Copyright ⓒ 2012 KSII

Data Alignment for Data Fusion in Wireless Multimedia Sensor Networks Based on M2M

Jose Roberto Perez Cruz, Saul E. Pomares Hernandez* and Enrique Munoz de Cote
Computer Science Department, National Institute of Astrophysics,
Optics and Electronics (INAOE), Puebla México
[e-mail: {jrpc, spomares, jemc@inaoep.mx]
*Corresponding author: Saul E. Pomares Hernandez

Received August 31, 2011; Revised November 5, 2011; Accepted November 28, 2011; Published January 31, 2012

Abstract

Advances in MEMS and CMOS technologies have motivated the development of low cost/power sensors and wireless multimedia sensor networks (WMSN). The WMSNs were created to ubiquitously harvest multimedia content. Such networks have allowed researchers and engineers to glimpse at new Machine-to-Machine (M2M) Systems, such as remote monitoring of biosignals for telemedicine networks. These systems require the acquisition of a large number of data streams that are simultaneously generated by multiple distributed devices. This paradigm of data generation and transmission is known as *event-streaming*. In order to be useful to the application, the collected data requires a preprocessing called *data fusion*, which entails the temporal alignment task of multimedia data. A practical way to perform this task is in a centralized manner, assuming that the network nodes only function as collector entities. However, by following this scheme, a considerable amount of redundant information is transmitted to the central entity. To decrease such redundancy, data fusion must be performed in a collaborative way. In this paper, we propose a collaborative data alignment approach for event-streaming. Our approach identifies temporal relationships by translating temporal dependencies based on a timeline to causal dependencies of the media involved.

Keywords: Data alignment, data fusion, in-network processing, wireless sensor networks

DOI: 10.3837/tiis.2012.01.013

1. Introduction

Recent advances in MEMS and CMOS technologies have motivated the development of low-cost and low-power devices (sensors) and wireless multimedia sensor networks (WMSN). The WMSNs were created to interconnect and to communicate devices that are able to ubiquitously harvest audio and video streams, in addition to still images and scalar data from physical environments. The sensor nodes in a WMSN have limited communication ranges, as well as constraints in energy and resources. A WMSN is planned to require minimal human intervention.

WMSNs have drawn the attention of research and industrial communities since this kind of networks has allowed them to glimpse at new M2M Systems, such as remote monitoring of biosignals for telemedicine networks, multimedia surveillance systems, and environmental monitoring systems [1], among others.

Such systems require the acquisition of a large number of data streams that are simultaneously generated and transmitted by multiple distributed devices. This paradigm of data generation is known as event-streaming, which represents the generation and the transmission of data as a continuous stream of events reported by multiple sources [2][3][4][5]. For these reasons, in order to be useful to the application, all the collected data needs a preprocessing called *data fusion*[1]. For example, suppose that there is a network of fixed wireless cameras along a way, with the aim of capturing on video the path followed by a car (see **Fig. 1**). As each camera has a limited vision field, the resultant single video of the path must be formed from the fusion of the multiple possible redundant views collected by different cameras. The redundant views result from the vision fields of two or more cameras that overlap (see **Fig. 1**).

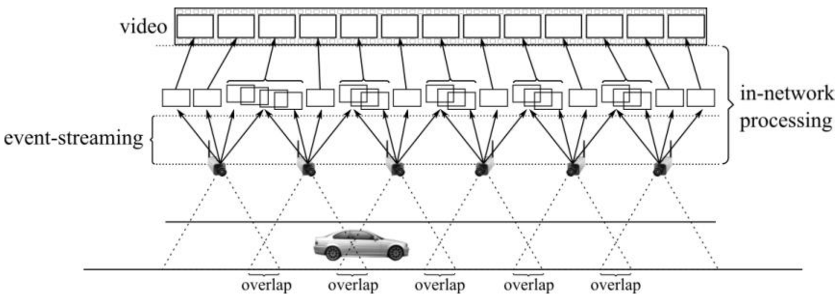


Fig. 1. In-network processing scenario.

Two main tasks in the data fusion process are the *temporal alignment* [6] and the *spatial correlation* [7][8] of the collected data. Temporal alignment consists in performing data temporal adjustments by identifying common temporal references (e.g. global timeline), whereas spatial correlation consists in establishing data spatial relationships mainly based on a common overlapped sensing area.

In this paper we address only the temporal alignment problem. A practical way to perform this task is in a centralized manner, assuming that the sensor nodes only act as collector entities

[1] Data fusion refers to the alignment, association, correlation, filtration and aggregation of the collected data [10], [11].

[6]. However, by following this scheme, a considerable amount of redundant information is transmitted to the central entity (sink).

According to Akyildiz et al. [1] and Gavalas et al. [9], data fusion must be performed in a collaborative way to reduce the redundancy in the transmitted data, thereby enhancing the use of network resources. This kind of collaborative processing is called *in-network* processing. For the data alignment problem, the *in-network* processing implies that the cameras collaborate directly in order to detect common temporal dependencies among the harvested data.

Unfortunately, the characteristics and restrictions of a WMSN make it difficult to establish a common temporal reference, such as a global timeline, mainly due to: 1) the resource constraints, 2) the channel variability, 3) the dynamicity in the topology, 4) the lack of perfectly synchronized physical clocks, 5) the absence of shared memory, and 6) the asynchronous nature of the event-streaming [4].

In the absence of a global timeline, the techniques based on the happened-before relation proposed by Lamport [12] result very useful to detect and to ensure temporal relationships among multimedia data [4][13][14]. In these works, the temporal relationships established according to the timeline are replaced by causal precedence relationships. Such solutions cannot be applied to event-streaming data alignment since they are all designed to perform temporal alignment among *local-streams*, where each local-stream is composed of events that originate from the same source.

In this paper, we propose a collaborative data alignment approach for event-streaming suitable for data fusion in WMSN. Our approach identifies temporal relationships by translating temporal dependencies based on a timeline to causal dependencies of the media involved by avoiding the use of global references.

The rest of the paper is organized as follows. Section 2 presents the system model and background. Section 3 introduces the In-network data alignment approach. Finally, Section 4 concludes with a few remarks.

2. System Model and Background

2.1 System Model

We specify a WMSN as a distributed system, which consists of three main base components: the processes, the messages, and the events.

• **Processes.** Each sensor node associated to the WMSN is represented as an individual process. Hence, a WMSN has an associated set of processes P that communicate with each other by message passing. A process can only send one message at a time.

• **Messages.** We consider a finite set of messages M sent by a process $p \in P$.

• **Events.** An event represents an instant execution performed by a process. In a WMSN, a process can only execute two kinds of events: internal events and external events. An internal event is an action that changes the state of a process and cannot be seen by other processes. An external event is also an action in a process, but it is seen by other processes, thereby affecting the state of the global system. For communication interactions, there are three types of external events: *send*, *receive* and *delivery*. For our problem of data alignment, we only consider the *send* and *delivery* external events. The *send* event refers to the emission of a message executed by a process. The *delivery* event refers to the execution performed by a process to present the received information to an application of another process.

2.2 Happened-before Relation

A suitable way to establish precedence dependencies among events in an asynchronous distributed system is the happened-before relation (HBR) defined by Leslie Lamport [8]. This relation establishes the rules to determine whether an event is the cause or the effect of another event, without the use of global references.

Definition 1: The causal relation "$\rightarrow$" is the smallest relation on a set of events *E* satisfying the following conditions:

1. If *a* and *b* are events belonging to the same process, and *a* was originated before *b*, then $a \rightarrow b$.
2. If *a* is the sending of a message by one process, and *b* is the receipt of the same message in another process, then $a \rightarrow b$.
3. If $a \rightarrow b$ and $b \rightarrow c$, then $a \rightarrow c$.

Based on Definition 1, Lamport defined that two events are concurrent if $\neg(a \rightarrow b)$ and $\neg(b \rightarrow a)$, which is denoted by "$a \parallel b$".

For example, assume that the cameras in the scenario presented in **Fig. 1** are in *standby* and that the vehicle begins to move from left to right. In this context, the cameras *wakeup* when they detect a motion activity, and immediately after, each camera *sends* a broadcast message to its neighbors. According to condition 1 of the HBR, the internal event *wakeup* at each camera causally precedes the event *sends* (*wakeup* → *sends*). On the other hand, if a chain of events exists between *wakeups* originated from different sources, such that *wakeup* → *sends* → *wakeup,* then according to conditions 2 and 3, these events are also causally related.

2.3 Happened-before Relation for Intervals

An interval is a set of events that originate from the same source and occur during a period of time. In this paper we also refer to an interval as a *local-stream*. Lamport establishes in [15] the following: let *A* and *B* be two intervals, where interval *A* happens before interval *B* if all the elements that compose interval *A* causally precede all the elements of interval *B*. We formally define the HBR for intervals as follows:

Definition 2: The causal relation "$\rightarrow$" is established at a set level by satisfying the following conditions:

1. $A \rightarrow B$ if $a \rightarrow b$ for all pair a,b $a \rightarrow b, \forall(a,b) \in A \times B,$ and
2. $A \rightarrow B$ if $\exists C \mid (A \rightarrow C \wedge C \rightarrow B)$.

Based on Definition 2, it is said that *A co-occurs* with *B* if either some portion of *A* or the entire interval *A* happens at the "same time" as interval *B.*

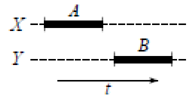
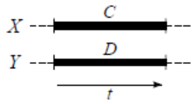
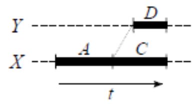
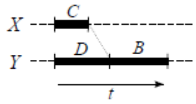
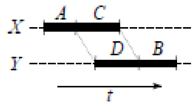
2.4 The Logical Mapping Model

The logical mapping model introduced in [13] is useful to represent "cause and effect" pairwise interactions between processes. Such model expresses temporal relations between local-streams in terms of the happened-before relation for intervals. The logical mapping translation can decompose every pair (*X*, *Y*) of intervals of a temporal relation into four subintervals: A, C, D and B, as shown in **Table 1**.

The logical mapping model identifies five logical mappings which are: *precedes, simultaneous, ends, starts* and *overlaps*. These five logical mappings are sufficient to represent all possible temporal relations between continuous media (interval-interval relations

[16]), discrete media (point-to-point relations [17]), and discrete-continuous media relations [17].

Table 1. Logical mapping model

Logical mappings		Scenario example
precedes:	$A \to B$	
simultaneous:	$C \vert\vert\vert D$	
ends:	$A \to (C \vert\vert\vert D)$	
starts:	$(C \vert\vert\vert D) \to B$	
overlaps:	$A \to (C \vert\vert\vert D) \to B$	

3. In-network Data Alignment Approach

One main task of the data fusion process is the data alignment, which consists in detecting temporal references and adjusting the collected data according to the detected references. With such adjustments, the data can be fused at a later stage. To achieve this, it is necessary to detect patterns, such as concurrency, among the generated streams. As has been shown in the works of Chandra and Kshemkalyani [3], [18], [19], a practical way to detect patterns among local-streams is by assuming a global time axis and by using the interval-interval relations defined by Allen [16]. Unfortunately, establishing a global timeline in a WMSN is difficult due to the lack of perfectly synchronized clocks [4]. However, as has been shown by Pomares et al. with their logical mapping model [13], the interval-interval relations can be expressed in terms of the happened-before relation, allowing the system to prescind from physical clocks.

As far as data alignment is concerned, we are specifically interested in detecting the concurrences among sets of events since the events within such sets need to be fused or filtered.

The logical mapping model [13] is focused on detecting temporal dependencies between a pair of intervals. For our problem, this refers only to the base case, which is the detection of temporal dependencies between local-streams. In the next sections, we present the data aligment proposal focused on detecting temporal dependencies between *event-streamings* that are composed by events originated from multiple sources.

3.1 Data Alignment Proposal

We begin by defining the *event-streaming* for our purpose as a finite collection *ES* of disjoint subsets $Q_i^{R_i}$ of events, originated from multiple sources, and causally arranged one after another without interruption. This collection has the general causal structure:

$$ES = Q_1^{R_1} \to Q_2^{R_2} \to \cdots \to Q_{n-1}^{R_{n-1}} \to Q_n^{R_n}$$

where each superindex R_i refers to the set of identifiers of the processes that originated the events.

This arrangement of subsets $Q_i^{R_i}$ allows us to establish a relative timeline where each subset $Q_i^{R_i}$ represents a unique time-slot. The fact that the subsets $Q_i^{R_i}$ are disjoint implies that each event in an event-streaming belongs to a unique subset $Q_i^{R_i}$, and therefore, it is located at a specific time-slot.

In terms of the data fusion problem, the concurrent events that belong to a same $Q_i^{R_i}$ represent data originated in the same time-slot by different sources that can be redundant. For this reason, such data needs to be fused or filtered in a later processing.

The identification and construction of the subsets $Q_i^{R_i}$ that compose an event-streaming is collaboratively performed according to the causal view of the involved processes as is explained in the following section.

3.2 Data Alignment Process Description

The data alignment process is described through four stages. The first stage, called A.1 describes how two local-streams are initially aligned to compose a first event-streaming. The next three stages, called B.1, B.2, and B.3, describe how an event-streaming and a local-stream are aligned to establish a relative timeline. To illustrate in a broad manner how these stages work, we present the following example.

In the scenario of **Fig. 1**, assume that the cameras *wakeup* from left to right when they detect the vehicle motion activity, and that immediately after, each camera begins to broadcast video frames (messages) to its neighbors. In this context, the initial stage A.1 is responsible for aligning the video sequences (local-streams) of the first two cameras by generating a first event-streaming. Once the first stage has finished, we continue through stages B.1, B.2, and B.3, to align the resultant event-streaming with the subsequent local-stream of the next camera to the right as the vehicle moves. A new event-streaming results from B.x stages which at the time will be aligned with the next local-stream and so on. The B.x stages will be repeated until no more local-streams need to be aligned.
A detail description of the alignment process is presented as follows.

A.1 Initial stage: alignment of two local-streams.

Initially, we have two local-streams, X_c and Y_d. X_c is a local-stream originated by a process p_c, while Y_d is a local-stream originated by a process p_d; and X_c precedes Y_d or X_c co-occurs with Y_d. Applying the native logical mapping, we generate the first event-streaming *ES* as follows.

Taking as reference the scenario presented in **Fig. 2**, we construct a first subset $Q_1^{\{c\}}$ with the first non-concurrent events in X_c. To determine those non-concurrent events, we need to identify all the events $x \in X$ that precede the beginning of Y (see **Fig. 2**).

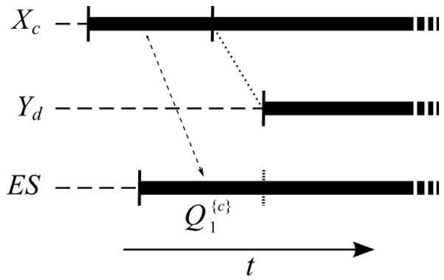


Fig. 2. Aligning the first subset of the first event-streaming.

We then proceed to construct a second subset $Q_2^{\{c,d\}}$ with the concurrent events between X_c and Y_d. The concurrent segments of both local-streams will be bounded by the beginning of Y_d and the end of either of the two local-streams (see **Fig. 3**).

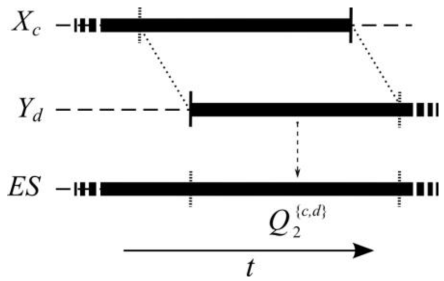
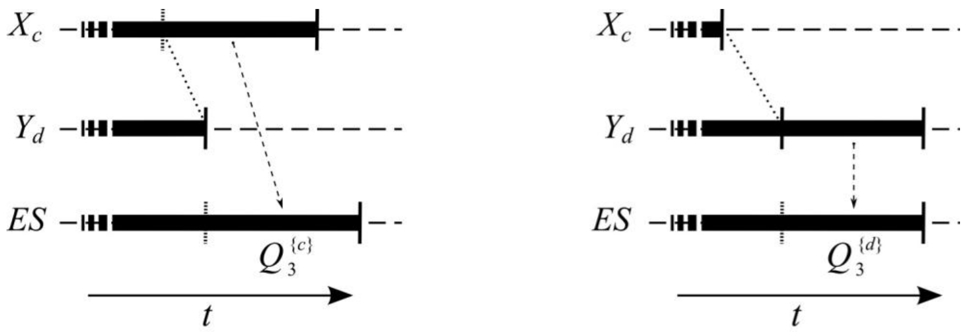


Fig. 3. Aligning the second subset of the first event-streaming.

The last subset $Q_3^{R_3}$ is constructed depending on which local-stream finishes first. If X_c finishes first, the last subset will contain the remaining events of Y_d. Otherwise, the last subset will contain the remaining events of X_c. These two cases are illustrated in **Fig. 4**.



(a) **(b)**

Fig. 4. Aligning the last subset of the first event-streaming:

(a) X finishes first, **(b)** Y finishes first.

Therefore, the first event-streaming has the general causal structure:

$$ES = Q_1^{\{c\}} \to Q_2^{\{c,d\}} \to Q_3^{R_3}$$

where $R_3 = \{c\}$ or $R_3 = \{d\}$ depending on which local-stream finishes first.

From now on, the event-streaming is labeled as X_β, where β is the set of identifiers of the processes that generated the events, and the local-stream is labeled as Y_k, where k is the identifier of the local process.

B.1 Aligning the first subsets of events without concurrences between an event-streaming and a local-stream.

In the first step, we form the first subsets of the new event-streaming with the subsets $Q_a^{R_a} \in X_\beta$ that precede the local-stream Y_k and have non-concurrent events. These subsets are relabeled and directly integrated to the new event-streaming to form the first subsets of events $Q_a^{T_a} \in ES$. Taking the example presented in the scenario of **Fig. 5**, the subsets $Q_1^{R_1}$ and $Q_2^{R_2}$ are the events that are directly integrated to the new event-streaming as the first subsets $Q_a^{T_a} \in ES$.

If a subset $Q_a^{R_a} \in X_\beta$ has events that are concurrent with a part of the local-stream Y_k, this subset is segmented to form two new subsets for the new event-streaming *ES*. The first of the two new subsets will contain the part of $Q_a^{R_a}$ whose events have no concurrence. For the scenario depicted in **Fig. 5**, the new subset $Q_a^{T_a}$, created with the non-concurrent events of $Q_a^{R_a}$, corresponds to the subset $Q_3^{T_3}$.

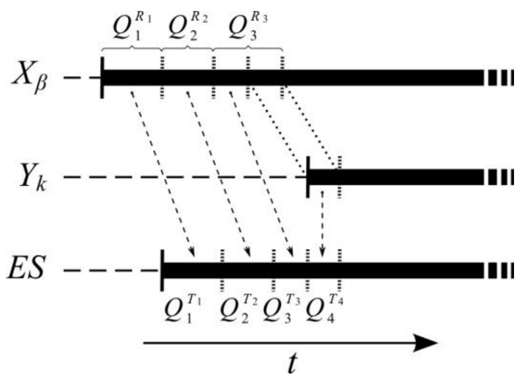


Fig. 5. Aligning the first subsets of events of an event-streaming.

B.2 Aligning the subsets of events with concurrences between an event-streaming and a local-stream. If during stage B.1, a subset $Q_a^{R_a}$ whose events are concurrent with a part of the local-stream Y_k was detected, the part of $Q_a^{R_a}$ with concurrent events forms a new subset

$Q_b^{T_b}$ along with the concurrent events of Y_k. For the example depicted in **Fig. 5**, the new subset $Q_b^{T_b}$, created with the concurrent events of $Q_a^{R_a}$, corresponds to the subset $Q_4^{R_4}$.

Once the beginning of the concurrent parts of both streams is detected, all the subsequent subsets $Q_b^{R_b} \in X_\beta$ are aligned with respect to the events of Y_k until one of the two streams finishes. This means that for each subset $Q_b^{R_b}$ in the concurrent part of X_β, a new subset $Q_c^{T_c}$ will be constructed for the new event-streaming. Taking as an example the scenario of **Fig. 6**, the new subsets $Q_c^{T_c}$ correspond to the subsets $Q_5^{T_5}, Q_6^{T_6}$ and $Q_7^{T_7}$.

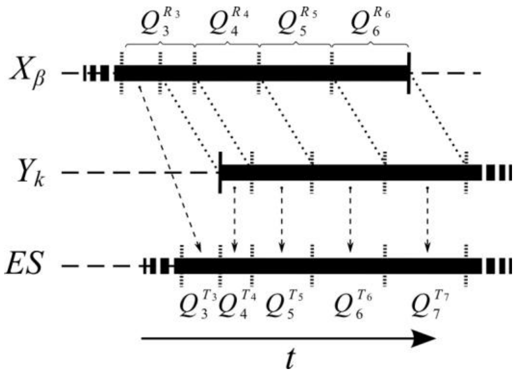


Fig. 6: Aligning subsets of events with concurrences.

The final subsets of the resultant event-streaming will be constructed depending on which stream finishes first. If local-stream Y_k finishes first, the last concurrent subset $Q_b^{R_b} \in ES$ can contain some events that are concurrent with Y_k and other events that have no concurrence (see **Fig. 7**). If this is the case, such subset $Q_b^{R_b}$ needs to be segmented to construct two new subsets for the new event-streaming. The first of the two new subsets will contain the concurrent part of $Q_b^{R_b}$ and the concurrent events of Y_k. For the scenario depicted in **Fig. 7**, the new subset $Q_c^{T_c}$, created with the concurrent events of the last subset $Q_b^{R_b} \in X_\beta$ and Y_k, corresponds to the subset $Q_7^{T_7}$.

B.3 Aligning the last subsets of events without concurrences. This stage is explained through two cases.

Case A. Y_k finishes before X_β. If at the end of stage B.2, the last subset $Q_b^{R_b}$ was segmented, the second created subset for the event-streaming, denoted as $Q_d^{T_d}$, will contain the remaining non-concurrent events of $Q_b^{R_b}$. In the scenario of **Fig. 7**, such subset $Q_d^{T_d}$ corresponds to the subset $Q_8^{T_8}$.

The fact that the local-stream Y_k finishes first implies that the concurrent parts of both streams finish along with Y_k. Therefore, the remaining subsets $Q_c^{R_c} \in X_\beta$ will become $Q_d^{T_d} \in ES$, which are the last subsets. In the scenario of **Fig. 7**, the last subsets $Q_d^{T_d}$ are the subsets $Q_9^{T_9}, Q_{10}^{T_{10}}$ and $Q_{11}^{T_{11}}$.

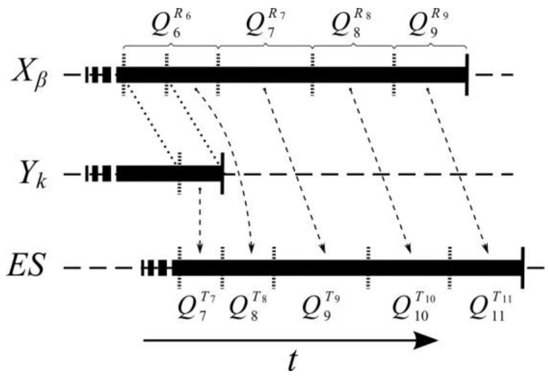


Fig. 7: Aligning the last subsets of events without concurrences

when Y_k finishes before X_β.

Case B. X_β finishes before Y_k. The fact that the event-streaming X_β finishes first means that the concurrent parts of both streams finish along with the event-streaming X_β. After the last subset $Q_c^{T_c} \in ES$ was constructed with the concurrent events of X_β and Y_k, only one more subset $Q_d^{T_d}$ is constructed. Such subset $Q_d^{T_d}$ will contain the remaining events of the local-stream Y_k. In the scenario of **Fig. 8**, the last subset $Q_d^{T_d}$ corresponds to the subset $Q_8^{T_8}$.

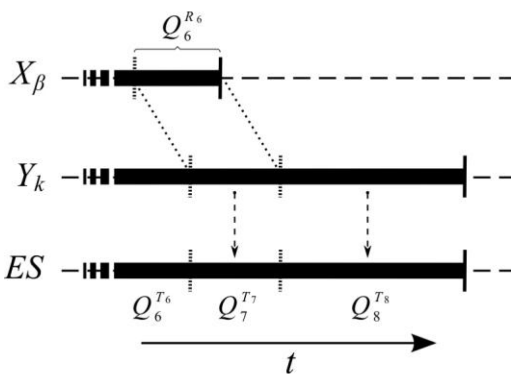


Fig. 8.: Aligning the last subsets of events without concurrences

when X_β finishes before Y_k.

4. Conclusion and Future Work

A temporal data alignment approach suitable for data fusion in a WMSN was presented. The approach was designed considering the event-streaming paradigm. One original aspect of our approach is that the data alignment is performed without using global references by translating temporal constraints to causal dependencies of the media involved. To achieve this, the *event-streaming* was defined and constructed as a finite collection *ES* of disjoint subsets, which are causally ordered and arranged one after another without interruption. As a direct consequence, each ordered set of events in an *ES* determines a specific and unique time-slot.

Extensions of this work will regard the integration of information from the spatial domain in conjunction with the temporal domain. For this, we propose to extend the alignment approach by using the principle of the *Fuzzy Causal Ordering* introduced in [20] which considers the incorporation of more than one heterogeneous domain. The integration of temporal and spatial information within a same approach will also be useful to perform filtration and association of data in WMSNs.

References

[1] I.F. Akyildiz, T. Melodia, and K.R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no.4, pp. 921-960, Mar.2007. Article (CrossRef Link)

[2] A.D. Kshemkalyani and M. Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press, New York, 2008.

[3] Punit Chandra and Ajay D. Kshemkalyani, "Causality-based predicate detection across space and time," *IEEE Transactions on Computer*, vol. 54, no. 11, pp. 1438-1453, Nov. 2005. Article (CrossRef Link)

[4] A.D. Kshemkalyani, "Predicate detection using event streams in ubiquitous environments," in *Proc. of the EUC 2005 Workshops: UISW, NCUS, SecUbiq, USN, and TAUES*, pp. 807-816, Dec. 2005.

[5] S. Reilly, "Multi-event handlers for sensor-driven ubiquitous computing applications," in *Proc. of the 2009 IEEE International Conference on Pervasive Computing and Communications*, pp.1-2, Mar. 2009.

[6] Guo-Liang Lee and Chi-Sheng Shih, "Clock Free Data Streams Alignment for Sensor Networks" in *Proc. of the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007)*, pp. 355-362, Aug. 2007.

[7] Rui Dai and Ian F. Akyildiz, "A Spatial Correlation Model for Visual Information in Wireless Multimedia Sensor Networks", *IEEE Transactions on Multimedia*, vol. 11, no. 6, pp. 1148-1159, Oct. 2009. Article (CrossRef Link)

[8] Pu Wang, Rui Dai, and Ian F. Akyildiz, "Collaborative Data Compression Using Clustered Source Coding for Wireless Multimedia Sensor Networks", in *Proc of the 29th conference on Information communications (INFOCOM 2010)*, pp. 2106-2114, Mar. 2010.

[9] Damianos Gavalas, Aristides Mpitziopoulos, Grammati Pantziou and Charalampos Konstantopoulos, "An approach for near-optimal distributed data fusion in wireless sensor networks", *Wireless Networks*, vol. 16, no. 5, pp. 1407-1425, Jul. 2010. Article (CrossRef Link)

[10] D.L. Hall and S.A.H. McMullen, "Mathematical Techniques in Multisensor Data Fusion," *Artech House Information Warfare Library*, 2004.

[11] J.M. Bahi, A. Giersch, and A. Makhoul, "A scalable fault tolerant diffusion scheme for data fusion in sensor networks," in *Proc. of the 3rd international conference on Scalable information systems(InfoScale '08)*, pp. 1-5, Jun. 2008.

[12] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, pp. 558-565, Jul. 1978. Article (CrossRef Link)

[13] S. E. Pomares Hernandez, J. Estudillo Ramirez, L. A. Morales Rosales, and G. Rodriguez Gomez,

"Logical Mapping: An Intermedia Synchronization Model for Multimedia Distributed Systems," *Journal of Multimedia*, vol. 3, no. 5, pp.33-41, Dec. 2008.
[14] K. Shimamura, K. Tanaka, and M. Takizawa, "Group communication protocol for multimedia applications," in *Proc. of the 2001 International Conference on Computer Networks and Mobile Computing (ICCNMC'01),* pp.303–308, Oct. 2001.
[15] L. Lamport, "On Interprocess Communication. Part I: Basic Formalism," *Distributed Computing,* vol. 1, no. 2, pp. 77-85, 1986. Article (CrossRef Link)
[16] J.F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM,* vol. 26, no. 11, pp. 832-843, Nov. 1983. Article (CrossRef Link)
[17] Marc B. Vilain, "A System for Reasoning about Time," in *Proc. of Second National Conference on Artificial Intelligence (AAAI-82),* pp. 197-201, Aug. 1982.
[18] P. Chandra and A.D. Kshemkalyani, "Detection of orthogonal interval relations," in *Proc. of the 9th International Conference on High Performance Computing(HiPC '02),* London, UK, pp. 323-333, Dec. 2002.
[19] P. Chandra and A. Kshemkalyani, "Data-stream-based global event monitoring using pairwise interactions," *Journal of Parallel and Distributed Computing,* vol. 68, pp. 729-751, Jun. 2008. Article (CrossRef Link)
[20] L. A. Morales Rosales, S. E. Pomares Hernandez, and G. Rodriguez Gomez, "Fuzzy Causal Ordering of Events in Distributed Systems," *Journal of Applied Sciences*, vol. 9 no. 8, pp. 1441-1449, 2009. Article (CrossRef Link)

Jose Roberto Perez Cruz is currently a Ph.D student in the Department of Computer Science at the National Institute of Astrophysics, Optics and Electronics (INAOE), in Puebla, Mexico. He holds the M.Sc. degree in computer Science from the same institute in 2009. His research interests include distributed systems, partial order algorithms, sensor networks and secure group communications. His post graduate studies are supported by the National Council of Science and Technology of Mexico (CONACYT).

Saul E. Pomares Hernandez is a Researcher in the Department of Computer Science at INAOE. He completed his Ph.D Degree at the Laboratory for Analysis and Architecture of Systems of CNRS, France in 2002. Since 1998, he has been researching in the field of distributed systems and partial order algorithms.

Enrique Munoz de Cote is an expert in machine learning, algorithms and computational game theory. He received a PhD in computer science from the Politecnico di Milano (Italy) and Rutgers University (USA) and a postdoc from Southampton University (UK). He is currently appointed as assistant professor by INAOE and works on the computational aspects of economic theory.