Clustering Schemes in Wireless Sensor Netwoks

Theodore Tsiligiridis

Informatics laboratory,
Division of Informatics, Mathematics and Statistics,
Department of Sciences,
Agricultural University of Athens,
75 Iera odos, 11855 Athens, Greece
mdrougka@aua.gr
tsili@aua.gr

***Abstract*— Sensor clustering at the network layer of Wireless Sensor Networks (WSNs) has been shown to be a scalable method of reducing energy dissipation. In this paper we propose some energy saving clustering schemes in order to compute the data gathered from sensor nodes in an overlapping zones in cluster regions. Specifically, we present an algorithm for partitioning the WSN graph into sub-graphs (or sub-clusters) enabling a considerable reduction of communication and processing overhead. Rather than individual sensor nodes transmitting directly their data to the cluster-head, they instead transmit to some other sensor node, designated as the local cluster-head, which is closest to the cluster-head and has more power than the near by sensor nodes. Data aggregation and in-network processing techniques are also presented.**

I. Introduction

Wireless Sensor Networks (WSNs) have a variety of applications mainly because of their powerful function and low energy cost. In various domains, such as national defence military affair, traffic management, long distance control of dangerous region, and so on, WSN has shown its significance and capability in application. Further interesting examples include environmental monitoring, which involves monitoring air, soil and water, condition-based maintenance, habitat monitoring, seismic detection, military surveillance, inventory tracking, smart spaces, etc. ([1], [2]). Despite their many diverse applications, WSNs pose a number of unique technical challenges, such as fault tolerance (robustness), scalability, production costs, operating environment, sensor network topology, hardware constraints, transmission media and power consumption. Future applications will extensively employ WSNs that function in real time in conjunction with communications systems, mechanical actuators, and even robots to monitor and intervene in particular processes of the applications.

A typical WSN permits remote monitoring of many parameters, depending on the type of sensors and/or actuators used and the coverage area. They may have homogeneous structures, where all sensors present similar characteristics, or heterogeneous structures, where some sensors are more powerful than others or are differentiated by physical characteristics, including the type of battery or antenna the individual sensors use, or whether specific sensors are static or mobile. In a static WSN, simple sensors are usually placed in a fixed and known locations, by means of their geographical coordinates. In other cases a group of different type of sensors may be connected physically to a sensor column, or the same sensors may be connected physically to different sensor columns, in which the sensors have the ability to monitor the environment, and collect various data, such as temperature, humidity, etc. For example, the geo-data that are measured for a specific application may include pore water pressure, ground vibration, soil moisture, tilt or acceleration and strain on the particular sensor column into which these analog sensors are placed and buried under the ground. To include both, simple and multiple sensor column cases we use the term Sensor Node (SN). However, SNs periodically sample and transmit the data at constant time intervals to the aggregating SN.

Data received at the gateway is transmitted to a data management center, using an appropriate network infrastructure allowing to install the gateway at any scalable distance. The infrastructure incorporates all appropriate technologies suitable for long distant data transmission. Data analysis and visualization tools provide the capability to compare and analyse data from different sensors in the same SN, the same sensors in different SNs, selective comparison, real streaming, etc., over the Internet. This makes it possible for the scientists around the world to analyse the data with very minimal delay and effective warning can be issued on time.

In this paper, we consider the problem of collecting data, from all SNs to the sink or Base Station (BS), in an energy-saving manner. The proposed scheme applies a hierarchical, two-phase, approach to the arbitrary topology. In the lower level the WSN will form many clusters for efficient network organization. Each cluster contains many SNs, one Cluster Head (CH) that collects and organizes the SNs, and one Local CH (LCH) that gathers and aggregates the sensor data from other SNs in the same cluster. Finally, the WSN contains a Beacon Nodes (BNs), sime times called also Master CH (MCHs), that will broadcast the time synchronization signal through the CHs to the SNs periodically, as well as and the BS. The sensor data collected by CHs are preprocessed, aggregated, and finally forwarded to the BS. By the nature of WSN, all SNs are supposed to be powered by batteries and the purpose of the scheme is to use less power in the process of data collection.

A subsequent problem tackled in this work is related with in on-demand information retrieval.There are many disadvantages of the conventional on-demand information retrieval the most important of which are summarized as follows: first, query messages are often broadcast throughout the network in a data-centric manner, without distinguishing the corresponding individual ID of SN. These on-demand data queries generate an excessive amount of redundant network traffic and energy consumption. Second, SNs reply to queries by sending data back towards the BS via various paths, such as the reversed paths obtained from query broadcasting, and most of the data are accumulated at a BS. This causes high and unbalanced energy consumption, since SNs near the BS are likely to consume more energy than SNs remote from the BS.

The rest of this paper is organized as follows. In Section II, some background information is provides whereas in Section III the model analysis is presented. Finally in Section IV some conclusions are drawn.

II. Preliminaries and System Model

In the following, we consider a simple but general enough WSN model, which consists of n SNs, say $S = \{s_1, s_2, \ldots s_n\}$, deployed in a certain geographic 2-D region, and a BS, say s_0. Each SN corresponds to a vertex in a communication graph G and two vertices are connected in G iff their corresponding SNs can communicate directly. Assuming that each SN has a transmission range r and a fixed interference range $R = \Theta(r)$, then a SN s_i can send data correctly to another SN s_j, if and only if s_j is within s_i's transmission range, and s_j is not within interference range R of any other transmitting SN.

Formally, we use the graph $G(S, E)$ to represent a WSN where S is the set of vertices (or SNs excluding s_0) in the network and E represents all the edges or links in the network. An edge $e_{ij} = (s_i, s_j)$; $s_i, s_j \in S,\ i \neq j$ exists if and only if s_i is in the transmission range of s_j and vice versa. All edges in the graph are bidirectional. The set of neighbours of a vertex s_j is represented by $N_1(s_j)$, i.e., $N_1(s_j) = \{s_i : e_{ji} = (s_j, s_i) \in E\ i \neq j\}$. The set of two-hop vertices of vertex s_j, i.e., the vertices which are the neighbours of vertex s_j's neighbours except for the vertices that are the neighbours of vertex s_j, is represented by $N_2(s_j)$, i.e., $N_2(s_j) = \{s_m : e_{im} = (s_i, s_m) \in E$, where $m \neq j$ and $s_m \notin N_1$ and $e_{ji} = (s_j, s_i) \in E\}$. The combined set of one-hop and two-hop neighbours of s_j is denoted as $N_{12}(s_j)$. We define a path from $s_i \in S$ to $s_m \in S$ as an alternating sequence of vertices and edges, beginning with s_i and ending with s_m, such that each edge connects its preceding with its succeeding vertex. If there is only one path between any two points in the graph, then G is defined as single connected graph; if there are k paths between any two points, then G is defined as k-connected graph [3]. The length of a path is the number of intervening edges. We denote by $d_G(s_j, s_m)$ the distance between s_j and s_m, i.e., the minimum length of any path connecting s_j and s_m in G, where by definition $d_G(s_j, s_j) = 0, \forall s_j \in S$ and $d_G(s_i, s_m) = d_G(d_m, s_i), \forall s_i, s_m \in S$. Note that the distance is not related to network link costs (e.g., latency), but it is a purely abstract metric measuring the number of hops.

Data collection, data aggregation, and data selection are three different data processing operations that have been extensively studied in the community of networking for WSNs. In data collection the problem is how to collect a set of data items stored in each individual sensor to the BS. If we collect data using the Breadth-First-Search (BFS) tree, as in [4], then the total number of packet relays will be the smallest. In data aggregation, the BS wants to know the value for a certain function of all data items stored in each individual SN, such as minimum, maximum, average, variance and so on. Lastly, data selection is defined as the process of determining the appropriate data type and source, as well as suitable instruments to collect data. In this sense, data selection precedes the actual practice of data collection. Data selection solves aggregation queries about order statistics and percentiles, such as to find the median, or the k^{th} smallest (or largest) value of all data items, where k could be any arbitrary value. However, little is known about distributed (network) selection, despite it is a significant part in understanding the data aggregation, especially for WSNs. Five distributive aggregations *max, min, count, sum* and *average* were carried out efficiently on a spanning tree in [5]. Subsequent work did not quite settle the time complexity, the message complexity and the energy complexity of data collection, aggregation, and selection, nor the trade-offs among these three possibly conflicting objectives.

Data processing operations may be formally presented with the help of data items A_i stored in SNs. Thus, in data collection the aim is to collect the set of data items A_i stored in each individual SN s_i to the BS s_0. Assume that $A = \{a_1, a_2, \ldots, a_N\}$ is a totally ordered multi-set of N elements collected by all n SNs, with $N = |A|$. Each SN s_i has a subset A_i of the raw data, n_i is the cardinality of A_i, namely $|A_i| = n_i$ and $A = \bigcup_{i=1}^{n} A_i$. Since A is a multi-set, it is possible that $A_i \bigcap A_j = \emptyset$.In data aggregation the purpose is to find the value $f(A)$ at the BS s_0 for a certain function f, such as *min, max, average, variance* and so on with minimum time delay. Finally, in data selection the interest is to find the kth smallest (or largest) value of the set A where k could be any arbitrary value, i.e., it solves aggregation queries about order statistics and percentiles.

As it has been pointed out the SNs aggregate their data with aim to provide accurate reports about their local regions and save from communication overhead and energy loss. In the usual approach each SN in the cluster sends its data to the CH, and the CH reports the aggregated data to the BS. To cluster the SNs a distributed randomized algorithm was used. Each SN takes a probability to become a CH, and broadcasts itself to other SNs within certain hops. The SNs that are not CHs join the closest CH. The optimal parameters of the clustering, which minimize the communication cost may also be derived. To lower down the communication cost the above algorithm was used to build a hierarchical clustering structure in [6]. However, the status of each SN may change, and therefore the overall structure may not always be optimal. For example, some SNs may use more energy to collect data, so they are dying faster than the others, resulting in the decreases of the lifetime of the whole cluster [7].

An alternative approach could be to periodically recompute the CHs based on the residual energy of each SN and its relationship to other SNs [8]. To maximize the lifetime of the WSN a hierarchical model is used that utilizes data aggregation and in-network processing at two-levels of the network hierarchy. In the first stage, a set of CHs is elected to form a fixed virtual routing architecture on, which the first level of aggregation and routing is performed. Actually, a CH is a vertex of $G(S, E)$ connected to maximum number of other vertices of G. The set of all such vertices (or CHs) define a subset S' of S connected to maximum number of vertices in S. This gives rise to define a CH formally, through the Dominating Set (DS). However, the problem of optimal selection of CHs is NP-complete since it is equivalent to the p-median problem in graph theory, which has been shown to be NP-complete [9]. In the second stage the problem is that of finding an optimal subset of CHs, called Master CHs (MCHs), which are selected to perform the second level of aggregation under the objective to maximize the network lifetime. A MCH plays the role of a BN in ad-hoc networks. They are placed outside the DS such that they are adjacent, by means of hearing, to maximum number of CHs, as well as spared CHs not connected with CHs, and also communicate with the BS.

Algorithm 1: Dominating Set-based Algorithm.

Input : A sensor network graph $G(S, E)$
Output: The Dominating Set $S' \subseteq S$ of $G(S, E)$

Step-1:
Input the adjacency matrix $A_G = [a_{ij}]$ of $G = (S, E)$.
$a_{ij} = 1$ if edge $e_{ij} \in E$ and 0 otherwise $\forall i, j = 1, \ldots, n$.
Step-2:
Create a vector $\vec{u}$ containing the row sums of A_G.
Step-3:
while *there is a vertex* $s_i \in S$ *of the highest degree of* G **do**
- Include s_i into the DS S'.;
- Remove all edges incident to s_i in the $G = (S, E)$.;
- Update the vector $\vec{u}$ of row sums.;
- **if** $A_G \neq 0$ *(or S has more vertices to add in S')* **then**
  - Move back to the beginning of step-3.;
- **else**
  - ($A_G = 0$ or S has no vertices to add in S');
  - Move forward to step-4.;

Step-4:
The vertices v form the Dominating Set S' of $G(S, E)$.

For the shake of completeness we recall that a DS of a graph $G(S, E)$ is a subset $S' \subseteq S$ such that for each vertex in S, it either belongs to S' or has at least one neighbour in S' (or, each vertex in $S - S'$ is adjacent to some vertex in S'). The Minimum Dominating Set (MDS) problem is to find S' with minimum cardinality, which is also called Domination Number (DN). **Algorithm 1** derives the MDS of $G(S, V)$, namely the set S' of all CHs in a WSN. The underline fundamental routing problem is to construct CHs in the network graph so as to rapidly and hierarchically send messages from one cluster to another. The CHs will oversee routing within and through the individual cluster. Since the members of S' are the CHs the main question still remains, namely to find the smallest set of efficient CHs. Finding a MDS is NP-hard in general, but efficient approximation algorithms do exist. However, these approximation algorithms require a central node capable of performing global computation. In our context the interest is whether the algorithm is computationally efficient, it doesn't use much bandwidth and, finally, can be performed in a distributed (local) manner in which each SN only communicates with its immediate neighbours, namely, no multi-hop packet forwarding is allowed.

III. Model Analysis

A. Clusters construction

A Connected DS (CDS) is a DS which induces a connected sub-graph. The vertices in the CDS are called the dominators, otherwise, dominatees. With the help of CDS, routing is easier and can adapt quickly to topology changes of a network. Only the vertices in the CDS need to maintain the routing information. Since there is no topology changes in the sub-graph induced by the CDS, there is no need to update the routing information, which reduces both storage and message complexities. If a dominatee wants to deliver a message to another dominatee, it first sends the message to its dominator. Then the search space for the route is reduced to the CDS. After the message is relayed to the destinations dominator, this dominator will deliver the message to the destination. In [10] a two-phase method for constructing the CDS is proposed. In the first phase a Maximal Independent Set (MIS) is formed. An Independent Set (IS) of a graph G is the vertex subset S where no two vertices in S have an edge in common. The MIS is the maximal IS, which means that it is not possible to include more vertices in S. In the second phase, the goal is to build a CDS using vertices that do not belong to the MIS. These vertices are selected in a greedy manner. At the end, the non-MIS vertex with the highest weight (the weight depends on the remaining energy and the degree of the vertex) becomes part of the CDS.

In the following we present **Algorithm 2** which forms many clusters from their CHs. The procedure starts when the BS sends a broadcast message at its maximum transmission range to the CHs. All SNs hearing the broadcast create their database and send an acknowledgement message tagged with the database details. Based on the acknowledgements the CHs decide the SNs of their cluster. As soon as the clusters are formed, the time synchronization signals will be transmitted from the MCH to the cluster SNs through their respective CHs. The SNs in the cluster start sampling at the same instant. Each CH determines its frame size and time slot size according to the number of SNs attached to it. This will help in the dynamic change of frame size and slot size, which will reduce a redundant delay that normally occurs in a TDMA channel access.

Algorithm 2: Cluster Construction

Input : A WSN graph $G(S, E)$.
Dominating Set (DS): $S' \subseteq S$.
Output: Cluster Construction.

Step-1:
Determine the CHs from DS S' of the graph $G = (S, E)$.
Step-2:
Input the adjacency matrix $A_G = [a_{ij}]$ of $G = (S, E)$.
$a_{ij} = 1$ if edge $e_{i,j} \in E$ and 0 otherwise $\forall i, j = 1, \ldots, n$.
Step-3:
Choose a CH.;
Include all neighbours of CH in the cluster.;
Step-4:
while *there are more CHs* **do**
  Go to the next CH.;
  Move to step-3.;

B. Sub-Clusters construction

As it is well known, clustering reduces channel contention and packet collisions, resulting in better network throughput under high load. Furthermore, the energy consumption in CHs is more when compared to other SNs, since they transmit data over long distances. For this reason a common policy is to allow SNs with spare energy to act as CHs. In the proposed scheme each cluster contains (apart from a CH and the SNs) a LCH which collects and aggregates the sensor data from other SNs in the same cluster. The purpose is to reduce the transmission energy spent, by aggregating the data packets in a a special SN, called LCH, having minimum power consumption and which is closer to the CH. In a case where more than one SN has minimum power consumption, it is obvious to choose the SN which is closer to the CH as LCH. As a last action in the procedure, the aggregated data would be sent to the CH from the LCH, thereby conserving the energy for transmission and processing. The CH will then forward the aggregated data coming from the LCH to the BS either directly or through other CHs.

Algorithm 3 provides an optimal clustering procedure such that the network lifetime with respect to energy is maximized. In a single cluster, say C, the LCH will aggregate the data values from the SNs and forward to CH to reduce the transmission overhead. However, if the cluster count is more than the energy handling capacity of the CH, say s_{ch}, then partition the whole cluster into two sub-clusters C_1 and C_2, where C_1 uses s_{ch} as the CH and C_2 uses the LCH, say s_{lch}, to coordinate and aggregate the data values from the respective cluster groups. The C_2 will then forward the aggregated data to the CH which lies in the CH_1.

Algorithm 3: Algorithm for Sub-cluster Construction.

Input : A WSN graph $G(S, E)$.
Dominating Set (DS): $S' \subseteq S$.
$W(s_i)$: Maximum energy capacity of a SN s_i.
$W(s_{lch})$: Maximum energy capacity of a LCH s_{lch}.
$W(s_{ch})$: Maximum energy capacity of a CH s_{ch}.
Output: Sub-cluster Construction.

Step-1:
Define the LCH s_{lch} to build C_2.
Define the CH s_{ch} to build C_1.
Step-2:
$sum_w = 0$; $W(s_{lch}) = w$ (Energy of LCH).
Step-3:
while *there is a SN* $s_i \in S$ *of the graph* $G = (S, E)$ **do**
  $sum_w = sum_w + w(s_i)$;
  if $sum_w \leq W(s_{lch})$ **then**
    Include s_i into the C_2.;
  else
    Go to next vertex.;
Step-4:
Include the remaining SNs into the C_1.

In the algorithm we assume that $W_{s_{lch}}$ and $W(s_i)$ is the maximum energy capacity of the LCH and SN, respectively. To form the C_2, each SN will be added one by one to C_2 until the maximum energy capacity $W_{s_{lch}}$ of the LCH is met. The rest of SNs which are not added to C_2 cluster are assigned to the C_1. Then we include the s_{lch} into C_2 and the s_{ch} into the C_1 as the CHs of the two sub-clusters. The algorithm presents a way the LCH at the intersection zone to aggregate the data from the SNs inside the intersection group and to form sub-clusters which can be identified from the SN database. Adopting a TDMA channel, each SN will send the corresponding SN database to the CHs from which the CH can find out the SNs in the overlapping zone between the CHs.

C. Query Resolution Mechanism

To facilitate efficient query dissemination and data aggregation in WSNs a query resolution mechanism is derived with aim to resolve the name of a query to the corresponding SNs before the query message will be forwarded to SNs. Table 1 illustrates an example of a query resolution process. We use an attribute name that describes the query's name as well as the properties of SNs. A user's query can be specified by the interested objects and attributes of sensor data. For instance, in a military application, a query could concern the smoke (attribute) status around the robot in a battlefield (object). In the same way, a SN can be described by the object that it is monitoring, and the physical attribute such as temperature or humidity that describes a SN's type.

TABLE I
QUERY-TO-ID RESOLUTION

contents !!

SN_ID Coordinates	AT: Attribute					
	AT_1	AT_2	AT_3	AT_4	...	AT_m
ID1 (x_1, y_1)	1	0	1	1	...	0
ID2 (x_2, y_2)	0	0	1	0	...	0
ID3 (x_3, y_3)	1	0	0	0	...	0
ID4 (x_4, y_4)	1	1	1	0	...	1
ID5 (x_5, y_5)	1	0	0	0	...	0
...					...	

A resolution table is adopted in the query resolution mechanism to discover the locality of SNs corresponding to a query. It maps the attribute name of each query and sensing IDs and it is implemented at the BS. The query resolution table is initially achieved by the following operations. Firstly, each SN registers its ID, location, sensing attributes and monitoring object (if the sensor knows it) with the BS and secondly, a BS constructs a table that maps sensing attributes and monitoring objects to SN_IDs. The registration can operate in conjunction with the network configuration in the phase of network formation, minimising additional overhead. Because the BS is generally a powerful node, it can maintain a large resolution table. Many applications of WSNs use location of SNs to know the network deployment, obtain sensing context, analyse sensing data, and to perform network maintenance, recovery, and task management, etc. Because a general static SN is used in a WSN, the requirements for updating the location of SN are minimal. The object that a sensor is monitoring refers to static objects that the SNs are monitoring over the long term, instead of the dynamic results that a sensor detected.

When the BS receives a query message, it resolves the name of the query (i.e. AT1, AT2, etc., in the example of Table 1) to the corresponding SN_ID, according to the resolution table. After a query′s name is translated into an ID group corresponding to sensors, the BS calculates the query area by deriving a rectangle, in which all corresponding SNs reside. Based on our previous previous work [11] it may be shown that the SNs which become the LCHs or CHs spend relatively more energy than other SNs because they have to receive information from all the SNs within their cluster, aggregate this information and then communicate to the higher level CHs or the information processing center. However, cluster-based algorithms along with data aggregation and in-network processing can achieve significant energy savings in the sensor networks.

IV. CONCLUSION

Sensor clustering at the network layer of WSNs has been shown to be a scalable method of reducing energy dissipation. In this paper we proposed some energy saving clustering algorithms in order to compute the data gathered from SNs in an overlapping zones in cluster regions. Rather than individual SNs transmitting directly their data to the CH, they instead transmit to some other SN, designated as the LCH , which is closest to the CH and has more powerful the near by SNs. The CH then sends aggregated, and possibly compressed, sensor information to the BS as a single transmission. Clearly, clustering makes some SNs more important than others, while increasing the energy dissipation of those same SNs.

The proposed a clustering based approach selects efficiently the LCHs/CHs in order to extend the WSN lifetime. Specifically, we presented an algorithm for partitioning the WSN graph into sub-graphs or sub-clusters enabling a considerable reduction of communication and processing overhead. In the analysed scheme, sensing data was collected at appropriate LCHs/CHs, at which data was aggregated and sent to the BS. Note that data aggregation and in-network processing techniques was performed at two levels. However, data aggregation was affected by several factors, such as the placement of aggregation points, the aggregation function, and the density of the sensors in the network. In this framework the determination of an optimal selection of aggregation points, by means of reducing the number of redundant data sent to the end user while preserving data integrity, is very crucial and still open problems.

REFERENCES

[1] L. Gavrilovska, S. Krco, V. Milutinovic, I. Stojmenovic, and R. Trobec, *Application and Multidisciplinary Aspects of Wireless Sensor Networks: Concepts, Integration, and Case Studies (Computer Communications and Networks)*. Springer Verlag, 2006.

[2] M. A. Hussain, "Wsn research activities for military application," in *Proc. of 11th International Conference on Advanced Communication Technology (ICACT'2009)*, IT and Telecom Dept., INHA Univ., Incheon, Feb. 2009, vol. 1, pp. 271–274.

[3] Y. Li, Y. Wu, C. Ai, and R. Beyah, "On the construction of k-connected m-dominating sets in wireless networks," *J Comb Optim*, vol. 23, pp. 118–139, 2012.

[4] O. D. Incel, A. Ghosh, and B. Krishnamachari, "Scheduling algorithms for tree-based data collection in wireless sensor networks," in *Theoretical Aspects of Distributed Computing in Sensor Networks. Monographs in Theoretical Computer Science(An EATCS Series'2011)*, vol. Part 4, 2011, pp. 422–427.

[5] Y. Li, Y. Wang, and Y. Wang, "Complexity of data collection, aggregation, and selection for wireless sensor networks," *IEEE Transactions on Computers*, vol. 60(3), pp. 386–399, Mar. 2011.

[6] K. Padmanabhan and P. Kamalakkannanand, "Energy efficient adaptive protocol for clustered wireless sensor networks," *International Journal of Computer Science Issues (IJCSI)*, vol. 8(5), pp. 296–301, 2011.

[7] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. on Mobile Computing*, vol. 3(4), pp. 366–379, 2004.

[8] K. P. Ferentinos and T. A. Tsiligiridis, "A memetic algorithm for optimal dynamic design of wireless sensor networks," *Computer Communications (COMCOM)*, vol. 33(2).

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability. A guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[10] Z. Yuanyuan, J. Xiaohua, and H. Yanxiang, "Energy efficient distributed connected dominating sets construction in wireless sensor networks," in *Proc. of (IWCMC'2006)*. Washington, DC, USA: IEEE Computer Society, 2006.

[11] M. Drougka and T. Tsiligiridis, "Data cleansing and selective query processing in sensor networks," in *Proc. in the Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services ((GEOProcessing'2012)*, Valenthia, Spain, EU, Feb. 2012.