Detecting and monitoring plumes caused by major industrial accidents with JPLUME, a new software tool for low-resolution image analysis

Nektarios Chrysoulakis a,*, Nektaria Adaktylou b, Constantinos Cartalis b

a Foundation for Research and Technology − Hellas, Institute of Applied and Computational Mathematics, Regional Analysis Division, FORTH/IACM, Vassilika Vouton, P.O. Box 1527, GR-71110, Heraklion, Crete, Greece
b University of Athens, Department of Physics, Division of Applied Physics, Greece

Received 10 December 2003; received in revised form 10 April 2004; accepted 6 July 2004
Available online 27 July 2005

Abstract

A new software tool for the automatic detection and monitoring of plumes caused by major industrial accidents is described. This tool has been designed in order to use near real time information as provided by satellite images, perform sophisticated image analysis and elaborate a user-friendly operational environment for the detection of plumes caused by major industrial accidents. The methodology, based on NOAA/AVHRR (Advanced Very High Resolution Radiometer) imagery, uses a two-dimensional feature space in order to discriminate pixels that contain plumes from those that correspond to clouds or to the underlying surface. The two-dimensional feature space is generated by combining AVHRR channels 1 (visible), 2 (near infrared) and 5 (thermal infrared). The software tool proposed has been coded in JAVA2 language, using the concepts of interoperability and object-oriented programming. This study demonstrates the applicability of the tool for the detection of a plume caused by a massive explosion in a fireworks factory in Enschede, The Netherlands, on May 13, 2000. The effectiveness and reliability of the software tool was found to be satisfactory, as plume was automatically detected and discriminated from the underlying surface.

Keywords: Industrial accidents; Plume detection and monitoring; Remote sensing; JAVA programming

1. Introduction

The total number of major industrial accidents reported each year in the EU from 1985 to 1999 shows a steady increase, with the maximum number reported during 1998. This may be due to a number of factors, including increased industrial and other economic activity and increased population densities around potentially hazardous sites, only partly compensated for by increased awareness and safety measures (EEA, 2003). An improved approach to safety and environmental management has been adopted, following the advent of the SEVESO II directive.

The European Council Directive 96/82/EC of 9 December 1996 concerning the control of major-accident hazards involving dangerous substances (SEVESO-II) aims at preventing such accidents and limiting their consequences for man and the environment, with a view to ensuring high levels of protection throughout the Community. This Directive has replaced Directive 82/501/CEE (SEVESO I). It is first to introduce substances considered dangerous for the (aquatic) environment in its scope. Industrial accidents can be connected to a number of different events and processes,
including spillage, sudden release of materials, fire, or explosion. The most common effect is the release of gases and liquids used and processed in the installations concerned. Fire and explosion are common effects as well, while a combination of the above is not rare. Releases to land, water or the air may be toxic. Airborne releases usually develop in plumes, which can thereafter be monitored either due to their optical depth or their temperature difference from the ambient air. More often, accidental discharges imply heavier than air gases characterized by a slumping phase giving high and toxic concentrations (Dandrieux et al., 2003). Damages may thus occur both as an immediate and direct consequence of the accident, as well as during the propagation and dispersion of the resulting plume. It should be mentioned that while the immediate ground-level effects in close vicinity to the installation have been examined in detail, limited attention is usually paid to the plume’s impacts in the wider geographic region affected during the hours following the accident.

In recent years, and after a number of incidents involving fires in industrial installations and warehouses, research has been oriented towards the definition of the properties and of the amount of the plume particulates generated by different materials, including pesticides, under various fire conditions (Lang, 1993; Bartelds et al., 1993; Atkinson and Jagger, 1994; Miles and Cox, 1994; Grant and Drysdale, 1994; Martins and Borrego, 1994; Marliere, 1996; Porter et al., 1996; Martins et al., 1996; Cozzani et al., 1996).

In several studies, (Kaufman et al., 1990; Ferrare et al., 1990; Kaufman et al., 1992; Cahoon et al., 1994) satellite data have been used for the analysis of the emitted smoke in order to quantify the gaseous output from forest fires. Ackerman and Toon (1981), Kaufman (1987) and Fraser et al. (1984), have related the carbon content of the plume to the single scattering albedo, as well as to the light extinction of the plume. Chung and Le (1984) have examined the feasibility of using satellite imagery to detect large-scale pollution episodes. Christopher and Chou (1997) have used a combination of spectral and textural measures in order to visually separate the plume aerosols from the underlying background. Baum and Trepte (1999) proposed a grouped threshold method for scene identification in NOAA/AVHRR (Advanced Very High Resolution Radiometer) imagery that may contain clouds, fire, smoke plumes or snow. Chrysoulakis and Cartalis (2003a) have proposed a software tool for the detection of major fires caused by technological accidents with the use of AVHRR imagery.

There is a specific philosophy in the use of satellite, since they constitute a trustworthy means to obtain information that does not depend upon the local structure. It should be stated that the GMES (Global Monitoring for Environment and Security), which is a joint initiative of the European Commission and the European Space Agency, has currently taken a very important initiative for the identification and standardisation of satellite data services that can be used for environmental management in general, and for risk management in particular (GMES, 2003).

In this paper a new software tool for the detection and monitoring of plumes caused by major industrial accidents is described. The software is named JPLUME and is coded in JAVA2 language using the JDK (Java Development Toolkit) 1.2.2 (Sun, 1999). JPLUME has been evaluated for four past accidents: in Enschede, the Netherlands (May 13, 2003) in Genoa, Italy (April 13, 1991); in Lyon, France (June 2, 1987) and in Kalohori, Greece (February 24, 1986). In this study, an AVHRR image acquired over the broader area of the Netherlands on May 13, 2000 (14.44 UTC) is used to present the functionality of the software tool proposed. This date refers to a massive explosion in a fireworks factory in the town of Enschede.

Several models and decision taking help tools exist (Quaranta et al., 2002; Bellasio and Bianconi, 2005) to assess the consequences of the different possible accidents. Software packages aiming at industrial risk assessment have already been developed; some of these software packages are WHZAN (Technica, 1992), RISKIT (VVT, 1993), EFFECTS (TNO, 1991), SAVE (TNO, 1992), MAXCRED (Khan and Abbasi, 1999) and OSIRIS (Tixier et al., 2002). JPLUME differs from the existing software packages in the following areas:

(a) it is a detection rather than a forecasting tool;
(b) it is not limited to the specific location of an industrial complex, but it is scalable, it may be applied for the monitoring of areas covering a single industrial installation, as well as for extended areas;
(c) as a JAVA developed programme it is independent of the hardware and the platform, a fact which provides the software with interoperability; additionally, users can easily modify the source code of the programme using the JDK 1.2.2 which is free-ware (Sun, 1999);
(d) JPLUME offers a window-based user interface and is user friendly. It carries menus, buttons and a flow chart of the software’s algorithm as well as an Image Viewer which gives the user the opportunity to follow the algorithm’s various steps.

2. Data and methodology

JPLUME uses AVHRR images as inputs. AVHRR has a spatial resolution of 1.1 km at the nadir, a swath coverage of 2700 km and a revisiting time of approximately 3 h (synergy of NOAA 12, 15, 16, and 17 polar orbiting satellites). AVHRR records incoming radiation in five spectral channels (μm): 0.58–0.68 (visible),
0.72–1.10 (near infrared), 3.55–3.93 (mid-infrared),
10.5–11.3 (thermal infrared) and 11.5–12.5 (thermal infrared).

An analysis of the physical background of the procedure
used to differentiate between plumes and clouds, as well as between plumes and underlying surfaces
has been given in previous studies (Chrysoulakis, 2000;
Chrysoulakis and Cartalis, 2003b). The algorithm used
in JPLUME carries the advantages of multispectral
analysis and provides important results for the detection
and monitoring of plumes caused by industrial accidents.
The main steps of the algorithm are:

(a) Calibration of channels 1, 2 and 5 of the AVHRR
image. The calibration process is based on the
conversion of the digital numbers to brightness
temperatures for thermal infrared channel and to
reflectance values for the visible and near infrared
channels (Kidwell, 1998; Goodrum, et al., 2000).
(b) Production of a pseudochannel image of a cloud
masking filter named CLD (Chrysoulakis, 2000).
CLD is used for the separation of pixels that contain
plumes from the pixels that correspond to clouds.
(c) Production of a pseudochannel image of the well
known NDVI (Normalized Difference Vegetation
Index). NDVI is used for the separation of pixels
that contain plumes from the ones that correspond
to the underlying surface.
(d) Application of the CLD and NDVI digital filters
and production of a pseudochannel image of the
two-dimensional feature space which is produced
with the use of both filtered pseudochannels CLD
and NDVI.
(e) Detection of pixels with values greater than a given
threshold in the pseudochannel image of the CLD
and for pixels with values lower than a given
threshold in the pseudochannel image of the NDVI.
Pixels meeting both conditions mentioned are
classified as plumes (Chrysoulakis and Cartalis,
2003b).

The implementation of the algorithm’s steps men-
tioned above, is performed by different modules in
JPLUME. In particular, each module consists of one or
more programme classes, which have been designed
using the object oriented JAVA2 language. The funda-
mental unit in object-oriented programming is the
“object”, consequently, languages which follow object-
oriented concepts describe the interaction among
objects. There are no stand-alone constants, variables
or functions. Any information or data are accessed
through classes and objects. A class is a type definition,
whereas an object is a variable declaration. Classes
capsulate objects, whereas a single class can be used to
instantiate multiple objects. The image manipulation in
JPLUME is achieved with the use of the immediate
mode imaging model of JAVA 2DTM API. The JAVA
2DTM API is a set of classes for advanced two-
dimensional graphics and imaging. The immediate mode
imaging model supports fixed-resolution images stored
in memory. The model, in which a number of classes and
interfaces are used (Sun, 1999), also supports filtering
operations on image data. The JAVA classes used to
instantiate objects in JPLUME are presented in Table 1.

JPLUME converts the original AVHRR channels to
“BufferedImage” objects with the use of BufferedImage
class capabilities. A BufferedImage object contains two
other objects: a “Raster” and a “ColorModel”. The
Raster class provides the possibility for image data
management. It demonstrates the rectangular coordi-
nates of the image, maintains image data in memory,
and provides a process for the creation of multiple sub-
images from a single image data buffer. It also enables
the assessment of specific pixels within an image.
A Raster object consists of two other objects: a
“SampleModel” and a “DataBuffer”. The SampleModel
class interprets data in the buffer and presents it as
individual pixels or rectangular ranges of pixels. The
DataBuffer class keeps the pixel data in memory and
access to them is performed at byte level. Taking into
account that AVHRR images are structured in 10 bits
format (2^10 = 1024 grey levels for each pixel), it is
obvious that 16 bits (2 bytes) are used for the
representation of each pixel. Therefore, each AVHRR
channel is stored in memory as a 16-bit image which
contains unsigned short values. One-dimensional arrays
are used for this storage. The digital number of the i
element of a DataBuffer object (DN_{(i)}) corresponds
to the digital number of the pixel with coordinates (x, y)
of the respective Raster object. The correspondence is
defined using the formula:

\[ DN_{(i)} = DN_{(i \times W + x)} \]  

where \( W \) is the number of columns of the AVHRR
image.

The ColorModel class provides a colour interpreta-
tion of the pixel data provided by the sample model of
the image. It defines methods for converting a pixel
value to a colour value in its associated colour space.
The class “ColorSpace” provides predefined ColorSpace
objects for any type of image. The dimensions of the
colour space are defined by the number of basic colours
used as components (Pantham, 2000).

3. Design and application of JPLUME software

In this section JPLUME is described in detail with
regard to its design and use. Fig. 1 demonstrates the
The software user interface, which presents the options available in JPLUME. A flow chart of the main stages of the algorithm appears on the right part of this window. The series of buttons appearing on the left part of the main window activates the modules which implement the various parts of the JPLUME algorithm. Each module includes one or more classes of the object-oriented code. Each class creates one or more objects which are called up by the programme on the appropriate time.

### Table 1

<table>
<thead>
<tr>
<th>JAVA class</th>
<th>Arguments used in JPLUME</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataBuffer</td>
<td>The name of the one-dimensional array, which contains the AVHRR image data (digital numbers of the pixels of a single AVHRR channel). The length of the above mentioned array (the number of its elements).</td>
</tr>
<tr>
<td>Sample model</td>
<td>The variable DataBuffer.TYPE_USHORT which represents the AVHRR data type. The variable $W$ which represents the number of columns of the image. The variable $H$ which represents the number of rows of the image. The number 1 which declares that if the image consists of more than one channels (bands), the pixels of each channel should be stored separately into the DataBuffer object (banded model structure). The variable bandOffsets which represents the position of the first pixel of each band into the DataBuffer.</td>
</tr>
<tr>
<td>Raster</td>
<td>The SampleModel object. A Databuffer object dependent on the AVHRR channel. A Point object which defines the upper left corner of the projection of the image.</td>
</tr>
<tr>
<td>ColorSpace</td>
<td>The ColorSpace.CS_GRAY instance (single channel representation) The ColorSpace.CS_sRGB instance (multiple channel representation)</td>
</tr>
<tr>
<td>ColorModel</td>
<td>The ColorSpace object. A matrix object which declares the number of the bits (10) should be used for the projection of each pixel. The Boolean false which declares that there are not transparent pixels. The Boolean false which declares that Alpha channel is not considered. The number 1 which declares that the banded model structure is used. The variable DataBuffer.TYPE_USHORT which represents the AVHRR data type.</td>
</tr>
<tr>
<td>BufferedImage</td>
<td>The ColorModel object. A Raster object dependent on the AVHRR channel. The Boolean true which declares that there are no transparent pixels. The variable null which declares that there are no other image properties to take into account.</td>
</tr>
</tbody>
</table>

Fig. 1. The JPLUME user interface.
Fig. 2 shows the object-oriented architecture of JPLUME which is organised in four essential stages. It can be seen that an Image Viewer, activated in a separate window, is also included in JPLUME. This Viewer can be used for the presentation of both input and output images and intermediate products as well. Three image input files (AVHRR channels 1, 2 and 5) and three input calibration files are necessary in order to implement JPLUME. Image files are raw NOAA/AVHRR data which can be retrieved from NOAA ground receiving stations. They can also be extracted from the standard NOAA Level 1b format. All image files must have the suffix .nek in order to be recognised by JPLUME (i.e. channel5.nek). Each file in .nek format contains a single AVHRR channel provided by the geometrically corrected AVHRR original image. All files must be uncompressed with no metadata information, since pixels in .nek format have been coded in 2 bytes words (16-bit images). The AVHRR image of May 13, 2000 (14.44 UTC) was used in this paper for the application of JPLUME.

The three input files remaining, are the calibration files provided in ASCII format and take the suffix .txt (text files). Each .txt file contains 1024 rows. The number of each row corresponds to a grey level of the 10 bits images (0–1023). Each row consists of the brightness
temperature or reflectance values which have been calculated according to equations given in the NOAA Polar Orbiter Data User’s Guide (Kidwell, 1998) and in the NOAA KLM User’s Guide (Goodrum et al., 2000). In the sections following, the function and structure of the JPLUME model are analysed, through the description of each button’s functionality, when activated.

3.1. Image Viewer module

This module is activated by pressing the “Image Viewer” button. The new window appearing on the screen contains buttons which allow the projection of input or intermediate image files. Ten different classes have been coded for the presentation of these images on the screen. In Fig. 3 the AVHRR channel 5 image of May 13, 2000 (14.44 UTC) is presented using this Viewer.

3.2. Calibration module

This module is activated by pressing the “Calibration” button. The new window opening, contains buttons which allow the calibration of each input image file. Three different classes have been coded for the image calibration. The digital number corresponding to each pixel of the original image is used as index in the lookup table and is included in the respective calibration file in order to specify the pixel’s reflectance or brightness temperature value. This value is then assigned to a pixel of a new Buffered Image. The position (row, column) of this pixel in the new image is the same as the position of the original pixel in the AVHRR image. Therefore, during the calibration process, a new Buffered Image is created for each AVHRR channel. For example, the calibration process for channel 1 receives channel1.nek and table1.txt as inputs and generates the file channel1calibr.nek.

3.3. Step 1 module

This module includes two sub-modules: The “CLD (Step 1)” and the “NDVI (Step 1)”. CLD (Step 1) is activated by pressing the respective button. The JPLUME class which is called up, is executed at the background using the calibrated channels 1 and 5 files in order to produce CLD*. The file produced takes the name cld1.nek. For each pixel in this pseudochannel image, CLD* value is given by the formula:

\[
\text{CLD}^* = \frac{R_5 - R_1}{R_5 + R_1}
\]

where \( R_5 \) is the AVHRR channel 5 radiance; \( R_1 \) is the AVHRR channel 1 radiance.

Chrysoulakis and Cartalis (2000, 2003b) have denoted that CLD* values lower than a given threshold correspond to clouds. CLD* thresholds from 0.85 to 0.95 can be used for cloud filtering in JPLUME. In practice, pixel values lower than 0.90 are set to zero by CLD (Step 1) sub-module.

NDVI (Step 1) is activated by pressing the respective button. The JPLUME class which is called up, is performed at the background using the calibrated channels 1 and 2 files in order to produce NDVI*. The file produced is named ndvi1.nek. For each pixel in this pseudochannel image, NDVI* value is given by the formula:

\[
\text{NDVI}^* = \frac{R_2 - R_1}{R_2 + R_1}
\]

where \( R_2 \) and \( R_1 \) are the AVHRR channels 2 and 1 radiance, respectively.

Negative NDVI* values threshold correspond to water bodies or clouds (Chrysoulakis and Cartalis, 2003b). In JPLUME, negative pixel values are set to zero by NDVI (Step 1) sub-module.

![Fig. 3. Presentation of the AVHRR channel 5 image of May 13, 2000 (14.44 UTC) with the use of Image Viewer module.](image-url)
3.4. Step 2 module

This module produces CLD and NDVI pseudochannels filtered both for clouds and water bodies. It includes two sub-modules: The “CLD (Step 2)” and the “NDVI (Step 2)”. CLD (Step 2) is activated by pressing the respective button. The JPLUME class which is called up, is executed at the background and uses files cld1.nek and ndvi1.nek in order to filter CLD* pseudochannel for water bodies and clouds. Pixels with zero NDVI* values are used to produce a digital mask appropriate for this filtering process. The output file receives the name cld2.nek. Fig. 4 shows the final CLD image.

NDVI (Step 2) is activated by pressing the respective button. The JPLUME class which is called up, is executed at the background and uses files cld1.nek and ndvi1.nek in order to filter NDVI* pseudochannel for clouds. Pixels with zero CLD* values are used to produce a digital mask applicable in this filtering process. The file produced is named ndvi2.nek.

3.5. Plume module

This module is activated by pressing the “Plume Detection” button. Following, a new window opens containing 2 different buttons. Each button activates one sub-module, respectively. The two sub-modules included in this module are: The “Plume Detection and Monitoring” and the “Export”. Plume Detection and Monitoring uses cld2.nek and ndvi2.nek files and produces JPLUME’S final output, whose projection is demonstrated on the screen. The final JPLUME product is a two-dimensional feature space (CLD—NDVI) image on which the plumes can be detected and monitored. The two-dimensional feature space is developed by superimposing the pseudochannels CLD and NDVI. This is practically achieved by creating a pseudo-coloured RGB image: CLD, NDVI, 0, with the double-masked CLD pseudochannel set to RED, the double-masked NDVI pseudochannel set to GREEN, and all values of the BLUE channel set to zero. Chrysoulakis and Cartalis (2003b) have shown that while the CLD values of pixels that contain plumes are high, the NDVI values are relatively low. Consequently, plumes should appear in dark red in the pseudo-coloured image. Both the CLD and the NDVI values of pixels that contain land surfaces are high. Therefore, land surfaces should appear in green in the pseudo-coloured image. The pseudo-coloured image resulting from the application of JPLUME to the AVHRR image of May 13, 2000 is presented in Fig. 5. The pixels in red in the vicinity of Enschede (within the white circle) correspond to the plume caused by the massive explosion in the firework factory. These pixels can be easily separated in the CLD—NDVI space, therefore the plume can be detected and monitored.

The final pseudo-coloured image can be saved on the hard disk with the use of the Export sub-module. The Band Sequential technique is used to store the three bands (CLD, NDVI, 0) of the pseudo-coloured image in a single file which is named plume.nek. This file can be opened in any image processing software as a generic binary file by giving the following three parameters: (a) the number of rows ($H$); (b) the number of columns ($W$); (c) the number of bands (3).

4. Conclusions

Using satellite data to detect a plume caused by major industrial accidents relates to the plume’s characteristics (dispersion, optical thickness and temperature structure) and to the sensor’s spatial resolution as well. A new low spatial resolution satellite image analysis software JPLUME, has been developed, as a comprehensive and user-friendly tool for the automatic detection and
monitoring of plumes caused by major industrial accidents. JPLUME methodology is based on the development of a two-dimensional feature space image in order to discriminate pixels that contain plumes from those that correspond to clouds or to the underlying surface. This two-dimensional feature space is generated by combining the normalized ratios \((5 - 1)/(5 + 1)\) and \((2 - 1)/(2 + 1)\) of AVHRR channels 1, 2 and 5. The first normalized ratio (named CLD) takes into account the radiation received in the visible and thermal infrared. It is used for the discrimination of pixels that contain plumes from those that contain clouds and for masking cloudy pixels. The second normalized ratio (the well known NDVI) takes into account the radiation received in the visible and near infrared. It is used in order to discriminate pixels that contain plumes from those corresponding to the underlying surface, as well as for masking pixels that contain water bodies. JPLUME, coded in JAVA2 programming language, has the following attributes:

(a) effectiveness and automatic operation: JPLUME can be used for the near real time detection and monitoring of plumes since its methodology is exclusively based on the use of satellite imagery;
(b) reliability: JPLUME’s reliability is dependent on the spatial and temporal resolution of the satellite data used as inputs;
(c) scalability: JPLUME can be applied for the monitoring of an area covering a single industrial installation, as well as for the monitoring of areas covering half of Europe, depending on the satellite data availability;
(d) interoperability: JPLUME, as a JAVA developed application, is independent of the platform and the operating system used;
(e) user-friendliness: JPLUME has a window-based user interface. It carries menus, buttons, a flow chart of the application algorithm and an Image Viewer on which the input and output images as well as the intermediate products can be presented.

The functionality and the applicability of JPLUME were demonstrated with an illustrative example of an industrial accident in a firework factory in the town of Enschede, The Netherlands on May 13, 2000. The effectiveness and reliability of JPLUME may be improved if used in conjunction with other sources of information, regarding for example the fire caused by an industrial accident. Such information may be retrieved from AVHRR imagery with the application of the appropriate fire detection algorithms. Finally, it is worth mentioning that MVISR (Multichannel Visible and Infrared Scan Radiometer) data from the Chinese Fen Yung 1C and 1D polar orbiting satellites may be used in JPLUME, provided that calibration files are available.

References


Internet
