

## ANALYSIS AND REAL-TIME VISUALIZATION OF GEO-SPATIAL DATA USING XDASH : APPLICATION TO FLAIR PROJECT

G. Berthe<sup>1</sup>, V. Rouchon<sup>2</sup>, M. Ben Gaid<sup>3</sup>, A. El Feki<sup>3,\*</sup>

<sup>1</sup> Geosciences Division, IFP Energies nouvelles, Rueil-Malmaison, France - guillaume.berthe@ifpen.fr

<sup>2</sup> Physics and Analyses Division, IFP Energies nouvelles, Solaize, France - virgile.rouchon@ifpen.fr

<sup>3</sup> Technology, Computer Science and Applied Mathematics Division, IFP Energies nouvelles, Rueil-Malmaison, France - (mongi.ben-gaid, abir.el-feki@ifpen.fr)

### Commission IV, WG IV/9

**KEY WORDS:** Pollutant dispersion, measurement data visualization, real-time visualization on maps, collaborative geo-visualization

### ABSTRACT:

The reduction of atmospheric greenhouse gas emissions is a major challenge. In this context, each natural or industrial release such as methane (CH<sub>4</sub>), carbon dioxide (CO<sub>2</sub>) has to be monitored, localized and quantified. IFP Energies nouvelles (IFPEN) is developing a mobile measurement system called Flair car whose purpose is the detection of different abnormal gas emissions. Flair car system incorporates various gas sensors, including a weather station and GPS (Global Positioning System) module, mounted on a plugin hybrid electric vehicle. This enables the real-time monitoring and the recording of geo-time-stamped gas concentration measurements. Flair map corresponds to the on board real-time visualization software.

Flair map development required two important challenges : a quick and agile software modification capability together with a real-time display of measurements on maps. In order to meet these two challenges, we adopted a software rapid-prototyping approach based on the xDash tool. In this paper, our proposed real-time data visualisation approach is first introduced. Then, the rapid-prototyping development methodology which resulted in the Flair map software is described. Finally, two main operational usages of Flair map are illustrated. The first involves real-time visualization aboard the car of the maps representing data acquisition from gas concentration sensors. The second shows the a-posteriori analysis of measurement campaigns for the purpose of methane anomalies study.

### 1. INTRODUCTION

In the context of global warming and for geological gas storages, IFPEN develops several monitoring devices allowing the detection of anomalous gas. In this context, the main challenges are the ability to detect very small variations in gas concentrations and locate their leaks accurately in the field. The Flair car equipment corresponds to the mobile monitoring version. Flair car system is a plugin hybrid electric vehicle (PHEV) that incorporates various gas sensors, such as UV spectroscopy (Elements'atmo) and IR laser spectrometry (Picarro G2210-i). A weather station (AIRMAR) records GPS positions, temperature, barometric pressure and wind direction (corrected from car motion). To exploit all this equipment, a real-time visualization software, called Flair map, was needed. Two software challenges had to be handled : a quick and agile software modification capability together with a real-time display of measurements on maps.

Handling real-time aspects with GIS (Geographic Information Systems) have been investigated in (Xu *et al.*, 2013) in application for indoor fire detection, in (Jeon *et al.*, 2019) for marine surveillance, and in (Oliveira *et al.*, 2018) for drone-based geo-referencing, but for different visualization goals, with different software methodologies and tools than those used in our work.

One technical problematic in our application was that major data acquisition and control software tools, especially those used in our context, do not include real-time capable geo-visualization features. Thus, the first software challenge in the Flair project was to extend the measurement and control system, based on the LabVIEW platform (Jeffrey, 2006), with real-time map display capabilities. This was performed by coupling LabVIEW and xDash (Ben Gaid *et al.*, 2020) tools.

Another originality of our approach comes from the use of a synchronous/reactive abstraction for the data acquisition, processing and graphical display, including geo-spatial visualization description. As the Flair system requires strong software agility, a synchronous dataflow block-diagram based approach was undertaken, thanks to the use of the xDash tool. This approach is largely used in cyber-physical systems (Broman *et al.*, 2012) modelling (Derler *et al.*, 2012), simulation and rapid-prototyping (Grzegorz *et al.*, 2008) tools like Simulink (Karris, 2008) or LabVIEW, but these tools do not have geo-spatial visualization features.

In this paper, we first describe our approach for the real-time geospatial data visualization and the underlying xDash tool. Then, we introduce the Flair car measurement system. In part 4, we illustrate the real-time geo-visualization xDash-based approach for building the Flair map application. Finally, we

---

\* Corresponding author

present the application of Flair map in methane leaks detection campaigns.

## 2. REAL-TIME GEOSPATIAL DATA VISUALIZATION WITH XDASH

### 2.1 Synchronous programming philosophy

Synchronous programming languages (Benveniste *et al.*, 1994) were developed for safely programming the control algorithms of reactive systems. In these systems, there is basic “sense-compute-act” sequence that have to be repeated. Synchronous programming languages relies on a fundamental idealization : the synchronous abstraction. In this abstraction, timing events are abstracted in logical ticks, and computations in one tick are assumed to be instantaneous. They have interesting theoretical properties, allowing their programs to be formally validated.

In our application, we have to handle an analogous periodic “sense-compute-display” sequence. The execution engine of the used xDash tool, relies on these principles.

### 2.2 Rapid-prototyping

State of the art measurement and control software allows for quickly developing application logic (Broman *et al.*, 2012) by assembling blocks of functions for forming a block-diagram (Karris, 2008), following the synchronous programming paradigm. Then, program variables are associated to widgets by drag and drop. As we work in part of a research project, where chemical species to measure as well as measurement equipment change frequently, our software needs to have this agility.

### 2.3 xDash

xDash is a web-based software, whose ambition is to allow technicians, scientists or engineers, not specialists in web technologies, to build and share their own interactive web applications. xDash follows rapid-prototyping principles, described above.

#### 2.3.1 The xDash execution engine

The xDash execution engine implements a synchronous/reactive scheduling algorithm. Computation modules, called datasources, are the nodes of a direct acyclic graph (DAG) where edges are data dependencies. This allow users to express computations to perform as block diagrams whose execution follows synchronous principles. Datasources produce JSON (JavaScript Object Notation) variables that are kept into a common workspace of xDash.

The graph is executed on the web-browser and data (or operations) flow as JSON variables. In fact, JSON is simple and versatile: it can handle simulation run results, CSV(Comma-Separated Values) information, geographic information (like GeoJSON), and, even web-services results.

Datasources in xDash are stateless and can be periodic or not. A datasource may have one or more inputs, when it hasn't an input it is called a “source”. However, a datasource must have at least an output (i.e. the result). Each datasource have these possible status:

- “None”: is the initial status and occurs at the beginning when a datasource is not executed yet. It can also occurs when the execution of a datasource is interrupted by user for example
- “Pending”: is the intermediate status and occurs all along datasource execution (computation)
- “OK”: is a final status and occurs when a datasource finishes its computation with success
- “Error”: is the opposite of “OK” status, it occurs when errors happen during computation

The DAG is executed using a single thread scheduler and can handle the asynchronous nature of web-services. Each operation (datasource) cannot be executed until all its predecessors have been executed and returned the “OK” status. If a datasource returns “Error” status, the sub-graph, to which all its successors belong, will not be executed. Besides, any datasource can be forced to be executed by the user. In that case, the scheduler algorithm verify if all its predecessors have been computed in the past before allowing its execution. If at least one of its predecessors hasn't the “OK” status, user will be informed that he couldn't update this datasource.

#### 2.3.2 xDash visualization components

xDash widgets are based on various open source JavaScript libraries for scientific visualization.

First use-cases with xDash involved mobility-related APIs (Application Programming Interfaces) validation needs. Thus several web-based mapping libraries were evaluated. We finally selected Leaflet.js (Crickard III., 2014) library for several reasons. Leaflet.js has several advantages. First, a clean and well-designed object model. Also, a complete and simple documentation. Finally a wide range of available plugins shared by an active community.

For specific Flair map project requirements, quick real-time rendering was a supplementary pre-requisite. Leaflet.js successfully fulfilled this requirement.

## 3. FLAIR MEASUREMENT SYSTEM

Flair car (Figure 1) comprises a pump allowing gas sampling at 2.5 and 0.5 meters from the ground level. It contains several sensors such as a Picarro G2201-I analyzer, GPS and weather stations (AIRMAR), an Elements'atmo Ultraviolet (UV) spectrometer, a computer and a real-time visualization software. The whole gas sampling material (such as tedlar bags, drill, pump) is also brought to complete on-site measurements with further laboratory tests (Rouchon *et al.*, 2017).

Methane (CH<sub>4</sub>), ethane (C<sub>2</sub>H<sub>6</sub>), carbon dioxide (CO<sub>2</sub>), water vapor and δ<sup>13</sup>C (the stable isotopic composition of carbon) in CH<sub>4</sub> molecules are recorded by the Picarro analyzer. The Elements'atmo UV spectrometer is mainly dedicated to detect odorous tetrahydrothiophene (THT) and mercaptan molecules. The AIRMAR weather station records temperature, barometric pressure, wind velocity and wind direction (corrected from car motion). Flair map corresponds to the real-time viewing software on board the PHEV.

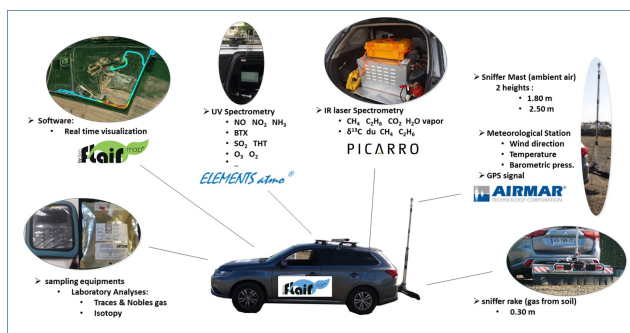


Figure 1. Mobile monitoring device: Flair car

#### 4. REAL-TIME VISUALIZATION BY FLAIR MAP

During field measurement campaigns, geo-time-stamped gas concentration measurements have to be displayed in real-time to guide the car driver in its search for methane leaks or other anomalies. This relies on the following architecture.

##### 4.1 Hardware/Software architecture

Sensor measurements acquisition, centralization, scaling and time-stamping is handled using a LabVIEW-based software running on an rugged PC. A Wi-Fi connection links the Tablet executing xDash and the rugged PC executing LabVIEW (Figure 1).

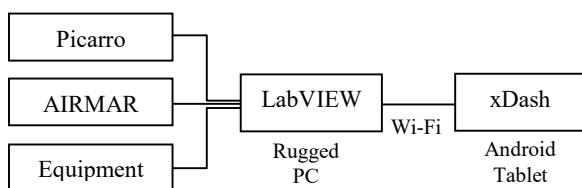


Figure 1. Hardware/Software architecture diagram

LabVIEW has a REST (representational state transfer) HTTP (Hypertext Transfer Protocol) server, allowing the exposition of all handled data using this protocol. This directly fits the xDash REST HTTP client features. Data exchange between LabVIEW and xDash uses the REST/HTTP protocol over the Wi-Fi connection.

##### 4.2 Using xDash for developing the real-time display software

An xDash project was created. Three dependant datasources were defined :

- **measurements** : REST web-service call to LabVIEW to poll for measurements. Executed every 1 second. Measurement are received as JSON object
- **CH4\_heatmap\_config** : JSON configuration of display of measurement points on the map : min and max for measurements colour scale, measurement point radius, opacity, and auto-scale options.
- **CH4\_data\_point** : aggregated JSON object from previous datasources outputs, to be displayed by the Leaflet-based mapping widget. Each single measurement is displayed as a circle whose colour is defined by the colour map ranging between min and max values

Datasources outputs are illustrated in Figure 2.

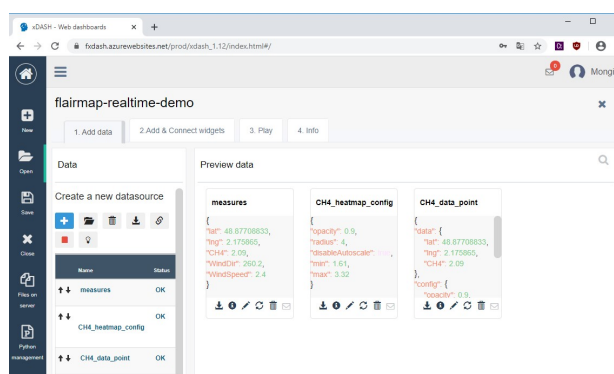


Figure 2. Flair map real-time project, open in xDash editor.  
Preview of datasources JSON output

Their execution graph in xDash is shown in Figure 3. This graph expresses both data and execution flow dependencies.

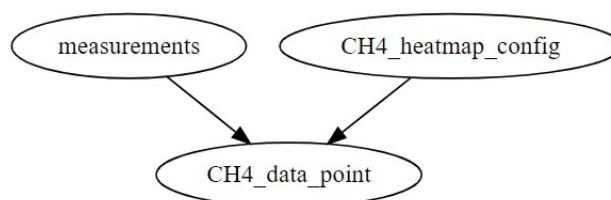


Figure 3. Datasources graph, for Flair map real-time xDash project

After describing the real-time application logic, the dashboard is built using the drag and drop of widgets. The main widget is a Leaflet.js map. The dashboard also include a wind rose (Figure 4), a double-slider and wind speed numeric display.

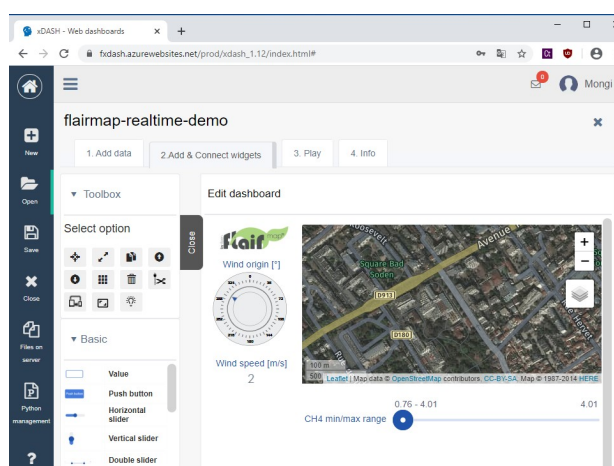


Figure 4. Flair map real-time project, open in xDash editor.  
Preview of widgets edition

Widgets are then connected to JSON variables describing display. In particular, **CH4\_data\_point** datasource output is connected to the Leaflet map widget, as illustrated in Figure 5.

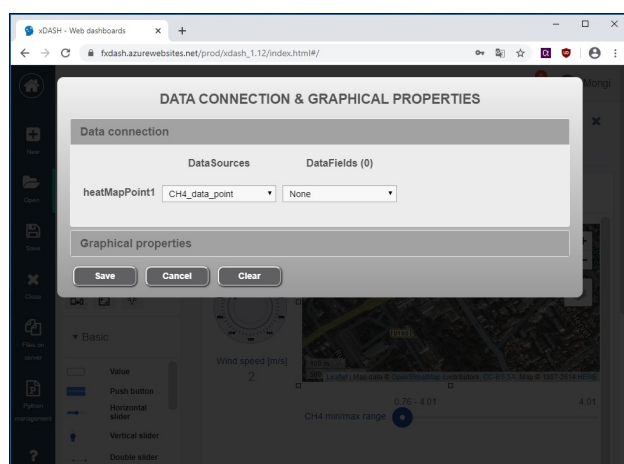


Figure 5. Flair map real-time display project, connection of map widget to measurement rendering JSON description

Finally, a standalone web-page is generated from xDash. It can be displayed using a web-browser. It constitutes “Flair map real-time” visualization software. This page can also be converted to an Android or an iOS app using the Apache Cordova software (Camden, 2015).

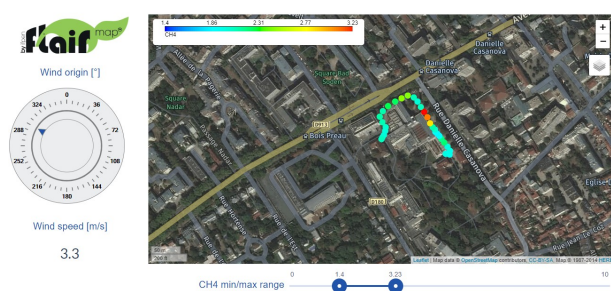


Figure 6. Flair map real-time display software, used on-board Flair car, running on an Android tablet

## 5. FLAIR MAP APPLICATIONS

### 5.1 Real-time leak detection

Detection of abnormal gases in the ambient air corresponds to very small variations in gas concentrations. One important aspect of real-time analysis is the ability to change live the min and max range of displayed points colour map, using the double-slider widget. The wind direction and speed guide the driver to find gas leaks.

Real-time visualization is useful for optimizing gas sampling. Indeed, in the case of diffusive leaks from the ground (recorded from the Flair car sniffer rake), real-time visualization makes it possible to precisely locate the sampling area. The three maps below in Figure 7 show the simultaneous recordings of methane ( $\text{CH}_4$ ), ethane ( $\text{C}_2\text{H}_6$ ) and the relationship between these two molecules ( $\text{C}_2 / \text{C}_1$ ) near the gas injection wells. These maps show exactly where the wellhead is leaking. The  $\text{C}_2 / \text{C}_1$  ratio equal to 5% corresponds to the ratio found in natural gas. Thanks to the real-time visualization (Flair map), soil and gas samples taken in this area have confirmed the diffuse leak from the well.

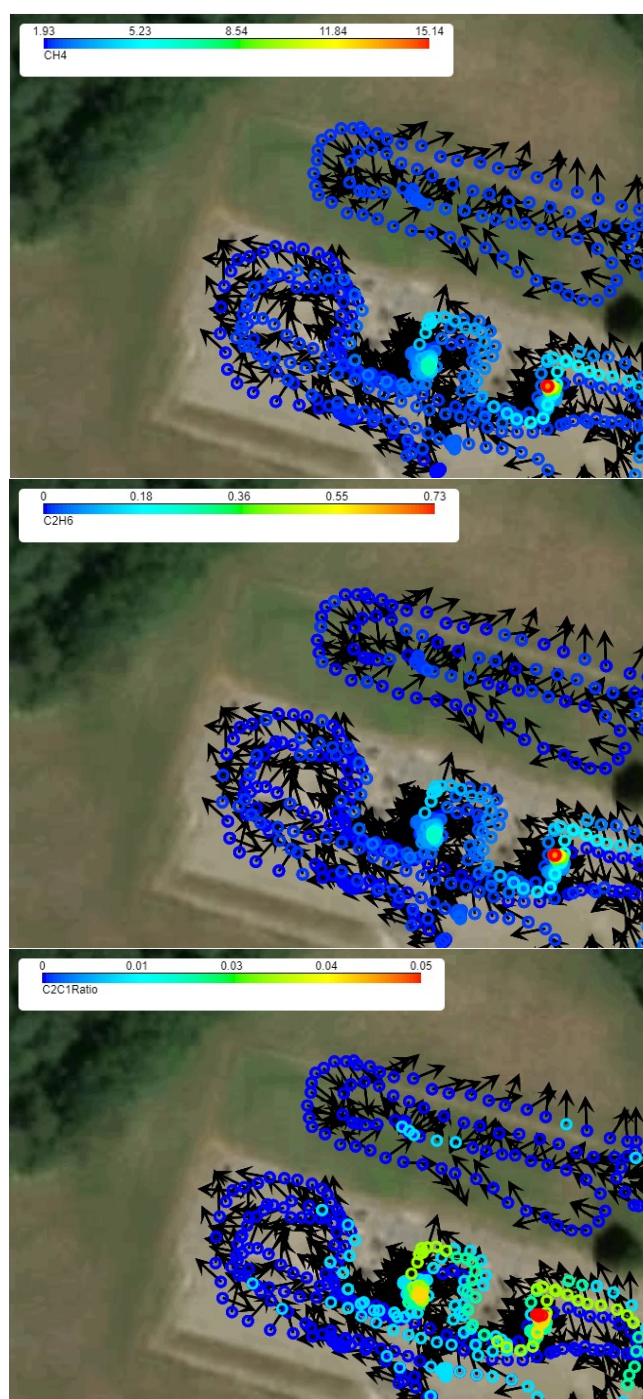


Figure 7. Simultaneous record of methane, ethane and their ratio and visualization by Flair map. Arrows represent measured wind origin and pointing the source.

In the context of field campaigns, the possibility of modifying the xDash project by any user was very helpful, in particular for correcting few bugs on the field. One field example was the need to correct sign error in the NMEA (National Marine Electronics Association) 0183 conversion to JSON from GPS module.

### 5.2 Post-analysis of measurements

LabVIEW records the field measurement in several CSV files.





Figure 8. Post-analysis Flair map dashboard : CSV

These CSV files are then opened in Flair map post-analysis dashboard for the purpose of analysis and filed report generation. This dashboard allows also to display statistical information about the wind, as shown in Figure 9.

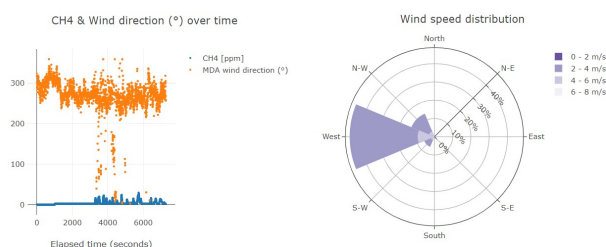


Figure 9. Post-analysis Flair map dashboard : wind statistics

## 6. CONCLUSIONS AND PERSPECTIVES

This paper presented the Flair map software and its application for methane leaks detection. The building of the Flair map application using xDash was detailed. The underlying methodology is general and can be applied to other geo-spatial data acquisition projects. The benefits of using rapid-prototyping methodology were also illustrated.

As collected data size increases, the need for a database is increasing. Future version will rely on a Warp10 ([www.warp10.io](http://www.warp10.io)) database system for data storage. Plume modelling based on data recorded by the Flair car is now under study and will soon be added to the post-analysis software.

## REFERENCES

Ben Gaid, M., El Feki, A., Lety, B., Pernet, N., 2020: xDash : Rapid-prototyping of Cyber-physical Systems in Web Browser. In *14th MODPROD Workshop on Model-Based Cyber-physical Product Development*, Linköping University, February 4-5, 2020

Broman, D., Lee, E. A., Tripakis, S., Törngren, M., 2012 : Viewpoints, formalisms, languages, and tools for cyber-physical systems. In *Proceedings of the 6th International Workshop on Multi-Paradigm Modeling (MPM '12)*, Association for Computing Machinery, New York, NY, USA, 49–54, doi.org/10.1145/2508443.2508452

Benveniste, A., Caspi, P., Le Guernic, P., Halbwachs, N., 1994: Data-flow synchronous languages. In: *de Bakker J.W., de Roever W.P., Rozenberg G. (eds) A Decade of Concurrency*

*Reflections and Perspectives. REX 1993. Lecture Notes in Computer Science*, vol 803. Springer, Berlin, Heidelberg

Camden, R. K., 2015: *Apache Cordova in Action (1st. ed.)*. Manning Publications Co., USA.

Crickard III, P., 2014 : *Leaflet.js essentials*. Packt Publishing Ltd

Derler, P., Lee, E. A., Sangiovanni Vincentelli, A., 2012 : Modeling Cyber-Physical Systems, *Proc. of the IEEE*, vol. 100, no. 1, pp. 13-28, doi.org/10.1109/JPROC.2011.2160929

Grzegorz, S., Tomasz, Z., Andrzej, B., 2008 : Rapid control prototyping with Scilab/Scicos/RTAI for PC-based ARM-based platforms, *2008 International Multiconference on Computer Science and Information Technology*, Wisia, 2008, pp. 739-744.

Jeffrey, T., 2006: *LabVIEW for everyone : graphical programming made easy and fun*. Kring, Jim. (3rd ed.). Upper Saddle River, NJ: Prentice Hall. ISBN 0131856723. OCLC 67361308

Jeon, I., Ham, S., Cheon, J., Klimkowska, A. M., Kim, H., Choi, K., and Lee, I, 2019.: A real-time drone mapping platform for marine surveillance, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-2/W13, 385–391, doi.org/10.5194/isprs-archives-XLII-2-W13-385-2019.

Karris, S., 2008 : *Introduction to Simulink with Engineering Applications*, Second Edition (2nd. ed.), Orchard Publications, ISBN: 978-1-934404-09-6

Oliveira, R. A., Khoramshahi, E., Suomalainen, J., Hakala, T., Viljanen, N., and Honkavaara, E., 2018: Real-time and post-processed georeferencing for hyperspectral drone remote sensing, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-2, 789–795, doi.org/10.5194/isprs-archives-XLII-2-789-2018

Rouchon, V., Sabattié, J-M, Colin, L., Tricard, L., Parent, F., Lievin, K. and Ben Gaid, M, 2017: GasMap : a high sensitivity photo acoustic, vehicle-based system for mapping underground fugitive methane emissions. In *Industrial Methane Measurement Conference, PEFTEC*, Antwerp, November 2017

Xu, W., Zhu, Q., Zhang, Y., Ding, Y., and Hu, M., 2013: Real-time GIS and its application in indoor fire disaster, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XL-2/W2, 121–127, doi.org/10.5194/isprsarchives-XL-2-W2-121-2013