. First, we calculate each individual's synchronization index multiplied by their CV. Then, the overall group synchronization index is computed by taking the average of this product. So, the overall group synchronization index, G_τ, is computed by:

$$G_\tau = \frac{\sum_{i=1,\ldots,H} I_\tau(s_i) \times CV(s_i)}{H} \tag{10}$$

3.4 Sampling techniques for implementation

To utilize this method in practice, a researcher will need to consider how to sample their time series data. One might select different sampling techniques for calculating individual, pair-wise, group synchronization indices. Generally, the sampling technique should be chosen depending on the nature of the group task.

For example, one sampling approach might be to divide the whole time series into a set of predefined, non-overlapping time windows (e.g., 10 *s* or 60 *s*). The synchronization indices are then measured during those windows. To represent the overall values of the synchronization indices, the average of all values measured during those time-windows can be used.

Another approach might be to use a sliding window model over the time series to measure the synchronization indices. To employ this technique, first the synchronization indices are calculated on a sliding window basis. To represent the overall value of the indices, an average over all the values calculated from the sliding windows can then be used. The size of the sliding window should be chosen based on the group task. We use this sampling technique in our work, described in the following section.

2. *Images are extracted from www.youtube.com/watch?v=Grb1oa72kmk*

4 Method Validation

We validated our method in four ways. First, we applied it to a group of humans performing a synchronous psychomotor task (playing the cup game) in order to measure group synchrony (See Section 4.1). Then, using the same data set, we compared our method to two other synchrony measurement methods from the literature (See Sections 4.2 and 4.3). Finally, Section 4.4 describes the validation of our method by employing it to measure group synchrony in synchronous and asynchronous marching tasks while being followed by an autonomous mobile robot.

4.1 Validation of our method applied to a synchronous psychomotor task

To validate our method and address our research questions, we first sought to analyze a group event where every member participates in a collective task, thus contributing to the overall group synchrony. Thus, we began by analyzing participants playing a tabletop game called "the cup game". This is a cooperative, rhythmic game, played by multiple players sitting in a circle, and consists of clapping, tapping, moving and passing cups (See Fig. 1). It was recently popularized in the Hollywood film *Pitch Perfect*.

In the cup game, each player must exhibit coordinated psychomotor skills, conduct a specific activity at a specific time, and synchronize his/her activity with the group. Thus, this game serves as a strong testbed for validating our method, and also enables us to explore how group synchronization emerges over time.

In terms of setup, each player stands or sits at a table (or on the floor), and plays the game with their hands and a cup. During every iteration of the game, each player performs a sequential and rhythmic activity with the cup, and ends the iteration by passing their cup to their neighbor. To maintain the overall rhythm of the game, all participants need to perform their tasks more or less synchronously over time. We will consider all of these activities performed by each player as the *events* of the game.

From a high level, different events during the game can be classified into five categories: clapping (c), tapping the cup (t), holding the cup (h), moving the cup (m), and passing the cup (p). Fig. 1 shows a single iteration of the game; all players perform the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAFFC.2015.2445335, IEEE Transactions on Affective Computing

following tasks in sequential order: $c-c-t-t-t-c-h-m; c-h-m-m-m-h-p-p$.

We ran a series of experiments where participants played two games in groups of four, for approximately two minutes per game. Two synchronized Kinect sensors recorded RGB and skeletal joint information of the participants during a game [45]. After the games, each participant rated on a discrete visual analog scale how well-synchronized they felt each game was, and which game was more synchronous.

In a highly synchronized game, all game events will happen more synchronously over time. On the other hand, in non-synchronous games, events may happen less synchronously. We hypothesized that each player's perception of game synchrony would be based on the relative timing of game events. (i.e., for games where events are well-timed, players would perceive the game to be more synchronous than those with poorly-timed events). We were also interested to see whether our system's automatic measurement of group synchrony would match player's perceptions.

Thus, in this work, we took players' post-game assessments to be the ground truth upon which our later validations are based. While measuring ground truth using external observers is a reasonable approach in other affective labeling work we do (c.f. [46], [47]), on this project, synchrony seemed better measured using a self-assessment approach. This was both to enable capturing immediacy of playing the cup game, as the tactile and auditory sensations which help players feel "in-sync" are challenging if not impossible to replicate for external observers. Furthermore, the literature suggests self-assessment is reliable for data collected over short time periods, as these data were [48].

4.1.1 Participants

A total of 22 people participated in our experiment, 50% female. Their average age was 24.8 years old (s.d. = 3.97), and the majority were undergraduate and graduate students. In total, there were six experimental sessions consisting of four players each. Participants were randomly categorized into six groups (two people participated twice).

Participants were trained in how to play the game before the experiment began, including playing one practice game as a group. They then played two games that were recorded. Following the experiment, participants completed a short questionnaire asking them to rate which if the two games they felt was more sychronous.

4.1.2 Data collection

Fig. 2-A shows the data collection setup. Four players stood around a table to play the game, two on each side. Two Kinect sensors were positioned approximately 86 inches above the ground and 28 inches from the table edge. The sensors tracked RGB and depth information, which afforded the ability to track the body joints of all players and the red-colored cups.

Before the game began, participants performed a brief sensor calibration process. The players stood in front of the Kinect for around 5 seconds to calibrate the sensor. After the calibration process, each Kinect tracked 15 body joint positions for two players. Fig. 2-C shows an overview of the system architecture.

The sensors were connected to two computers running Ubuntu. Both machines ran the Robot Operating System (ROS) Electric release. ROS is an open source platform which provides libraries and tools to develop robotics applications. Before data recording began, both systems were synchronized with an Ubuntu time-server to ensure they were accurately keeping time. Data were stored in the *rosbag* file format, which includes timestamps for sensor readings.

4.1.3 High level event detection

After completing the data collection, we labeled different steps of the game as the aforementioned high level events (c, m, and p). If any of the hand positions of a player were poorly tracked in a given frame, then we excluded that frame from analysis for that player.

Exclusion was based on the tracking accuracy of the Kinect sensors. Due to hand occlusion at some stages of cup game, the Kinect sensor can not track hand joint positions with full confidence. Therefore, we excluded these frames while performing event detection. On average, we excluded around 35% of frames for a session across the whole group; in the best case we excluded 12% of frames. However, this frame exclusion rarely interfered with our event detection as it was well-distributed across the sessions. Furthermore, we manually validated the accuracy of each event class before employing our event detection methods. This was performed via human labeling using a representative sample of each event class.

Cup tracking

Cups were tracked using standard blob tracking techniques in each frame. As only red color cups were used, the red blobs were tracked from the RGB image using the ROS cmvision package. After discarding very small blobs as noise, the rest of the blobs were considered as candidates for the cup's position. An undirected graph was generated using each blob center as a vertex. Two vertices were connected in the graph if they were closer than a threshold value.

From this resultant graph, we then calculated the connected components. All of the blobs in a connected component were clustered together. The center of each cluster was calculated as the mean position of the blob centers of that cluster, weighted by the area of each blob. Each cluster center was a cup center candidate, and those closer to the hands of the players were considered as the cup positions during the game.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAFFC.2015.2445335, IEEE Transactions on Affective Computing

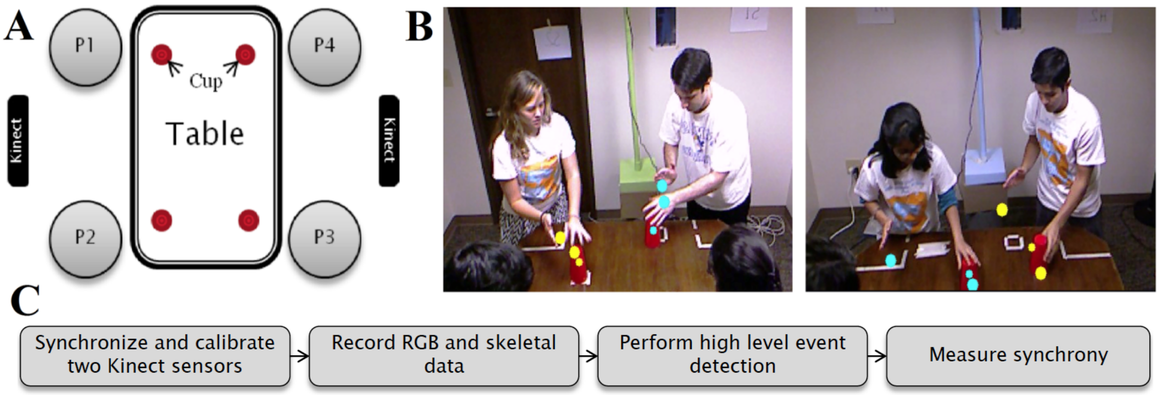

Fig. 2. A) Block diagram of the setup. P_1, P_2, P_3, and P_4 refer to Players 1, 2, 3, and 4 respectively, B) Four players playing the cup game. The players' movements are tracked by two synchronized Kinect sensors. The small solid circles represent the center of the cup, and the large solid circles represent the projected 3-D hand joint positions on the RGB image plane. C) High level system architecture.

Clap (c) event detection

A clap event was detected when the hand joints from the skeletal data were closer to one another than a threshold distance. While calculating the distance between hand joint positions, only the x and y coordinates from the 3D skeletal position were used.

A clap event happened when: 1) the hand joints' distance was within a threshold, 2) the distance between hand joints reached a local minima, and, 3) none of the hand joints were closer to the cup position in the RGB image, as calculated by projecting each 3D hand joint position on the RGB image plane. This helped distinguish clapping events from tapping events. Thus, clap events only lasted for one frame.

Move (m) and Pass (p) event detection

Cups move at several points during a game iteration, note the m or p steps in Fig. 1. If a player moved the cup with their right hand, this was denoted as a move event (m in Fig. 1), and with their left hand, a pass event (p in Fig. 1).

If a cup position was changed from the previous frame, then it meant that the cup had been moved by a player. In this case, our system assumed that the player closest to the cup's position moved the cup. To determine the hand positions of each player in the RGB image, the 3-dimensional hand joint positions from the skeletal data were projected into the RGB image plane. Then, in RGB coordinates, our system calculated the distances of the hand positions from the cup's center. If the player's right hand was closer to the moving cup than their left hand, then our system assumed that the player was moving the cup with his/her right hand, and the event was denoted as a move event (m) for that player, but if their left hand was closer, we assumed they passed the cup (p).

As move and pass events may last for several frames in the data, instead of denoting a move or pass event for every frame, we considered the starting and the ending frame of each event sequence as a new event. The frame when a move event started was denoted as a 'move start' (m_s) event, and when the move ended, that frame was denoted as a 'move end' (m_e) event for that player. Similarly, for pass event sequences, two events were generated, a 'pass start' (p_s) and a 'pass end' (p_e) event.

To demonstrate what that looks like, suppose for a few frames our event detector detected move events as follows:

$Time\ stamps:$	1.1s	1.2s	1.3s	1.4s	1.5s
$Events:$	-	m	m	m	-

This means that a move event started at timestamp 1.2 s, continued for one frame, and ended at timestamp 1.4 s. From this move event sequence, our system will generate two new events, m_s and m_e. A move start event happened when the move sequence started. When the move sequence ended, our system will label that as a move end event. Thus, our processed events would be:

$Time\ stamps:$	1.1s	1.2s	1.3s	1.4s	1.5s
$Raw\ events:$	-	m	m	m	-
$Processed\ events:$	-	m_s	-	m_e	-

4.1.4 Overall synchrony detection

After detecting the high level events for each player, each player's data were represented by a time series. Frames received from both Kinects were adjusted to represent the same frame across all time series data. This yielded four time series, each of which represented the high level events associated per player.

For these time series data, the individual and overall synchronization indices of the system were calculated using our method. For an example calculation, suppose the time series were s_1, s_2, s_3, and s_4. From these time series, pair-wise synchronization indices ($Q_\tau^{s_1 s_2}, Q_\tau^{s_1 s_3}, \ldots Q_\tau^{s_3 s_4}$) were calculated.

The pair-wise synchronization indices measure the degree to which the events are happening synchronously in both time series. In this case, these two time series are isolated from the rest of the time series of that system. Thus, this measure gives us the notion of how synchronous two players are among themselves when they are isolated from the rest of the group. For example, a higher pair-wise synchronization index may be observed if two players have the continuous tendency to synchronize their events to each other over time. We may observe a higher pair-wise synchronization index value if the participants previously played the game together, and they may synchronize with each other more easily.

After calculating these values, the topology graph was generated from the pair-wise synchronization indices. From the topology graph, each player's synchronization index $I_\tau(s_i)$ was calculated. This individual synchronization index yielded how well each player was synchronized with rest of the group.

As the group size was small and all of the players were in close proximity, all players influenced one another. If a player made a mistake at any stage of the game, not only did it affect the overall synchrony of the group, but it also affected other individuals' synchrony with the group. From the individual synchronization indices, the overall synchronization of the group was calculated using Equation 10.

4.1.5 Results

We conducted a total of ten experimental sessions. However, four sessions were excluded due to technical malfunctions (calibration or tracking errors), so here we report the results from six sessions. A session is defined as a group of four players participating in two games, where each game lasts approximately two minutes. We used a sliding window of 20 s for calculating the pair-wise, individual, and group synchronization indices, as this is approximately how long it took to complete one iteration of the game.

Table 1 shows the individual and group synchronization indices from all sessions and games. Here, the values of the synchronization indices are averaged over the duration of each game. The last column of the table represents the concordance between players' perception and our method's perception of which game was more synchronous. For example, for $Session_3$, the group synchronization indices produced by our method are 0.44 and 0.48 respectively for $Game_1$ and $Game_2$, and all players rated $Game_2$ as more synchronous; thus indicating 100% agreement.

Rather than go into depth for each game, we will now focus on one game in detail, $Session_2, Game_1$. This analysis method is identical for each game, and all games showed similar characters, so the following results reporting is generalizable to all games.

We present the individual synchronization index of four players of one game in Fig. 3-A. As we used

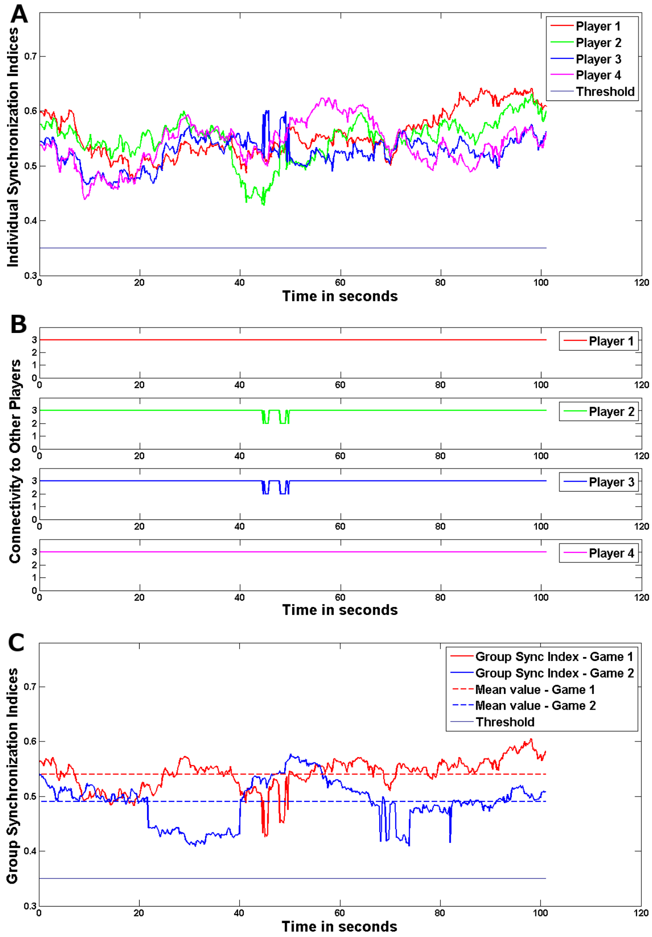

Fig. 3. A) Time vs. Individual Synchronization Indices of four players of $Session_2$, $Game_1$. B) Time vs. Connectivity to Other Players of $Session_2$, $Game_1$ (Player 1 to Player 4, from top to bottom). A Connectivity value of 3 suggests the group was well connected; 0 means no connectivity. C) Time vs. Group Synchronization Indices of both games of $Session_2$.

a sliding window of 20s, the values present at time 0s actually represent the values calculated from the time window 0s to 20s. We used $\tau = 0.21s$ for our calculation, and $Q_{thresh} = 0.35$ as the threshold of synchronization index to generate the topology graph. Based on the data from our representative sample, we found that two people were not synchronous when their pair-wise synchronization index fell below 0.35. Thus, we used this as the threshold in our experiment.

An example of the connectivity of the nodes in the topology graph is shown in Fig. 3-B. This contains four sub-graphs, each showing the connectivity of a player over time to other nodes (players) in the graph. Each sub-graph shows the connectivity of one player, with all other players in the topology graph. Time is plotted along the x-axis, and the number of connected nodes are plotted along the y-axis.

Each individual synchronization index depends on that player's connectivity with others in the topology graph, i.e., the player's synchrony with the other

TABLE 1
Individual and Group Synchronization Indices for All Sessions

Sessions	Games	Indi. Sync. Indices*				Group Sync. Index		Measurement of Precision◇ (in s.d.s)	Automated measure agreed with % of players' perception§
		P_1	P_2	P_3	P_4	GSI value†	s.d.‡		
$Session_1$	1	0.51	0.53	0.50	0.52	**0.51**	0.05	6.72	**100%**
	2	0.48	0.53	0.47	0.51	0.49	0.04		
$Session_2$	1	0.56	0.55	0.53	0.54	**0.54**	0.03	22.38	**75%**
	2	0.51	0.55	0.48	0.51	0.49	0.04		
$Session_3$	1	0.43	0.49	0.48	0.48	0.44	0.18	-4.79	**100%**
	2	0.52	0.50	0.50	0.48	**0.48**	0.05		
$Session_4$	1	0.32	0.40	0.38	0.44	0.29	0.13	-41.94	**75%**
	2	0.49	0.57	0.52	0.56	**0.54**	0.03		
$Session_5$	1	0.46	0.55	0.48	0.56	0.47	0.05	-19.19	**100%**
	2	0.45	0.56	0.51	0.57	**0.52**	0.03		
$Session_6$	1	0.49	0.53	0.45	0.54	0.47	0.07	-7.80	**75%**
	2	0.49	0.54	0.46	0.55	**0.50**	0.05		

* Mean value of individual synchronization indices of four players (P_1 to P_4).
† Mean value of group synchronization indices. For each session, the higher group synchronization index value of the game is highlighted in bold.
‡ Standard deviation (s.d.) of group synchronization indices for each game. A lower s.d. value reflects stronger, or more stable, synchrony, and a higher value reflects weaker, or less stable, synchrony.
◇ Measurement of precision for each $Session$ in standard deviations (see Supplemental File for detailed analysis).
§ Percentage of players in each session for which our automated measure produced a match with the players' perception about both games. For example, in $Session_1$ our method produced group sync. indices of 0.51 and 0.49 for Games 1 and 2 respectively. For this session, all four players agreed that Game 1 was more synchronous than Game 2.

players. From Fig. 3-B, one can see that Player 2 is connected with two more nodes in the topology graph in-between the time window of 45 s to 50 s. This means that one of the pair-wise synchronization index values of P_1-P_2, P_2-P_3, or P_2-P_4 was less than the threshold value. This is also the case for Player 3. This observation means that the pair-wise synchronization index of Player 2 and Player 3 (P_2-P_3) had fallen below the threshold during that period. Thus, there was no edge between that pair in the topology graph during that time window.

As a result, in Fig. 3-A, one can see that the individual synchronization index for Player 2 and Player 3 was decreasing before they became disconnected in the topology graph. However, after they were disconnected in the topology graph, their individual synchronization index started to increase. We can explain this situation using an example. Suppose a time window pair-wise synchronization indices of P_1-P_2, P_1-P_3, and P_1-P_4 are 0.5, 0.5, and 0.4 respectively. As the $Q_{thresh}=0.35$, P_1 is connected with all other players in the topology graph.

Given this scenario, P_1's individual synchronization index is $(0.5+0.5+0.4)/3=0.47$. Now, assume that in the next time window these values have been changed to 0.5, 0.5, and 0.3 respectively. In the changed scenario, the pair-wise synchronization index value of P_1-P_4 is less than the threshold. Thus, there will not be any edge connecting these nodes in the topology graph. For this window, the individual synchronization index of P_1 will be $(0.5+0.5)/2=0.5$. Although it is not synchronous with respect to each player in the group, for ones with which it is synchronous, the level of synchrony is relatively higher in degree. One can see this phenomenon in Fig. 3-A, just after the connectivity between P_2-P_3 was lost, the individual synchronization indices of both P_2 and P_3 increased.

For each game of a session, the group synchronization index is calculated. For example, we display group synchronization indices of this session (both games of $Session_2$) in Fig. 3-C. In this figure, time in seconds is shown along the x-axis, and group synchronization index is shown along the y-axis. We calculate group synchronization index for each time window of size 20 s. This graph also presents the mean value of group synchronization indices.

The group synchronization index depends on the individual synchronization index and the topology graph. One can see from Fig. 3-C, the group synchronization index of $Game_1$ drops between the time window of 45 s to 50 s. During the calculation of the group synchronization from the individual synchronization index, we also take each node's 'connectivity value' into consideration. This also supports the fact that the individual synchrony drops and pair-wise synchrony breaks during that period, see Fig. 3-A and B.

4.1.6 Discussion

From Table 1, one can find the percentage of the players' perception for which our measure produced a match for each session. For example, from the table one can see that the majority of participants agreed as a group that $Game_1$ was more synchronous than $Game_2$ in $Session_1$. On the other hand, for $Session_4$, the majority of participants agreed as a group that $Game_2$ was more synchronous than $Game_1$. For all of the sessions, the group synchronization indices produced by our method agreed with the perception of the majority of participants in 100% of the sessions (6 out of 6 sessions).

From the participants' perception, and from our automated method, one can see that the $Game_2$ is more synchronous than $Game_1$ for the last four sessions. One might suspect that this pattern may be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAFFC.2015.2445335, IEEE Transactions on Affective Computing

TABLE 2
Group Synchronization Indices for All Sessions Computed Using The Three Comparison Methods

Sessions	Games	Our method		By Varni et al.		CRQA
		GSI †	s.d.‡	GSI	s.d.	GSI
$Session_1$	1	**0.51**	0.05	**0.72**	0.02	**0.19**
	2	0.49	0.04	0.70	0.03	0.17
$Session_2$	1	**0.54**	0.03	**0.74**	0.03	**0.18**
	2	0.49	0.04	0.74	0.03	0.17
$Session_3$	1	0.44	0.18	**0.70**	0.03	0.17
	2	**0.48**	0.05	0.67	0.03	**0.18**
$Session_4$	1	0.29	0.13	0.61	0.04	0.15
	2	**0.54**	0.03	**0.68**	0.02	**0.20**
$Session_5$	1	0.47	0.05	0.69	0.03	0.18
	2	**0.52**	0.03	**0.72**	0.02	**0.19**
$Session_6$	1	0.47	0.07	**0.69**	0.03	**0.18**
	2	**0.50**	0.05	0.64	0.04	0.17

† Mean value of group synchronization indices. For each session, the higher group synchronization index value of the game is highlighted in bold.
‡ Standard deviation of group synchronization indices for each game. A lower s.d. value reflects stronger, or more stable, synchrony, and a higher value reflects weaker, or less stable, synchrony.

attributable to learning effects. After the first game, the players may become used to the rhythm of the game, and learned how to be synchronous as a team. Therefore, they showed a higher degree of entrainment during the second game. One also can see that most of the individual synchronization indices show higher values during the second game.

From the results, however, one can see that the first two sessions do not agree with this assumption. This may be due to the fact that most of the participants of the first two sessions had prior experience playing the game. This may explain why a learning effect was not visible during these sessions. Regardless, our automatic method still successfully measured synchrony of these two sessions, which matched with the perception of the players.

Table 1 also presents the standard deviation (s.d.) for the group synchronization indices for every game. A lower s.d. value reflects stronger or more stable synchrony, and a higher value reflects weaker or less stable synchrony. For example, the s.d. of the group synchronization index is less for $Game_1$ than $Game_2$ during $Session_2$. During the other five sessions, the s.d. values of $Game_2$ are less than the values of $Game_1$. This means that $Game_1$ exhibits more stable synchrony than $Game_2$ for $Session_2$. During the other five sessions, $Game_2$ exhibits more stable group synchrony than $Game_1$. These s.d. indices are also aligned with the answers of the participants as a group in 83.33% cases (5 out of 6 sessions).

Table 1 also presents the measurement of precision calculation for each $Session$. The measures produced by our method for two games of a session are different by at least an order of magnitude of the standard deviation. Although it appears that the two games only differ slightly in their synchronization indices, the differences are in fact significant.

4.2 Validation of our method through comparison with an alternative ES method

Our method takes multiple task level events into account to measure the synchronization of the group. One may wonder if this approach is comparable or more accurate than singular event-based methods in the literature. Thus, we first validated our approach by comparing it to the method proposed by Varni et al. [41]. As discussed in Section 3.3, this is a singular event-based synchronization detection method and represents a reasonable point of comparison.

4.2.1 Data collection

To measure group synchrony using the method by Varni et al. [41], we used the same data as described in Section 4.1.2. We considered the same six experimental sessions, consisting of two games in each session for the comparison. Here, we incorporated the skeletal data of the participants, which is aligned with the approach Varni et al. employ in their paper.

4.2.2 Method description and event detection

To employ the method by Varni et al. in [41], one first measures the pair-wise synchronization index of two participants from a single type of event. From these indices, a connectivity graph is generated and then the group synchronization index is calculated.

In their work, Varni et al. [41] measured the class of events from participants' body motion features. As described in Varni and Camurri [10], these body motion features might include the contraction index, fluidity index, etc; however, in their recent work they used motion index (MI). The authors reported calculating MI by performing silhouette-based background subtraction; however, for our comparison study we extracted MI using upper-joint skeletal data, as this was a more robust measure given the overall background illumination of our dataset.

To calculate MI, we first projected the 3D skeletal coordinates to the 2D image plane. Then, from two consecutive frames, we calculated the distance each upper body joint moved for each participant during the game. If any joint position was poorly tracked due to occlusion, we discarded that joint movement from the calculation, using a comprable exclusion method as described in Section 4.1.3. During the next available well-tracked position of that joint, we measured the distance moved from the previously well-tracked frame by that joint and divided the value by the number of consecutive poorly-tracked frames for the calculation. We took the local maxima of the sum of the distances of all the upper body joints' movements in a sliding window of n frames as our event class.

Here, we used a window size of 3 ($n = 3$). We removed local maximas with a value less than 200 pixels as noise. To calculate the group synchronization indices, we used the same time window ($20s$), $\tau = 0.21s$ and $Q_{thresh} = 0.35$ as used with our method.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAFFC.2015.2445335, IEEE Transactions on Affective Computing

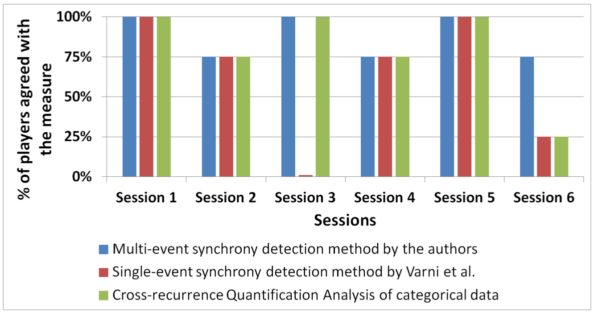

Fig. 4. Agreement between players in each session and the synchrony measures (our measure, the measure by Varni et al. [10], and the CRQA measure). One can see that the majority of the participants (at least 75%) agreed with the measurement produced by our method in 100% cases (6 out of 6 sessions).

These events represent when a participant moves their upper body the most within a time window. It can approximate any abrupt body movements, or even pulsing movements made in synchronous rhythm with the game. Thus, this class of events was similar to the class of events generated using the MI by Varni et al. [41] in their original experiment.

4.2.3 Results

Table 2 shows the group synchronization indices measured using the method proposed by Varni et al., as well as our method for all six experimental sessions. The table shows the values of the synchronization indices averaged over the duration of each game, as well as the standard deviation (s.d.) for the group synchronization indices for every game.

In Fig. 4, we present the percentage of players in each session for which the synchrony detection measures produced a match with the players' perception. For example, for $Session_3$, the group synchronization indices produced by our method are 0.44 and 0.48 respectively for $Game_1$ and $Game_2$, and all players rated $Game_2$ as more synchronous; thus indicating 100% agreement. In contrast, the group synchronization indices produced by Varni el al.'s method are 0.70 for $Game_1$ and 0.67 for $Game_2$; indicating 0% agreement with players' perceptions.

4.2.4 Discussion

From the group synchronization indices presented in Table 2, one can see that the values measured using the Varni et al. method are higher in degree than the values produced by our method for all the games. These higher values might indicate over-estimation of the synchronization indices, as only a singular type of event is considered in the Varni et al. method. Also, as our method considers multiple types of task-level events, it may be more conservative in nature.

As one can see from the data presented in Fig. 4, the group synchronization indices produced by the method proposed by Varni et al. agreed with the perceptions of the majority of participants in only 66.67% of the sessions (4 out of 6 sessions), whereas our method agrees with participants' perceptions in 100% of the sessions (6 out of 6 sessions).

Moreover, the standard deviation values in Table 2 also suggest that our method was more accurate in assessing group synchrony stability than the method proposed by Varni et al. (As a reminder, a lower s.d. value reflects more stable synchrony, and a higher s.d. value reflects weaker synchrony). If the s.d. values are equal for both games (e.g., see $Session_2$), we consider them to be aligned with the perceptions of participants as a group. The s.d. values for the method by Varni et al. are aligned with participants as a group only in 50% of cases (3 out of 6 sessions), whereas our results are aligned in 83.33% cases (5 out of 6 sessions).

4.3 Validation of our method through a comparison with CRQA

Dale et al. [42] used the cross-recurrence analysis method for quantifying the relationship between two categorical time series data through use of a contingency table. Recently, Coco et al. [43] released a package in *R* that implements CRQA and other methods. In this section, we use this package to perform a comparison between CRQA and our method.

4.3.1 Data collection

We used the same cup game data as described in Section 4.1.2. We considered the same six experimental sessions, consisting of two games per session.

However, the data contains instances where none of the synchronous events we were measuring occurred during a given moment in time. This "non-event" condition may have happened in between existing events, such as between clapping or tapping events. Our method is capable of supporting non-events, however, were we to use these data within CRQA, there is a chance these data will overestimate the recurrence profile of the time series. (i.e., both the pair-wise and group synchrony indices would be artificially inflated). To avoid this potential chance for overestimation, we performed a pre-processing step on each time series pair to remove any instances where "non-events" occurred both both players.

4.3.2 Description of the analysis

The *'crqa' R* package by [43] contains a function named *CTcrqa* which uses contingency tables (CT) to perform a cross-recurrence analysis on categorical data. First, it finds all the categories from both time series. In our data, different categories are different event types. Then it calculates all the co-occurrences of different sets of events between those two time series to build a CT. After that, the recurrence profile is computed along the diagonal of the CT [43]. Different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAFFC.2015.2445335, IEEE Transactions on Affective Computing

delays can be used to generate different recurrence profiles for the time series.

We used *CTcrqa* to compute the cross-recurrence profile for our data. Two delays were used to compute the recurrence profile. First, we used a delay of 0, which means the package computed the co-occurrences of the same event types for both time series. Then, we used a delay of 1, as we also wanted to count events as synchronous if they appeared consecutively in the time series.

For each time series pair, we first calculated the cross-recurrence of the pair for each delay. Then, we took the average of these values for the two delay patterns (0 and 1) as the measure of the pair-wise synchrony between these two time series. We used the same procedure for all of the pairs to calculate the pair-wise synchronization index.

After computing the pair-wise synchronization indices, we computed the individual and group synchronization indices for each game by following the method described in Section 3.3. First, we built group topology graph to calculate the individual synchronization indices. Then, from the connectivity value and individual synchronization indices, we computed the group synchronization index of that group. Based of the values produced by CRQA, we used Q_{thres} = 0.1 to generate the *group topology graph*. We did not use any sliding windows during these calculations.

4.3.3 Results

Table 2 also shows the group synchronization indices measured using CRQA for all six experimental sessions. In Fig. 4, we present the percentage of players in each session for which the synchrony detection measures produced a match with the players' perception.

For example, for $Session_6$, the group synchronization indices produced by our method are 0.47 and 0.50 respectively for $Game_1$ and $Game_2$, and all players rated $Game_2$ as more synchronous, thus indicating 100% agreement. In contrast, the group synchronization indices produced by CRQA are 0.18 and 0.17 respectively for $Game_1$ and $Game_2$, thus indicating 0% agreement with players' perceptions.

4.3.4 Discussion

CRQA was slightly more accurate than Varni et al.'s method in assessing group synchrony, as it reached agreement with participants in 5 of 6 sessions. However, one can see in Table 1, there was a significant difference between the two games in $Session_6$ and players' perceptions (which we take as ground truth), suggesting that $Game_2$ was more synchronous than $Game_1$. Our method concurred with players on this assessment, suggesting it is more accurate than the other two methods.

One also may observe that the synchronization indices produced by the CRQA method are fairly low. This might happen as we used the relative ordering of the events during CRQA, instead of an equally sampled time series (which is what the CRQA measure assumes). Thus, there might be cases when the same events occurred in both time series slightly apart in the event order, but not as co-occurent or consecutive events. These events might be considered as non co-occurent events, which might be the cause of the CRQA synchronization index to be lower.

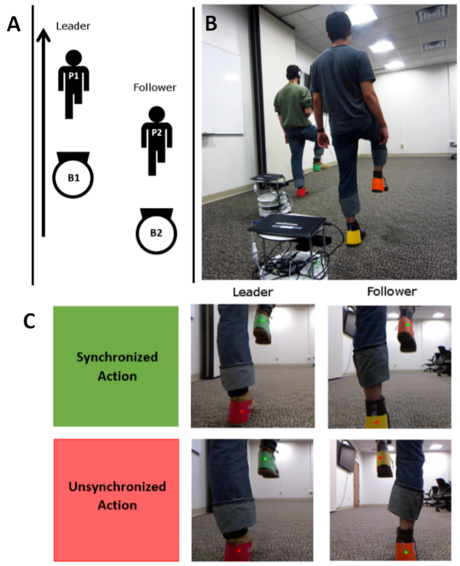

Fig. 5. A) Experimental setup of the human-robot scenario. P1 & P2 are the humans, and B1 & B2 are the robots. B) P1 & P2 demonstrate a synchronous marching pattern. C) The conditions in the experiment.

4.4 Validation of our method by applying it to a human-robot teamwork scenario

Our proposed method is capable of detecting group synchrony in other scenarios as well, and robust in identifying synchronous and asynchronous conditions. In a fourth validation experiment, we employed our proposed method on a human-robot teamwork scenario to automatically measure group synchrony. This work is part of another research project where we are exploring using our synchrony models to aid in human-robot joint action [45], [49]–[51].

In this experiment, both robots and people were in motion in a naturalistic setting [52]. This differs from the setup described in Section 4.1; there, the sensors were static; whereas here, all sensors were dynamic and in motion. We describe the experiment briefly below; full details can be found in Iqbal et al. [51].

4.4.1 Data collection

Fig. 5 shows an overview of the experiment. Two participants (performers) marched either synchronously and asynchronously, and two autonomous mobile robots followed behind each performer. The robots followed their performer autonomously, and recorded RGB and depth data from their respective on-board

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAFFC.2015.2445335, IEEE Transactions on Affective Computing

Kinect sensors. All sensors were precisely calibrated in the same manner as described in Section 4.1.2.

Performer 1 (P1) acted as the leader in the experiment, and the leader always marched at a consistent pace. To keep constant time, the leader always faced forward, and wore noise-canceling headphones that were playing "Stars and Stripes Forever", a march by John Phillip Sousa. The second performer (P2) acted as the follower, and was approximately two feet behind the leader. Both performers performed a "high-march" action together. (See Fig. 5-C for an overview of the experiment and its conditions).

We recorded a total of four scenarios during this experiment. Each scenario consisted of different patterns of synchronous and asynchronous marching actions. The follower was verbally instructed to adjust their marching pattern based on the scenario to become synchronous or asynchronous with the leader. We timed all scenarios using a stopwatch, and each lasted approximately 35 seconds. (Full details are in [51].)

4.4.2 Method description and event detection

We defined two types of task-level events to measure overall group synchrony. The first type of event was when a person begins to raise their leg from the ground. The second type of event was when a leg reaches its maximum height. As a result, a total of four types of events occur when a person is marching (one of the aforementioned events for each leg).

The events were detected offline using the recorded RGB data of the mobile robots. To track each performer's feet, we used the ROS *cmvision* package to perform standard blob-tracking. These blobs corresponded to the four unique squares of colored paper attached to the performer's left and right feet (see Fig. 5). From the task-level events for the performers,

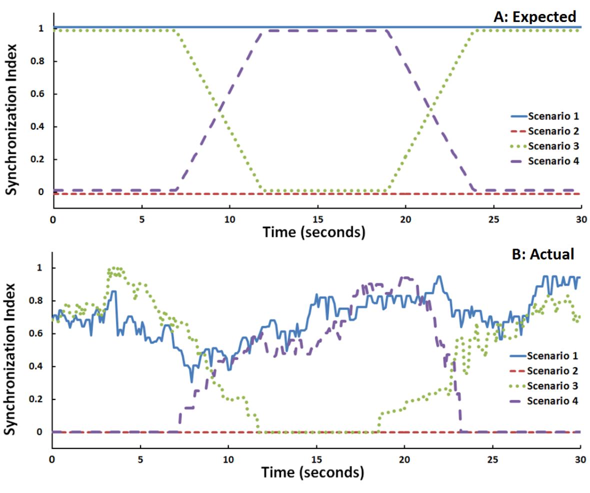

Fig. 6. A) Expected synchronization indices over time under ideal settings. B) Actual synchronization indices measured using our method applied to the four marching patterns with both robots and humans in motion.

we measured the overall group synchrony using the method described in Section 3. We used $\tau = 0.21s$ and a sliding window of $5s$ for the calculation.

4.4.3 Results

Figure 6-A shows the expected synchronization indices for these four scenarios. We expected to see a high value for a synchronization index for the entire duration of a session for Scenario 1, and a value of zero for Scenario 2. For Scenario 3, we expected to see our measured synchronization index decrease beginning around seven seconds to a value of zero at 12 seconds, and increase again at about 20 seconds. For scenario 4, we expected similar results, however in reverse order. From Fig. 6-B, one can see that the measured synchronization indices over time reasonably match those of our expected synchronization indices.

4.4.4 Discussion

The results suggest that our model is effective and robust measuring synchronized events that occur while both people and sensors are in motion. This work is encouraging for future work in understanding high-level group behavior detection and measurement in real-time for robotics. Considering motion may distort sensing, our results show that our model was capable of detecting synchronized events and measuring synchrony between people in motion, independent of the height and pacing of steps.

5 General Discussion

This paper presented a novel method to automatically detect group psychomotor entrainment by incorporating multiple types of discrete, task-level events. We described four experimental validations of the method. First, we successfully applied it to a synchronous tabletop game using fixed sensors to detect both individual and group synchronization indicies. Our method closely matched players' own perceptions of synchrony across multiple games.

Then, using the same data, we compared our method to a single-task level detection method in the literature, and showed our approach was more accurate overall. As our method incorporates multiple task level events, it is more conservative than the comparison method. In general, the synchronization indices measured using the comparison method were larger in value than ours, and were less likely to match players' perceptions of synchrony.

We also compared our method to a categorical approach from the literature, CRQA, and again demonstrated our approach yields more accurate results. Furthermore, unlike CRQA, our method is capable of dealing with "non-events", i.e., events that occur between synchronous events, and thus also provides a more conservative estimation of synchrony than CRQA. Indeed, failing to conduct a pre-processing step on the types of data we work with (psychomotor synchrony), CRQA is likely to overestimate both the pair-wise and group synchronization.

Finally, we applied our method to an experiment involving a synchronous marching task, with moving people and moving sensors (robots), and found the method robust in estimating synchrony. The results of this study suggest that our method can successfully detect both synchronous and asynchronous actions while both the robots and people were in motion.

Our method presents several advantages. First, it is simple, fast, and suitable for online implementation. In our lab, we have recently implemented this method to work online an autonomous mobile robot to measure group synchrony, and move in synchrony with others in real time [50]. We plan to release this software as open source in the near future.

Second, new types of events can easily be added without requiring any changes to the algorithm. This enables great flexibility, should a researcher wish to explore increasing the granularity of synchronous events, or want to incorporate new types of social and/or affective behaviors.

Third, the method is robust, and works successfully with data from both fixed and mobile sensors, unimodal or multimodal. This robustness is advantageous if a researcher is fusing synchronous data from sensors with varying frame rates.

In addition to group synchrony detection, our method can also be used to measure each participant's individual synchrony. Some group members may be more likely to be "team players", tending to synchronize more readily than others, and it could be useful to detect their role in how group synchrony emerges. This may particularly prove useful in the field of psychiatry, where researchers are interested in assessing how individuals with schizophrenia physically interact in groups of matched controls, and how they experience non-verbal social exclusion [53]. It also is useful to researchers in pscyhology, who often seek methods to study rapport building, social encounter smoothing, and cooperation efficiency, all of which depend on how well one can synchronize to others [25].

Our method can be helpful for other researchers in the affective computing community in several ways. It can enable the next generation of human-machine systems to estimate the affective behavior of a group as a whole, as well as individual group members, by assessing how well-entrained they are to one another. This has broad applications across the field, such as in dominance detection in groups, the affective behavior of crowds, or assessing the emergence of group roles [54]–[56]. Also, it is robust to include many different types of events from multimodal data sources; few methods exist in the field to allow for this. The field in general has been trending toward multimodal affect modeling and social scene understanding (c.f. [56], [57]; this work provides both a theoretical and practical contribution in this area.

Finally, our method could be used to detect engagement and skill acquisition in affective learning contexts. Many researchers in this community seek to ensure ecological validity in their work [58]; for most students the naturalistic learning context involves relationships with peers. The ability to model the behavior of a student within the context of group learning could be useful in this space.

This work also has implications in fields such as human-robot interaction, human-computer interaction, and ubiquitous computing. As intelligent systems begin to proliferate in human social environments, their ability to understand, predict, and respond to human group social behavior becomes increasingly important [52]. Joint action has recently become an important and popular topic across these communities, as well in the affective science and neuroscience communities [59]. Our method will directly support these research efforts.

References

[1] N. Campbell, "Multimodal Processing of Discourse Information; The Effect of Synchrony," *Int. Symp. Univers. Commun.*, 2008.

[2] D. B. Jayagopi, H. Hung, C. Yeo, and et al., "Modeling Dominance in Group Conversations Using Nonverbal Activity Cues," *IEEE T Audio Speech*, 2009.

[3] O. Brdiczka, J. Maisonnasse, and P. Reignier, "Automatic detection of interaction groups," in *Proc. Int. Conf. Multimodal Interfaces*, 2005.

[4] J. Bluemink, R. Hämäläinen, T. Manninen, and et al., "Group-level analysis on multiplayer game collaboration: how do the individuals shape the group interaction?" *Interact Learn Envir*, 2010.

[5] D. Gatica-Perez, "Automatic nonverbal analysis of social interaction in small groups: A review," *Image Vision Comput*, 2009.

[6] D. R. Forsyth, *Group dynamics*, 4th ed. T. Wadsworth, 2009.

[7] J. Phillips-Silver, C. Aktipis, and G. Bryant, "The ecology of entrainment: Foundations of coordinated rhythmic movement," *Music Percept*, 2010.

[8] M. J. Richardson, R. L. Garcia, T. D. Frank, and et al., "Measuring group synchrony: a cluster-phase method for analyzing multivariate movement time-series," *Frontiers in Physiology*, 2012.

[9] M. Clayton, R. Sager, and U. Will, "In time with the music: the concept of entrainment and its significance for ethnomusicology." *European Meetings in Ethnomusicology*, 2005.

[10] G. Varni, G. Volpe, and A. Camurri, "A System for Real-Time Multimodal Analysis of Nonverbal Affective Social Interaction in User-Centric Media," *IEEE T Multimedia*, 2010.

[11] S. G. Barsade, "The ripple effect: Emotional contagion and its influence on group behavior," *ASQ*, 2002.

[12] L. Noy, E. Dekel, and U. Alon, "The mirror game as a paradigm for studying the dynamics of two people improvising motion together," *P Natl Acad Sci USA*, 2011.

[13] R. C. Schmidt and M. J. Richardson, "Dynamics of interpersonal coordination," *Und Com Sys*, 2008.

[14] E. Pereda, R. Q. Quiroga, and J. Bhattacharya, "Nonlinear multivariate analysis of neurophysiological signals." *Prog Neurobiol*, 2005.

[15] S. Kirschner and M. Tomasello, "Joint drumming: social context facilitates synchronization in preschool children." *J Exp Child Psychol*, 2009.

[16] S. Khalfa, M. Roy, P. Rainville, and et al., "Role of tempo entrainment in psychophysiological differentiation of happy and sad music?" *J Psychophysiol*, 2008.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAFFC.2015.2445335, IEEE Transactions on Affective Computing

[17] T. Kreuz, F. Mormann, R. G. Andrzejak, and et al., "Measuring synchronization in coupled model systems: A comparison of different approaches," *Physica D: Nonlinear Phenomena*, 2007.
[18] T. Kreuz, J. S. Haas, A. Morelli, and et al., "Measuring spike train synchrony," *J Neurosci Meth*, 2007.
[19] T. Himberg and M. R. Thompson, "Learning and synchronising dance movements in south african songs: Cross-cultural motion-capture study," *Dance Res*, 2011.
[20] M. M. Louwerse, R. Dale, E. G. Bard, and et al., "Behavior matching in multimodal communication is synchronized," *Cognitive Sci*, 2012.
[21] D. Lakens, "Movement synchrony and perceived entitativity," *J Exp Soc Psychol*, 2010.
[22] D. Lakens and M. Stel, "If They Move in Sync, They Must Feel in Sync: Movement Synchrony Leads to Attributions of Rapport and Entitativity," *Soc Cognition*, 2011.
[23] F. Cummins, "Rhythm as entrainment: The case of synchronous speech," *J Phonetics*, 2009.
[24] C. D. Looze, S. Scherer, B. Vaughan, and et al., "Investigating automatic measurements of prosodic accommodation and its dynamics in social interaction," *Speech Commun*, 2014.
[25] E. Delaherche, M. Chetouani, A. Mahdhaoui, and et al., "Interpersonal Synchrony: A Survey of Evaluation Methods across Disciplines," *IEEE T on Affective Computing*, 2012.
[26] A. Vinciarelli, M. Pantic, and H. Bourlard, "Social signal processing: Survey of an emerging domain," *Image and Vision Computing*, 2009.
[27] G. Hoffman and G. Weinberg, "Gesture-based human-robot jazz improvisation," in *Proc. of IEEE ICRA*, 2010.
[28] R. Dale and M. J. Spivey, "Unraveling the dyad: Using recurrence analysis to explore patterns of syntactic coordination between children and caregivers in conversation," *Lang Learn*, 2006.
[29] M. C. Romano, M. Thiel, J. Kurths, and et al., "Detection of synchronization for non-phase-coherent and non-stationary data," *Europhys Lett*, 2005.
[30] N. Marwan, M. Carmenromano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," *Physics Reports*, 2007.
[31] J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle, "Recurrence plots of dynamical systems," *Europhys Lett*, 1987.
[32] N. Marwan and J. Kurths, "Nonlinear analysis of bivariate data with cross recurrence plots," *Phys Lett A*, 2002.
[33] C. L. Webber Jr and J. P. Zbilut, "Recurrence quantification analysis of nonlinear dynamical systems," *Tutorials in contemporary nonlinear methods for the behavioral sciences*, 2005.
[34] J. P. Zbilut and C. L. Webber, "Embeddings and delays as derived from quantification of recurrence plots," *Phys Lett A*, 1992.
[35] I. Konvalinka, D. Xygalatas, J. Bulbulia, and et al., "Synchronized arousal between performers and related spectators in a fire-walking ritual," *P Natl Acad Sci USA*, 2011.
[36] I. Konvalinka, P. Vuust, A. Roepstorff, and et al., "Follow you, follow me: continuous mutual prediction and adaptation in joint tapping." *J of Experimental Psychology*, 2010.
[37] S. M. Boker and J. L. Rotondo, "Symmetry building and symmetry breaking in synchronized movement," *Adv Consc Res*, 2002.
[38] R. Q. Quiroga, A. Kraskov, T. Kreuz, and P. Grassberger, "Performance of different synchronization measures in real data: a case study on electroencephalographic signals," *Phys Rev E*, 2002.
[39] Z. Néda, E. Ravasz, Y. Brechet, and et al., "Self-organizing processes: The sound of many hands clapping," *Nature*, 2000.
[40] R. Q. Quiroga, T. Kreuz, and P. Grassberger, "Event synchronization: a simple and fast method to measure synchronicity and time delay patterns," *Phys Rev E*, 2002.
[41] G. Varni, G. Volpe, and B. Mazzarino, "Towards a social retrieval of music content," in *Proc. IEEE Conf. on Social Computing, Privacy, Security, Risk and Trust*, 2011.
[42] R. Dale, A. S. Warlaumont, and D. C. Richardson, "Nominal cross recurrence as a generalized lag sequential analysis for behavioral streams," *Int J Bifurcat Chaos*, 2011.
[43] M. I. Coco and R. Dale, "Cross-recurrence quantification analysis of categorical and continuous time series: an r package," *Frontiers in psychology*, 2014.
[44] G. Kress, *Multimodality: A social semiotic approach to contemporary communication*. Routledge, 2009.
[45] T. Iqbal, M. J. Gonzales, and L. D. Riek, "A model for time-synchronized sensing and motion to support human-robot fluency," in *Proc. of ACM/IEEE HRI, Timing in HRI*, 2014.
[46] C. J. Hayes, C. R. Crowell, and L. D. Riek, "Automatic processing of irrelevant co-speech gestures with human but not robot actors," in *Proc. of ACM/IEEE HRI*, 2013.
[47] L. D. Riek, M. F. Oconnor, and P. Robinson, "Guess what? a game for affective annotation of video using crowd sourcing," in *Proc. of ACII*, 2011.
[48] J. A. Ross, "The reliability, validity, and utility of self-assessment," *Practical Assessment Research & Evaluation*, 2006.
[49] T. Iqbal and L. D. Riek, "Role distribution in synchronous human-robot joint action," in *Proc. of IEEE RO-MAN, Towards a Framework for Joint Action*, 2014.
[50] S. Rack, T. Iqbal, and L. Riek, "Enabling synchronous joint action in human-robot teams," in *Proc. of ACM/IEEE HRI*, 2015.
[51] T. Iqbal, M. J. Gonzales, and L. D. Riek, "Joint action perception to enable fluent human-robot teamwork," in *Proc. of IEEE RO-MAN*, 2015.
[52] L. D. Riek, "The Social Co-Robotics Problem Space: Six Key Challenges," *In Proc. of RSS*, 2013.
[53] M. Lavelle, P. G. Healey, and R. McCabe, "Participation during first social encounters in schizophrenia," *PloS one*, 2014.
[54] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. of IEEE CVPR*, 2009.
[55] B. Raducanu, J. Vitria, and D. Gatica-Perez, "You are fired! nonverbal role analysis in competitive meetings," in *Proc. of IEEE ICASSP*, 2009.
[56] A. Vinciarelli, M. Pantic, D. Heylen, and et al., "Bridging the gap between social animal and unsocial machine: A survey of social signal processing," *IEEE T on Affective Computing*, 2012.
[57] G. Castellano, I. Leite, A. Pereira, and et al., "Multimodal affect modeling and recognition for empathic robot companions," *International Journal of Humanoid Robotics*, 2013.
[58] S. Afzal and P. Robinson, "Natural affect data: Collection and annotation," in *New Perspectives on Affect and Learning Technologies*, 2011.
[59] N. Sebanz, H. Bekkering, and G. Knoblich, "Joint action: bodies and minds moving together," *Trends Cogn Sci*, 2006.

Tariq Iqbal received his BS in Computer Science and Engineering from Bangladesh University of Engineering and Technology, and an MS in Computer Science from the University of Texas at El Paso. He is currently working toward his PhD in the department of Computer Science and Engineering at the University of Notre Dame. His research interests include computer vision, social signal processing, and human-robot interaction. He is a student member of the IEEE.

Laurel D. Riek (M'00-SM'14) received the BS degree in Logic and Computation from Carnegie Mellon University and the PhD degree in Computer Science from the University of Cambridge. She is the Clare Boothe Luce Assistant Professor of Computer Science and Engineering at the University of Notre Dame, and directs of the Robotics, Health, and Communication Laboratory. Her current research interests include robotics, social signal processing, human behavior understanding, and healthcare informatics. She is the author of more than 75 technical articles in these areas. She serves on the editorial board of IEEE Transactions on Human Machine Systems, the Steering Committee of the ACM/IEEE Conference on Human-Robot Interaction, and numerous conference program committees. She is a senior member of the IEEE.