Distributed Geospatial Data Processing Functionality to Support Collaborative and Rapid Emergency Response

Dominik Brunner, *Student Member, IEEE*, Guido Lemoine, *Senior Member, IEEE*, Francois-Xavier Thoorens, and Lorenzo Bruzzone, *Senior Member, IEEE*

***Abstract*—This paper presents a novel approach to integrate the latest generation very high-resolution earth observation imagery into the operational workflow of geospatial information support for emergency response actions. The core concept behind this approach is the implementation of an image pyramid structure that allows each image tile to be addressed separately. We propose a novel way to collate geospatial feature data from distributed sources and integrate them in visualization and image processing. The system components enable rapid collaborative mapping, support for *in situ* data collection, customized on-demand image processing, and geospatial data queries and near instantaneous map visualization. We adapt functional software modules that are available in the public and open source domain. The approach is demonstrated with a test case in a rapid damage assessment scenario using very high-resolution optical satellite QuickBird and IKONOS imagery over Southern Lebanon from 2006.**

***Index Terms*—Collaborative mapping, feature extraction, geospatial queries, image processing, information management, on-demand processing, visualization, web services.**

I. Introduction

THE increased availability of very high-resolution (VHR) satellite imagery is prompting an expansion of operational Earth Observation (EO) applications to provide detailed infrastructure assessment and monitoring. In Europe, the Global Monitoring for Environment and Security (GMES) initiative aims to develop a service infrastructure for integrated access to space based data assets and *in situ* data networks. It has recently turned its focus to strengthening the security component of GMES, which includes amongst others, emergency response to natural and man-made disasters and conflict monitoring. This effort builds, to some extent, on the expertise developed in emergency response activities triggered by the International Charter on "Space and Major Disasters" [1]. "Reducing loss of life and property from natural and human-induced disasters" is recognized as one of the nine societal benefits by the Group on Earth Observations [2] which is the international coordinating effort to build a Global Earth Observation System of Systems (GEOSS). That system is to address thematic applications in areas such as emergency response, early warning, vulnerability reduction and preparedness.

The reported occurrence of natural disasters is on the rise [3], [4] which has led to increased public awareness of the impact of catastrophic events. The use of satellite imagery as a mapping tool for forecasting, monitoring and postevent assessment is growing due to the availability of a wide range of commercial earth observation imagery that can be publicly disseminated and also used for specialized geospatial analysis for supporting emergency response situations.

Recent advances in software development have significantly expanded the role of remote sensing imagery and geospatial features data as important and up-to-date information sources. This is primarily driven by the uptake of web mapping applications, virtual globe viewers, the pervasive use of GPS-based tracking and routing devices, and novel collaborative geo-tagging applications. Much of the relevant functionality is available as free or open source software modules or as web-hosted applications, which has led to the expansion of geospatial processing capabilities well beyond the traditional community of remote sensing and GIS experts. Furthermore, new public and commercial data supply models are evolving for high quality remote sensing data which will challenge the traditional supply mechanisms, in particular for near real time supply of postevent satellite data in emergency or conflict situations.

A. Background

Emergency response is generally understood as supporting the organized intervention of civil security entities after a catastrophic event either caused by a natural disaster or resulting from human conflict. Geospatial processing activities in support of emergency response range from the provision of relevant archived map products to dedicated data processing to provide thematic inputs into the different phases of emergency response, e.g., situation assessment, logistical planning, detailed damage assessment and postdisaster reconstruction. The dynamics of the typical emergency response cycle and the institutional structures that are in place to trigger geospatial support actions, especially in the case of natural disasters, are detailed in [5].

Manuscript received July 25, 2008; revised October 03, 2008 and December 01, 2008. First published March 16, 2009; current version published April 29, 2009.

D. Brunner is with the European Commission Joint Research Center, I-21027 Ispra (VA), Italy, and also with the Department of Information Engineering and Computer Science, University of Trento, 38100 Trento (TR), Italy (e-mail: dominik.brunner@jrc.it).

G. Lemoine and F.-X. Thoorens are with the European Commission Joint Research Center, I-21027 Ispra (VA), Italy (e-mail: guido.lemoine@jrc.it; francois-xavier.thoorens@jrc.it).

L. Bruzzone is with the Department of Information Engineering and Computer Science, University of Trento, 38100 Trento (TR), Italy (e-mail: lorenzo.bruzzone@ing.unitn.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2009.2015770

In [6], some of the operational aspects of integrating earth observation-derived data into the information flow for emergency response are highlighted. The authors propose technical solutions for automatic change detection, using the Indian Ocean tsunami 2006 disaster as a case study. Another example of the application of remote sensing data for rapid damage detection is described in [7] for the 2005 Bam earthquake in Iran. Many technical papers on EO data processing methodologies increasingly address themes that are relevant to emergency response e.g., [8]–[11] and [12].

Other technological initiatives in emergency response focus on early warning systems (e.g., [13]), communication and navigation solutions for first responders, and information technology support for coordination efforts.

The characteristics of EO imagery and geospatial data integration in emergency response are distinct from those in typical environmental applications. Emergency response actions tend to be localized (e.g., at local or regional, rather than national or continental scales), thematically specific (e.g., classification of individual urban structures, rather than generic land use classes), and have stringent timing requirements for the delivery of the relevant data layers.

A typical sequence of geospatial support activities when responding to a call for emergency response support is as follows.

1) In the prealert stage, early warning indicators may trigger the search for suitable archive EO data that could be used to establish the pre-event reference situation. This stage is relevant only for events of a probabilistic nature for which adequate early warning mechanisms are in place (e.g., tropical cyclones [13], forest fire risk [14], and flood forecasts [15]).
2) Either at a predetermined high-alert level or directly after the event, there is an immediate need to provide access to digital repositories of appropriate EO imagery and geospatial feature data at the required scale and accuracy to establish the reference situation. Data layers derived from these data sets need to be specific to the thematic needs of the response effort (i.e., populated area delineation, infrastructure mapping, vicinity analysis). At this stage, the ability to distribute the analysis workload amongst domain experts and image analysts and assemble the contributions from a large number of contributors is paramount.
3) Directly after the event, high-resolution (airborne or satellite) imagery is tasked to be acquired over the event site. The primary use of this data is for the assessment of the postevent situation compared to the reference situation. Automatic classification and change detection algorithms [8]–[11] are particularly relevant in this context, because they assist in the visual inspection and feature capturing stage to quantify the impact of the event. At this stage, early postevent collateral information may become available, e.g., the exact impact area, logistics of the relief effort and media reports, that will help guide the geospatial analysis effort.
4) After the initial geospatial analysis results are disseminated to emergency response actors, new queries may be formulated that require the data layers to be revisited or the geospatial analysis to be fine-tuned. Depending on the nature of the event, several situation updates may be necessary (e.g., forest fires, extended conflicts). In certain cases, the emergency response may be followed by a reconciliation stage, e.g., reconstruction, whereby progress reporting may be supported with further geospatial analysis.

The description provided above is generic and covers a broad range of emergency situations. Functional and technical implementation details and performance requirements are, however, specific to each particular event. The common requirements at all stages of the emergency response cycle are: 1) the collation of data resources from different archives and acquisition capabilities, 2) the need for customized rapid visualization for each of the actors in the emergency response community, and 3) the need to communicate, in near real-time, requests for geospatial processing, organize collaborative efforts, and inform decision makers with tailored output.

B. Scope of the Paper

This paper describes a system integration effort that leverages the functionalities of publicly available and open source components to enable collaborative and rapid processing of distributed geospatial data, including large high-resolution image coverages. We introduce a novel concept for the integration of privately held very large VHR images both in customized visualization environments and accessible to extensible image processing capacities via a web service mechanism. The system enables distributed access to geospatial feature sets, that can be collected as digitized feature sets from a community of contributors and exposed through web services. Geospatial feature data can be integrated into the distributed image processing capacity which is optionally backed up by a grid computing architecture to enhance processing speed. We demonstrate the use of the system in an emergency response context.

The remainder of the paper is organized in the following manner. Section II introduces the system architecture. In Section III, we discuss how we integrate very large image data sets providing the algorithm implementation details and illustrating the procedure with an example (Section III-E). In Section III-F, we describe how we can trigger image processing requests on the image data sets. We then highlight the feature capturing (Section IV) and visualization (Section V) capabilities of the system. In Section VI, we introduce the data sets we have used for testing the system. A discussion of the results is found in Section VII, and conclusions are presented in Section VIII with a summary of the key aspects of our system and outlining a number of issues we intend to address in future work.

II. System Architecture

The requirements to share data between teams suggests the need for a common distributed software platform based on a client-server model [16]. This architectural model allows communication between several distributed clients with one or more servers using network connections. The client initiates server requests and waits for and receives server replies. It is typically a graphical user interface through which the user can interact with the data sets and the server. The server responds to client requests by performing a triggered processing task and providing the result. The use of the client-server model is generic to many

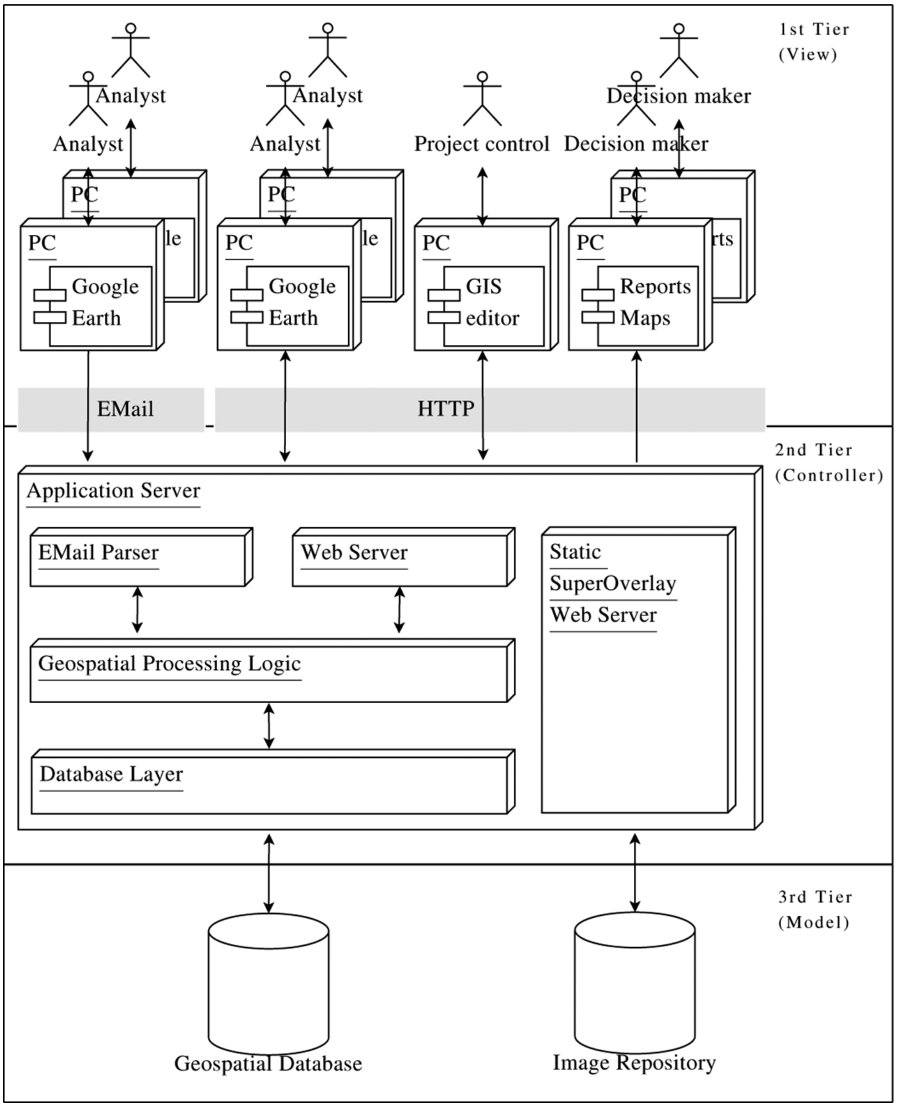

Fig. 1. UML overview of a multitier client-server architecture for collaborative feature capturing and visualization.

current web applications. Its use in web mapping applications is an alternative to desktop or locally networked GIS applications because it functions across the boundaries between diverse organizations, stimulates the use of standard formats and exchange protocols, and permits the distribution of geospatial functionalities to relevant user communities in a tailored fashion. Drawbacks of the client-server model are the need for increased efforts supporting transactional management, including authentication, and possibly limitations in client functionality. A full discussion on the pros and cons of web mapping platforms are given in [17].

The overall system architecture is illustrated in Fig. 1 using the Unified Modeling Language (UML) notation [18]. To manage the functional and computational complexity of geospatial data processing on the server, we use a three-tiered architecture [16] following the so-called Model-View-Controller design pattern for distributed systems [19]. The first tier is the client (the View) which captures user data and composes client requests, forwards these to the application server and visualizes the application server's response to the user. The second tier is the application server (the Controller), which handles all the business logic required to process a client request (e.g., selection, processing actions, response compilation). The third tier (the Model) communicates with the data stores.

A. The View

The View, which is the interface running on the client side, distinguishes three functional roles: 1) the geospatial analyst, 2) the project manager, and 3) the decision maker. The geospatial analyst inspects the satellite imagery and manually extracts features using the client interface. The analyst can either be a local expert familiar with the area under investigation, a domain expert specialized in urban areas, or a user who has been introduced to geo-spatial feature capturing using the client (see Section IV).

In our scenario, we use the Google Earth™ virtual globe viewer as the client platform for this role. Google Earth™ can be downloaded for free and runs on the Windows, Mac, and Linux operating systems and has a very large user base. Furthermore, Google Earth™ provides ready access to a global archive of high and very high-resolution imagery stored on the Google Earth™ server which is regularly updated, providing a background for reference situation mapping. In Section III, we demonstrate how the Google Earth™ background imagery can be complemented with access to full resolution image products held in private archives.

The project manager is responsible for managing, controlling and supporting the overall collaborative analysis effort. Initially, she prepares the system by creating the project structure, uploading relevant archive data for the region of interest, and processing the satellite images for display in Google Earth™. Based on requests and instructions received from the emergency response coordinator, the project manager distributes the work effort amongst collaborators, for instance, by assigning analysis tasks by theme or sub-region of interest. At this stage, the use of ancillary and collateral information to stratify the region of interest into priority zones is very important. The project manager is the main contact point for technical problems and collection of feedback on the use of the system. She uses the Google Earth™ client or other specialized GIS clients for data editing, quality control and management. In practice the project manager, or her technical team, will also have direct access to the various components of the server infrastructure to intervene when appropriate.

The decision maker roles are taken up by the emergency response coordinators who incorporate the geospatial analysis outputs and steer the intervention effort. They access the system in read-only mode receiving customized reports and maps that are compiled in real time from the available data via the web server interface of the system.

We have implemented a simple role-based authentication mechanism to ensure that the user of the system is only able to accomplish the actions in the system that his role prescribes. The analyst is allowed to create new vector data and update or delete data he has previously created. The project controller is the system administrator and is able to create, change and delete globally any kind of data (i.e., raster and vector data). The decision maker is only granted read access to generate customized situation reports and map products. All data submissions are tagged with the user ID and time stamped allowing changes to be tracked by user over time.

The client-server communication is synchronized via the Hypertext Transfer Protocol (HTTP). The refresh mechanism in Google Earth™ ensures that any data submitted by an analyst is distributed to other users who are connected to the system. In cases where there is no reliable Internet connection, which may be the case in developing countries or areas that are affected by a catastrophic event, client data can alternatively be forwarded by email to a functional mailbox. Google Earth™ can also work with locally stored feature data sets in offline mode.

Note that the choice of using Google Earth™ as the client places a number of constraints on data formats and standards. Imagery integrated into Google Earth™ must be reprojected to Plate-Carrée (or geographic) projection using the WGS84 ellipsoid. The mandatory geospatial feature data format for Google Earth™ is the Keyhole Markup Language (KML) [20]. The geospatial data abstraction library (GDAL) [21] is particularly useful for converting other feature data formats (e.g., ESRI shape files) to KML and reprojecting imagery as required.

B. The Controller

The second tier consists of the application server which is made up of five components: a web server, an email parser, the geospatial processing logic, a database abstraction layer, and a static `SuperOverlay` web server. The web server is an Apache TOMCAT instance and receives the HTTP requests from the clients. The requests are either for data access or requests for data submission. An email parser makes asynchronous uploading to the system possible via a functional mailbox. The web server and the email parser delegate the requests to the geospatial processing logic component, which provides functionality for querying and manipulating geospatial data (create, read, update and delete). This component communicates indirectly via the database abstraction layer with the geospatial feature database. The static `SuperOverlay` web server serves static imagery as `SuperOverlays` (see Section III) accessing the image repository directly and, thus, does not need to use the database abstraction layer. Our application server is implemented in Java and runs on a Linux platform. However, given the inherent platform independence of Java, the server can be deployed on any system for which a Java run-time environment is available.

C. The Model

The geospatial database manages geographical feature data and their attributes. For organizing the geospatial information the object-relational PostgreSQL database management system (DBMS), with the PostGIS extension is used. It is available for Windows and Linux platforms. PostGIS allows storage of geographical objects and includes support for spatial indexing and functionality for the analysis and processing of geographical objects. The database layer in the controller, however, abstracts the implementation specific details of the geospatial database so that it is possible to change the DBMS quite easily.

The proposed separation between the geospatial feature data server and image data server is not obligatory. In fact, one web server would be enough to handle both feature and image data requests from the clients. Our set-up does not include direct interaction between the two server parts. Such interaction is logically separated in the controller, which enhances portability and scaling of individual server components. This also makes it easier to integrate other data access protocols [e.g., the File Transfer Protocol (FTP)] to serve specific purposes.

III. Integration of Very Large Image Data Sets

The distribution of large image coverages among project collaborators, either for image processing or for the visualization of the processed data is a well known issue. This is particularly evident in operational scenarios such as emergency response, when large areas are analyzed at large scales for impact assessment, usually within stringent temporal constraints. The OpenGIS Consortium (OGC) [22] proposes the use of the Web Coverage Service (WCS) [23] and the Web Map Service (WMS)

[24] image serving protocols respectively, for image sub-selection and recomposition. Both services can provide reprojection functionality. These services, however, tend to be computationally expensive and unable to handle a large user base in real time, i.e., the typical scenario in collaborative rapid mapping.

For the rapid visualization of large terrain data sets, [25] proposes the use of image pyramids (also known as MIP maps), i.e., a multiresolution stacked representation of the image. Each level of the pyramid is usually a 2 × 2 up-sampled version of the underlying, higher resolution level. The intrinsic relation between level of detail, tiling coordinates and the relative address in the stored image file allows for very fast access to the raw data. Image pyramids may either be stored inside a (propriety) binary image format (e.g., GeoTIFF, ENVI, ERDAS) or stored as individual files in a physical tile structure on disk. The latter is used in combination with a unique fixed tile coding convention by virtual globe server software such as Google Earth™, Microsoft Virtual Earth™ and NASA's World Wind in order to efficiently visualize their respective global remote sensing archives.

To display privately owned static image pyramids within the Google Earth™ virtual globe viewer, the `SuperOverlay` [26] element was introduced in version 2.1 of KML. KML has recently become an OGC implementation standard (the current version is 2.2). A `SuperOverlay` is a hierarchical tree-like structure of `NetworkLinks` of regionalized `GroundOverlays`, which is the standard KML element to display small georeference image tiles within Google Earth™. At the top of the tree, a `Region` element defines the geographical extents of the image composing the `SuperOverlay`. This `Region` loads the top level (i.e., lowest resolution) up-sampled image as a `GroundOverlay`. The top level KML file contains four `NetworkLinks` to the four quad-regions of the next tiling level. The "level of detail" (`minLodPixels`, `maxLodPixels` KML elements) controls the visibility of each tile level in such a way that the higher resolution levels become visible when the user zooms in closer. This mechanism ensures that only those tiles are loaded that are needed to fill the zoom window of the viewer. Google Earth™ suggests the use of tiles of 256 × 256 for optimum performance, but the user is free to decide the actual tile size. A key difference between static `SuperOverlays` and standard MIP maps is that each image tile is individually addressable and loadable as a single file. In other words, it can be served via the static web server as a Uniform Resource Locator (URL) resource. An added advantage of static `SuperOverlays` is that they can be provided on separate media for offline integration into Google Earth™. For mobile platforms having limited disk space, caching parts of a `SuperOverlay`, e.g., to support *in situ* data collection, requires a simple synchronization operation.

Several software implementations for generating a `SuperOverlay` are available (e.g., as part of the GDAL toolkit). We have implemented a Java application for batch generation of `SuperOverlays`. Their creation can be broken down in the following processing steps [27].

1: Reproject the image to Plate-Carrée projection and extend row and column dimensions to a quad-multiple of 256.
2: Tile the extended image into tiles of 256 × 256.
3: Create up-sampled tiles for each of the lower resolution pyramid level.
4: Create the `SuperOverlay` KML tree structure.

The actual processing steps are described in the following subsections.

A. Image Reprojection and Extension

Typically, images used in a project or local scope are in a local projection system (e.g., UTM) and retain the original spatial and spectral resolution of the sensor. Visualization in Google Earth™ requires the data to be reprojected to Plate-Carrée projection and spectrally resampled to 8-bit. The spatial resolution can normally be retained, but some re-sampling is performed in the reprojection process.

Reprojection often results in imagery that is rotated with respect to the North-South orientation, with zero-filled boundary areas. In order to avoid complex re-sampling at the lower resolution pyramid levels, it is useful to extend the reprojected image to an image size that is a quad-multiple of 256 (i.e., 2, 4, 8, 16, ... times 256 in both the row and column dimensions). For instance, an image with a width and height of 800 × 800 is extended to 1024 × 1024, corresponding to quad-level 2, resulting in 4 × 4 tiles. Our implementation centers the original image inside the extended image frame.

B. Image Tiling

The tiling operation is only performed for tiles that are not completely in the background, i.e., either in the zero-filled boundary area of the projected image or the zero-filled quad-multiple extension. Before writing each tile to disk, all zero-filled pixels are made transparent so that they do not hide the background imagery when visualized in Google Earth™. Tiles are written as Portable Network Graphics (PNG) formatted files that provide lossless compression and retain the transparency properties of the image tiles.

C. Creation of Up-Sampled Image Pyramid Layers

The quad-level sizing of the reprojected image is particularly useful at this step, because it guarantees a 1 to 4 (2 × 2) relationship between tiles at each higher quad-level.[1] That is, each up-sampled tile is the result of a 2 × 2 re-sampling of exactly 4 higher resolution tiles. If a higher resolution tile does not exist, e.g., because it lies in the zero-filled background, the relevant quadrant in the up-sampled tile remains zero-filled, which is made transparent before the tile is written. A side effect of this operation is that all zero-valued pixels become transparent.

We iterate the process until we reach the top quad-level, which results in a single 256 × 256 tile. In the previous example, the 1024 × 1024 extended original image has three quad-levels 2, 1 and 0 with 16, 4, 1 tiles respectively.

D. Creation of the `SuperOverlay` KML Tree Structure

Each tile at each of the quad levels is a georeferenced image in Plate-Carrée projection. The `SuperOverlay` KML tree struc-

[1] The quad-levels are numbered according to their hierarchy in the image pyramid. The "top of the pyramid" corresponds to quad-level 0 and has 1 tile. Each lower pyramid level has an increasing quad-level index q, each with 2^{2q} tiles.

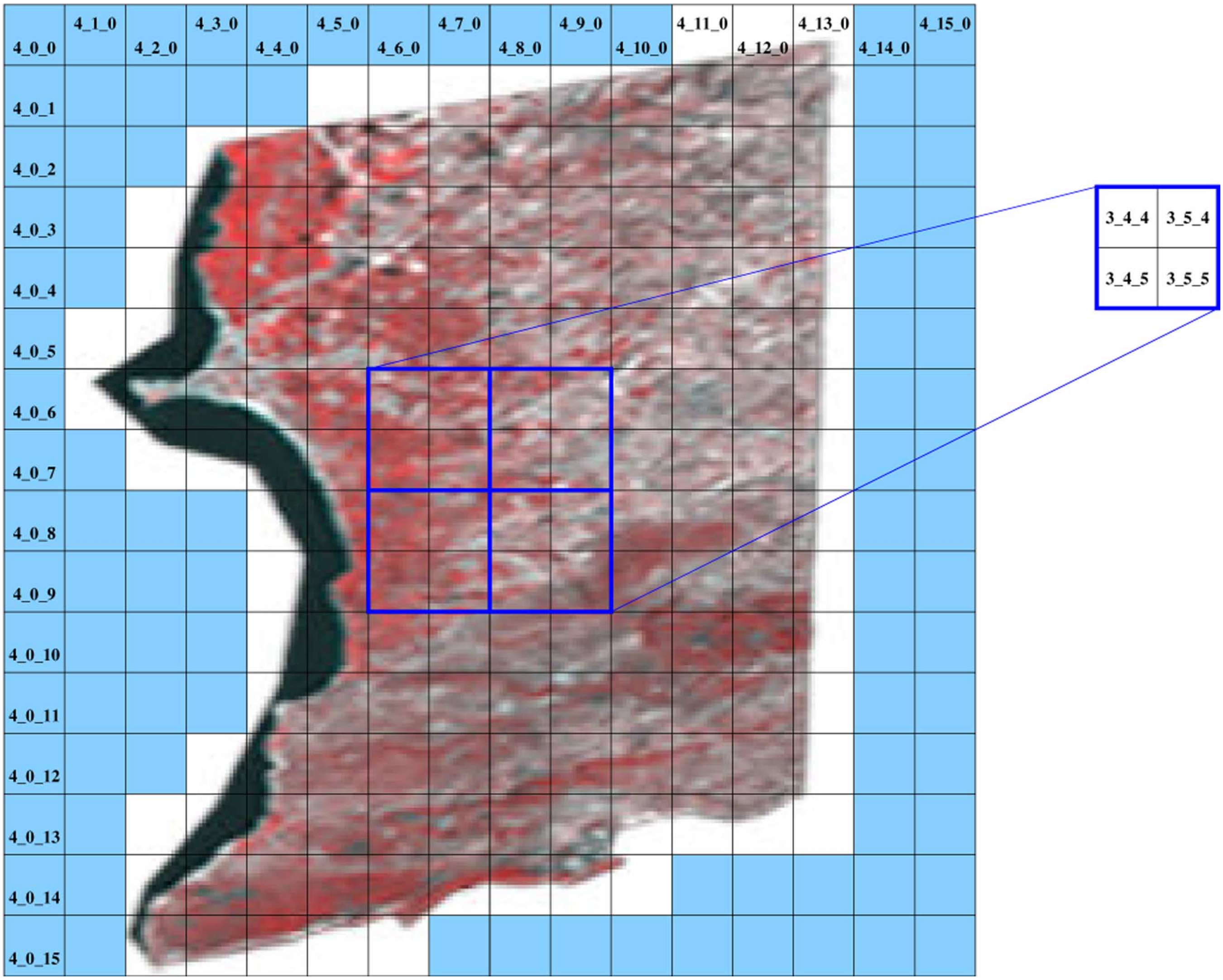

Fig. 2. Schematic overview of the tiling process for the sample image. The input image in Plate-Carrée projection is first extended to a grid at quad level 4, centered on the image. The grid is then tiled into 16 × 16 tiles. Tiles exclusively in the zero-filled background values (light blue in the figure) are not saved to disk. Tiling at each of the higher quad levels 3, 2, 1, and 0 results from composing each subsequent level from the up-sampled four tiles at the preceding level. Satellite image distributed by EURIMAGE S.p.A. © DIGITALGLOBE 2006.

ture persists both the geolocation of each tile and the hierarchical pyramid tile structure in KML constructs. For each of the lowest quad level (highest resolution) tiles, the KML document defines the geographical `Region` covered by the tile defined in a `GroundOverlay` element. Since these tiles are the leaf nodes of the tree, they do not link to others. Leaf node tile KML files are only created for tiles that are already on disk, i.e., created in the previous steps.

For each of the higher quad levels, the KML document is similar to the one for the leaf node tiles, except that each has in addition up to four `NetworkLinks` giving the `Region` definition and link to the underlying tile KML descriptors.

E. A `SuperOverlay` Example

To illustrate the process with a practical example, we show the typical output for a very high-resolution (2.4 meter pixel size) multispectral QuickBird image. The image is ortho-rectified to the relevant UTM projection, masked to remove a significant section that covers the sea, and reprojected to Plate-Carrée projection resulting in an image of 3378 × 3875 pixels.

In the first step of the algorithm, the image is extended to 4096 × 4096 pixels (quad level 4) and then tiled into 16 × 16 tiles of 256 × 256 pixels. This process is illustrated in Fig. 2. The tile naming format is QQQ_NNN_MMM, where QQQ is the quad level, NNN is the column index and MMM is the row index in the tiled grid. Three digits are used in this naming scheme so that a maximum tile index of 512 (i.e., quad level 9) is possible. This maximum tile index corresponds to a grid size of 131072 × 131072, equivalent to an upper limit of 48 Gb for the input image. This limitation can be relaxed by using more than three digits for the tile index. An alternative for processing images which are larger than 48 Gb is to split up the image into blocks of not more than 48 Gb and `SuperOverlay` each block separately. The results can then be linked using KMLs `NetworkLinks`. However, the handling of files larger than 12 Gb becomes difficult in practical terms because it exceeds the single-band 4 Gb limits of the widely used GeoTIFF image

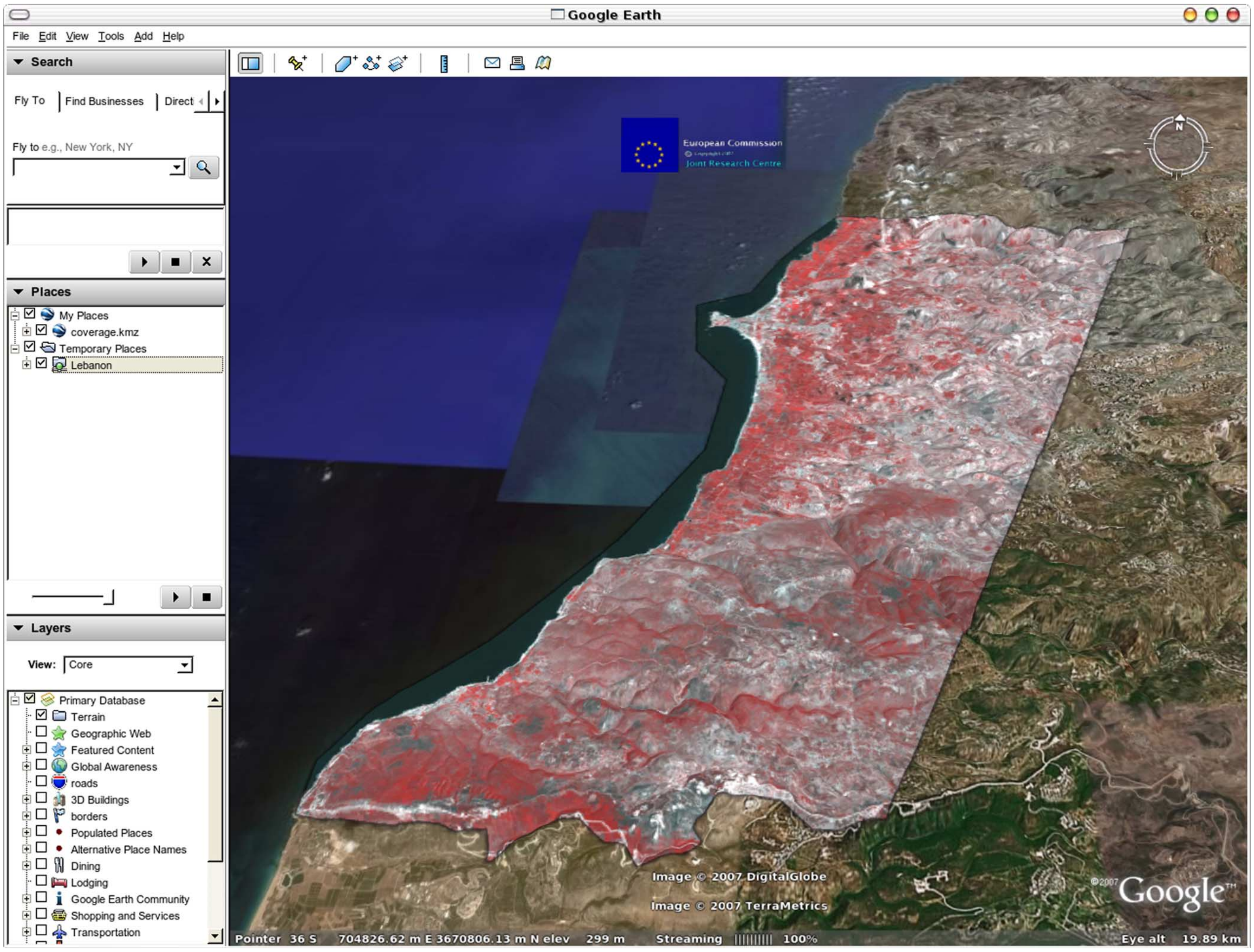

Fig. 3. Screenshot of the Google Earth™ viewer after opening the `SuperOverlay`'s top level KML file. Satellite image distributed by EURIMAGE S.p.A. © DIGITALGLOBE 2006.

format, which corresponds to an upper quad level of 256 or a grid size of 65536 × 65536. Most single-scene satellite image data sets are well below the 12 Gb limit. For very large data sets (e.g., image mosaics), raw or proprietary input file formats, which do not have a size limitation, may be used. For performance reasons `SuperOverlays` are currently pregenerated from the imagery and stored with their full structure in the image repository.

A screenshot of the integration of the large QuickBird test image into Google Earth™ using `SuperOverlays` is shown in Fig. 3. Using the Google Earth™ navigation widgets, we are able to zoom into each of the subsequent quad levels seamlessly, visualize the data in the 3-D landscape mode and overlay geographical features from other KML sources. Of special interest is the transparency slider that is associated with the `SuperOverlay`. We can control the transparency of the added `SuperOverlay` from fully opaque to fully transparent. If more than one `SuperOverlay` is loaded, we can use this control to look at each visualization separately. This is very useful in manual change detection scenarios, e.g., comparing "before" and "after" states in multitemporal series, which we can then capture with KML annotations.

`SuperOverlays` are also suitable for the visualization of multitemporal imagery. The KML standard provides the `TimeStamp` element to allow rendering of the KML `GroundOverlay` at specific time intervals. Animation of time series can be controlled via Google Earth™'s time navigation widget.

F. `SuperOverlays` *for Image Processing*

The above example refers to using the `SuperOverlay` in a visualization environment, such as the Google Earth™ client. However, `SuperOverlays` can also be used to support image processing tasks. Image processing algorithms that run on byte-formatted input, can integrate the `SuperOverlay` directly. If radiometric resolution requires the image to be kept in a different number format (i.e., float or 16-bit integer format) the `SuperOverlay` algorithms can be adapted to produce image tiles in the TIFF format.

`SuperOverlays` are especially suited to dedicated and CPU-intensive image processing tasks (e.g., image segmentation and/or classification) to be performed on a predetermined region of interest. The pyramid tile structure can be used directly for multiscale image analysis and with feature extraction algorithms. This structure is also suitable for integrating image processing and pattern recognition algorithms specially designed for managing large images in split-based approaches [6]. In such cases, the tile structure can be automatically associated with the image split necessary for multilevel processing. The `SuperOverlay` also implicitly supports the use of multisource and multitemporal data sets since all are georeferenced to the same projection. Outputs generated by the image processing tasks, whether applied to the entire image pyramid or specific layers or tiles, can be made available as `SuperOverlay` structures for integration into the project data collection.

1) Integrating Image Processing Functionality: Our distributed system is able to provide dedicated image processing services that can be incorporated as third party web services. The relevant `SuperOverlay` URL and region of interest (ROI) feature (typically a polygon) can be forwarded, together with relevant processing parameters, to the published web service address. The public web service interface is separate from the privately hosted implementation of the image processing functionality. This is an especially attractive feature for specialized processing algorithms that are not available in off-the-shelf image processing software packages, for experimental routines, or where code protection is essential.

The workflow for integrating image processing into our system is shown in Fig. 4. The process is divided into two systems: a public system that contains the data to be processed as a `SuperOverlay` and a private system that encapsulates the image processing routines. The project controller or expert analyst selects in a first step the ROI, which might be a predefined area or the result of a geospatial query (e.g., the automatic detection of built-up areas in optical VHR imagery within a 1-km buffer area around a river flood plain). In a second step, the imagery is selected, which is addressed by the URL of the `SuperOverlay`. In a third step, the image processing request is triggered by forwarding the URL and the ROI to the public interface of the web service hosting the private algorithm implementation. Depending on the algorithm and its implementation, it may be necessary to compose the tiles of the `SuperOverlay` in a fourth step to a single or several larger images before the algorithm can be applied. In such cases, the resulting imagery has to be tiled back in a fifth step to the `SuperOverlay` structure so that it can be made accessible to the project partners through the Google Earth™ client via a `NetworkLink` posted to the server (sixth step).

2) Combining Image Processing and Grid Computing: In order to decrease the processing time of the image processing server, the algorithm designer may incorporate grid computing facilities as shown in [28]. Grid computing [29] is a hardware and software infrastructure that provides high-performance computational capabilities by combining the processing capacities of distributed CPUs to handle large processing tasks. The individually addressable tiles in the `SuperOverlay` structure are uniquely qualified to be processed in a grid environment, especially when image processing tasks can be easily parallelized as tile operations. Fig. 5 illustrates the combination of `SuperOverlays` and grid-enabled image processing. This figure is an extension of the right hand side of Fig. 4. The key difference is that the image processing server, which receives the processing requests, acts as a task broker to subdivide the workload between available grid computing nodes based on the analysis of the ROI and `SuperOverlay` descriptors received in the request. Each grid node acts in the same way as a stand-alone instance processing the tile subset identified by the broker and storing the output in the private repository. The task broker monitors the processing nodes and publishes the address of the repository upon completion in exactly the same manner as the stand-alone process described in Fig. 4. Note that the `SuperOverlay` integration enables the distribution of image tile identifiers, rather than image tiles themselves, leading to a significant reduction in intra-node communication. Each of the processing nodes only loads the relevant tile(s) for the identifiers passed to it by the broker. This mechanism allows for sophisticated processing load balancing, leading to a significant increase in image processing throughput.

IV. Collaborative Feature Capturing

Our concept of "collaborative feature capturing" is some form of organized collation of geospatial feature data sets to support a given mapping task. The collaborative part assumes that more than a single person contributes to the collection, more or less simultaneous, but at different (machine) locations. The feature sets to be captured may be existing KML layers, those converted from other formats, or freshly digitized features in Google Earth™.

In our system, we exploit the forwarding mechanism found in Google Earth™ to collect the captured feature sets via the application server, into the spatial data repository. Capturing tasks can be organized either by geographical area, outlined as a polygon defining the region of interest, by thematic layer (e.g., roads, buildings, etc.), or by a combination of both. Instructions to individual contributors can, in fact, be communicated using the system set-up. We have implemented a basic authentication mechanism for contributors, based on preregistered login and password credentials, which are linked to a project identifier. Credentials are propagated and checked each time new features are submitted. Other checks can be incorporated in the application server logic, e.g., to limit contributions to the predefined region of interest or thematic layer for that contributor.

In order to be able to collect captured data in a systematic way, we use predefined class hierarchies. A hierarchy includes a tree-like class definition and sub class definition up to the third level. For instance, a captured highway may be stored in the Road/Paved/Highway node of the hierarchy tree. In Google Earth™ the class hierarchy takes the form of a set of folders and sub-folders, much like a directory structure in a file manager. The class hierarchy is stored for each of the submitted features. The possibility to link a project to a predefined class hierarchy allows for a flexible mechanism to address specific capturing tasks that need to conform to a user-defined map legend (e.g., topographic maps). Furthermore, the persisted class hierarchy or parts thereof can be transformed into hierarchies that are used in other mapping domains (e.g., tactical maps).

The feature-capturing task supports feature updating and deletion. The spatial data repository uses feature versioning

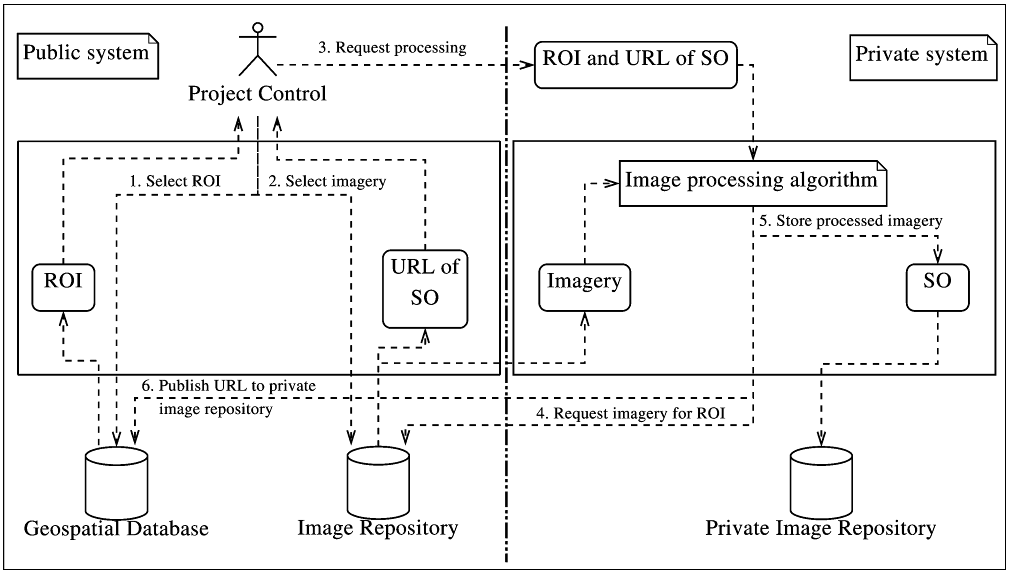

Fig. 4. Schematic overview of the integration of image processing into the system. This process is divided into a public and a private system. The data to be processed are stored as `SuperOverlay` (SO) in the public image repository. The image processing routine is triggered by specifying the SO and the region of interest (ROI) to be processed by an algorithm that is provided by a partner on their private system. The results are stored in the private image repository which is accessible by the public system.

to keep track of changes to individual features. Versioning is an important mechanism in the support of feature updating, rollbacks, replays and release tagging. It protects the process from malfunctioning or from human error that could lead to unwanted data removal.

A feature can include a number of attributes that describe relevant properties (e.g., the road surface type and width). The system also allows the inclusion of additional multimedia information such as photographs, videos, or audio files. This is particularly interesting for integration of ground observations. Links to data held on other systems can be forwarded as `NetworkLinks` that relate to a geographic feature. The latter is a simple way of publishing the availability of a new image `SuperOverlay` to the project team.

V. Data Visualization

All geospatial feature data are held in the spatial repository. A separate system component provides controlled access to the data stored in the system. The open source module GeoServer, which is an OGC-compliant web map server, is used for visualization. GeoServer supports both the WMS and Web Feature Service (WFS) [30] protocol standards that are relevant in our context. GeoServer exposes geospatial features from the spatial repository in a number of standard formats for integration into map clients, including KML. Feature querying and conditional styling capabilities [31] that are part of the OGC standards can present the GeoServer outputs in predefined class hierarchies and in map presentations that can be tailored to the required end-use. Data access and customization is based on predefined users and roles specified in the data repository.

Using the built-in refresh mechanisms in the Google Earth™ client, the user is able to display an instantaneous view of the data holdings even while the feature capturing process is on-going. Since all submitted feature data is time-stamped, the Google Earth™ time animation functionality can be deployed to display the collaborative effort chronologically. This is particularly useful for project managers, who must monitor work progress, identify processing bottlenecks and provide progress reports and intermediate map outputs to help inform the emergency response decision makers.

The use of the PostgreSQL data base extended with PostGIS permits data access to geospatial clients that can perform enhanced processing beyond the feature capturing and visualization capabilities of Google Earth™. This may include data integrity checking, topology creation, spatial querying, and data conversion tasks that are performed by a limited number of project actors. This functionally extended Google Earth™ client interface is particularly useful because of its widespread exposure to the community of potential contributors.

VI. Test Case Description

The system presented in the previous sections is generic and can be deployed within several application contexts that require collaborative and rapid geospatial analysis integration. It was specifically applied to a damage assessment scenario during the Lebanon crisis which was an international armed conflict between Lebanese and Israeli forces in July of 2006. Geospatial data were collected to assess the situation in the conflict areas in order to estimate EU support for funding of reconstruction and humanitarian aid in South Lebanon after the conflict. The scenario is particularly demanding for four reasons: 1) strict time constraints were imposed because the situation assessment had to be produced within two weeks; 2) very large data volumes had to be processed because the damage assessment analysis required very high-resolution imagery for the entire Southern Lebanon area; 3) the accuracy of the ortho-rectified imagery had to be high for the comparative analysis; and 4) the distribution of the workload between two geographically dispersed image analyst teams.

Several mosaicked IKONOS scenes from 2005 covering the entire South Lebanon area (covering approximately 30 km north-south and 40 km east-west centered around 33.19° N, 35.36° E), with 4-m spatial resolution in four multispectral channels and 1 meter resolution in the panchromatic channel were used as pre-event base data. The preprocessing of the image, with dimensions of 56260 × 41417 pixels and a size of 7.1 Gb, comprised of pan-sharpening [32], ortho-rectification [33] and radiometric scaling to 8 bits. The preprocessed image was `SuperOverlaid` using the algorithm outlined in Section III resulting in 30016 tiles with a total size of 2.3 Gb. The total processing time to produce the `SuperOverlay` on a standard workstation (dual Xeon 2.8-GHz CPU, 3-Gb RAM, Ubuntu 8.04) was slightly more than 2 h (125 min and 45 s).

The postevent image was made up of a mosaicked QuickBird scene, which was acquired after the conflict ended in August of 2006. This image mosaic has a panchromatic spatial resolution of 0.6 meter and a spatial resolution of 2.4 m for the three multispectral channels. Preprocessing was the same as for the pre-event IKONOS mosaic. Applying the `SuperOverlay` algorithm on this 10.6-Gb image whose dimensions were 73136 × 51821 pixels resulted in 49188 tiles and 4 Gb in size. Total processing time was 176 minutes and 44 s.

Both pre- and postevent `SuperOverlays` were uploaded to the web server by the project manager. The geospatial feature database was initialized with ancillary data that were made available through the Lebanese Council for Development and Reconstruction (CDR). Data sets included cadastral limits at 1:200 000 map scale, major populated places, roads and airports at 1:100 000 map scale, and rivers at 1:50 000 scale, which were converted and reprojected from ESRI shape files in UTM33N to Plate-Carrée projection.

As control data, we had a large set of point features that were digitized during a standard photo-interpretation exercise by our colleagues and our partners at the European Satellite Center (EUSC) immediately after the crisis, at which time our system was not yet available. Point features outlined partially or fully damaged structures based on the visual interpretation of pre- and postevent imagery. Furthermore, field missions to the area in the aftermath of the crisis had yielded a large set of GPS-tagged photos that were very useful in assessing the quality of the visual interpretation.

VII. Results and Discussion

The collaborative effort is triggered when the pre-event imagery and ancillary feature data sets are available in the system. First, the analyst familiarizes himself with the area under investigation and checks whether the uploaded vector data are correctly geo-located based on the reference VHR image coverage. This is important because the data with which the system is initialized usually has its origins from different sources with varying spatial resolution, projection, and quality. If the precision of the initial data is not sufficient, it can be postprocessed within the system. For instance, a western offset was found in the 1:100 000 scale road data set with respect to the georeferenced IKONOS pre-event mosaic. The analysts digitized a set of ground control points across the region of interest to assess the overall shift and submitted these corrections to the system. This permitted the project manager to calculate the mean shift of $82.3^{\pm}0.8$ m and apply a translation to the road data set correcting for the shift. After a refresh request of the road layer, all users have access to the corrected road set.

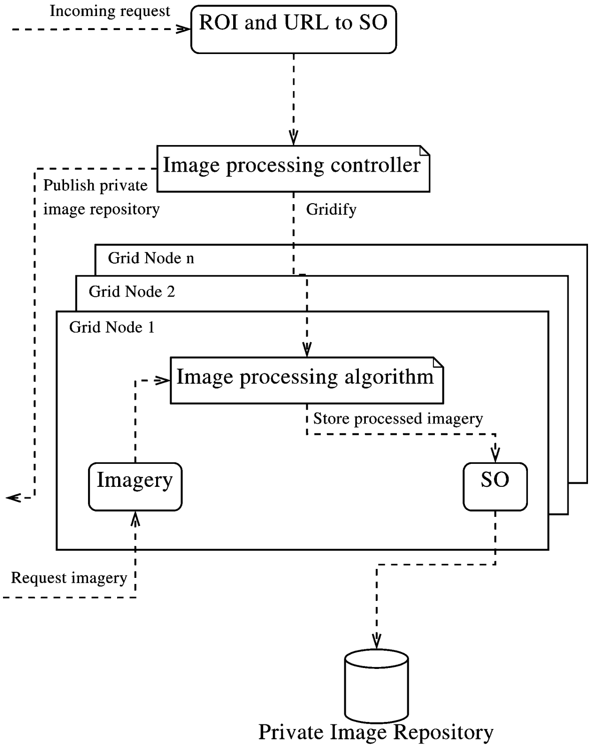

Fig. 5. `SuperOverlay` integrated into a grid-enabled image processing task.

The primary task of the project manager, at this stage, is to divide the region of interest into manageable areas for eventual visual inspection and digitization. Since the damage assessment focuses on populated areas (humanitarian situation assessment) and infrastructure (logistics, damage value estimates) the priority is focussed on locating potentially affected settlements. The ancillary feature data on populated places was too coarse to be useful beyond queuing. An in-house implemented web service processing algorithm for the calculation of a built-up presence index [34] on the tiles of the pre-event `SuperOverlay` was used instead. The algorithm is based on the fuzzy rule based composition of anisotropic textural measures derived from the gray level co-occurrence matrix of the byte-scaled panchromatic image channel and highlights built-up areas. The results of this image processing algorithm is accessible as a `SuperOverlay` (see Fig. 6) to the analysts, who can then outline the populated areas to be analyzed at the damage assessment stage. Groups of outlined areas were assigned to individual analysts to avoid duplication of digitization efforts (assignment is stored as an attribute to each polygon outline). Note that at this point media reports related to the conflict can be used to further focus on the areas most impacted by the conflict and store these as flagging features in the project data store.

For logistical support of the reconstruction effort road networks and specifically crossroads and bridges, are strategically important targets, which had a higher priority during the early assessment stage. One analyst team was tasked to digitize the road network, completing, as much as possible, the 1:100 000

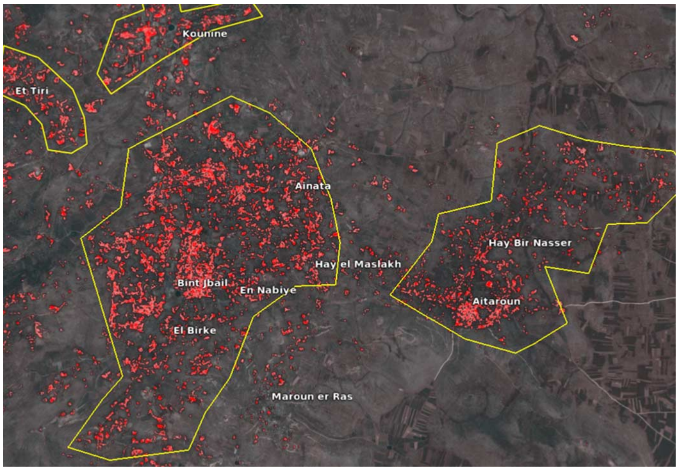

Fig. 6. Example of the output of the built-up presence index algorithm applied to the pre-event IKONOS `SuperOverlay`. The pre-event image in the background is overlaid with the output, which depicts in the red channel the density of the built-up structure. Non-built-up areas are made transparent so that they do not obscure the background. Settlements are outlined as yellow polygons, using the built-up presence index as reference. The satellite image is © Space Imaging International, Ltd., 2005, distributed by European Space Imaging GmbH.

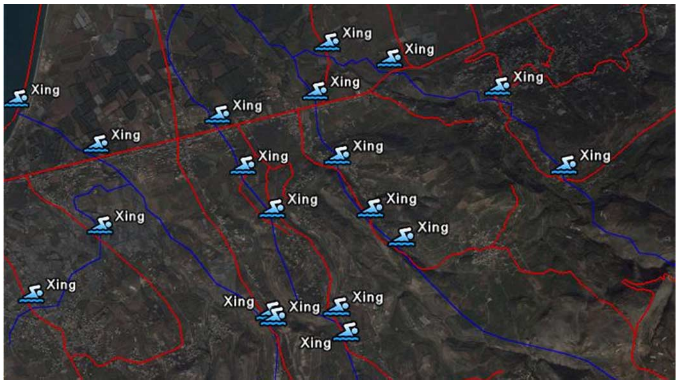

Fig. 7. Example of automatic derived road-water crossings. The geospatial intersection function was applied to the road (red) and river (blue) networks resulting in the highlighted crossings. Satellite image is © Space Imaging International, Ltd., 2005, distributed by European Space Imaging GmbH.

scale feature set that includes only a nonexhaustive set of highways and primary roads. Another team digitized water ways, which were mostly irrigation channels and dry riverbeds in the South Lebanon scene (only the Litani river was available in the ancillary feature set). Note that automatic or semi-automatic algorithms for feature extraction can be applied at this stage to derive relevant layers in support of the digitization tasks.

The project manager can trigger spatial queries on the road and river network data sets to locate the relevant crossings and store these as point features, as shown in Fig. 7. Each point feature is assessed by the analysts at the damage assessment stage. Furthermore, the point features may be used as seed points for a change detection algorithm which automatically assesses whether the region around those points has changed.

The actual damage assessment phase started as soon as the postevent QuickBird mosaic image was integrated as a `SuperOverlay`. The digitization proceeded according to a predefined class hierarchy as discussed in Section IV. Fig. 8 shows the result of the digitization of a portion of the Bent Jbail settlement (33.119° N, 35.435° E) which was in a heavily damaged area of the conflict zone. Digitized paved and unpaved roads are shown with red and brown lines, respectively. Moderately damaged buildings are outlined with light blue polygons, while completely destroyed buildings are highlighted with dark blue polygons. Impact craters causing damage to the terrain rather than infrastructure are indicated by yellow polygons. In the "Places" section in the left part of Fig. 8, the class hierarchy is shown. The class hierarchy was decided in a rather ad-hoc manner, as classification standards for damage assessment are not yet available. Future application of our system in similar damage assessment efforts would benefit from standardization efforts at both the data capturing and map styling stages.

Ground truth data collected during the assessment phase can be uploaded to the system. In particular, GPS-referenced reports and field photography are very useful to the analysts. For example, damaged buildings may be annotated with the corresponding images taken in the field with a digital camera. The Google Earth™ client pops up the relevant photo once the building polygon is selected, as shown in Fig. 8. In an armed conflict, ground truth collection is a hazardous task and Internet connectivity is erratic at best. Our ground data was collected after the conflict and was uploaded in an offline fashion. Various community sites on the Internet provided detailed geo-located photographs that were taken during and immediately after the conflict. We were able to correlate these photographs with visible damage in the QuickBird scenes, demonstrating the potential of this methodology, especially in less challenging emergency scenarios. It is also possible to deploy our system to identify objects or areas of interest for *in situ* inspections and communicate these to local field teams.

All activities related to the creation, modification and deletion of data in the system were timestamped and associated with a user. This allowed us to monitor the system and retrieve information on the progress of digitization tasks. This was not only important for the identification of bottlenecks during the digitization phase, but also for postanalysis of the rapid mapping project to draw pertinent conclusions in order to improve collaborative feature capturing for the next crisis response.

VIII. Conclusion

In this paper, three fundamental issues were addressed that, in our opinion, hinder the wide-spread take up of earth observation information in operational applications. Firstly, we showed how very large image coverages can be made accessible to a geographically distributed audience by implementing the `SuperOverlay` mechanism available within Google Earth™. Secondly, it was shown how the use of `SuperOverlays` can be extended to accommodate novel mechanisms for integrating dedicated image processing tasks. The image processing tasks can optionally be supported by grid computing resources. Finally, we demonstrated that access to distributed geospatial feature data in typical workflows that require an integrated analysis of image and feature information in support to decision making can be implemented.

We have demonstrated the proposed system use in the context of an emergency response following the armed conflict in

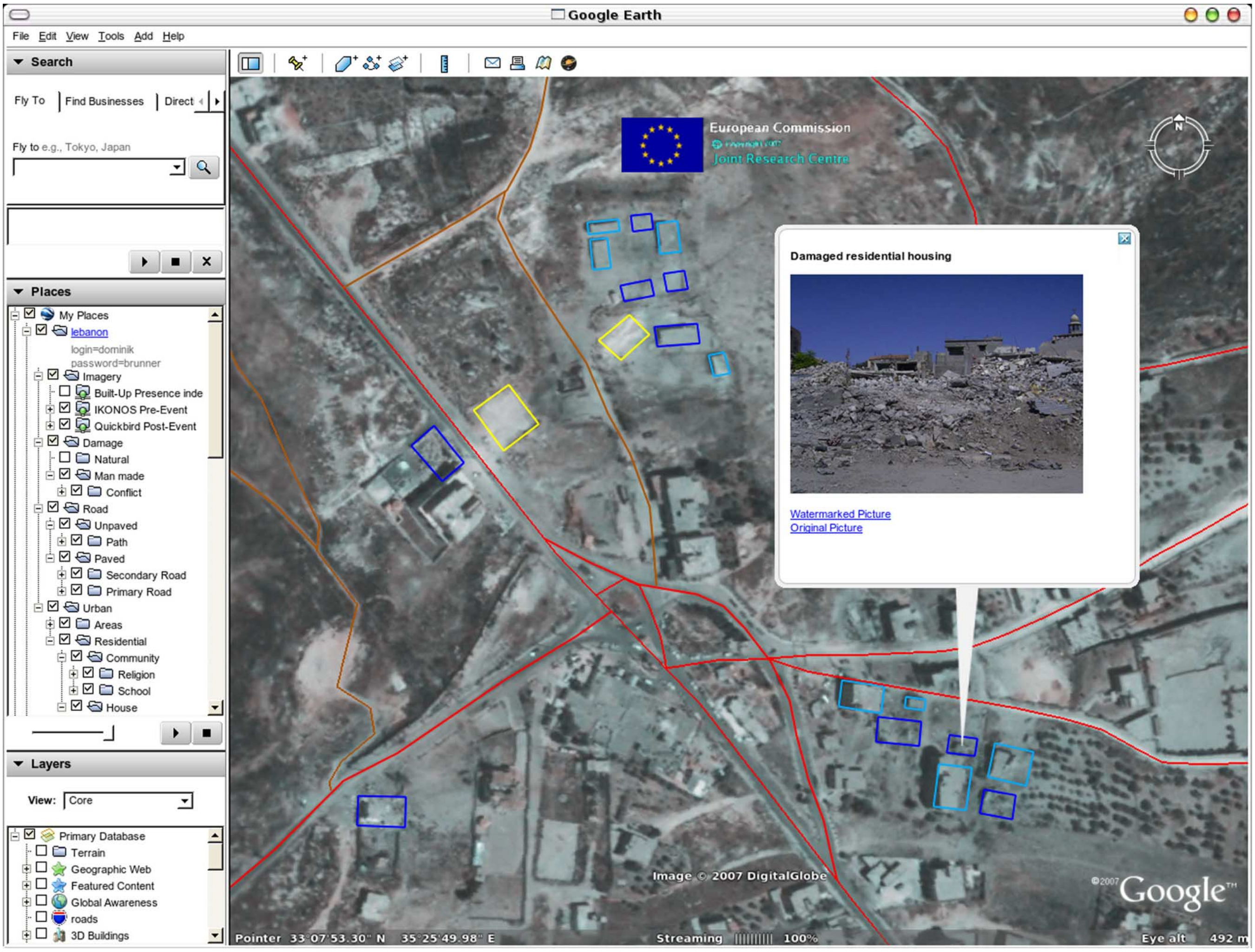

Fig. 8. Screenshot of the Google Earth™ client interface showing the result of the feature capturing for an area in Bent Jbail, South Lebanon. Paved roads are marked red, unpaved streets are shown in brown. Moderately damaged buildings are outlined with light blue and destroyed buildings with dark blue polygons. The yellow polygons mark impact craters in non-built-up terrain. The pop-up shows the image of a destroyed building which is associated with the corresponding blue polygon. Satellite image © DIGITALGLOBE, 2006, distributed by EURIMAGE S.p.A.

South Lebanon in 2006. The system was also used during the recent Sichuan earthquake (China, May 12, 2008) and the Georgia armed conflict (August 2008) in support of damage assessment. Note that in the first case, very high-resolution satellite pre- and postevent imagery was provided for public access, as `SuperOverlays`, directly by the satellite image providers [35]. More recently, NOAA has demonstrated a similar approach for the analysis of airborne orthophotos supporting relief efforts after the September 2008 Hurricane Ike [36].

The system functionality is sufficiently generic to be used in other contexts. We believe it is particularly useful in applications that require near real time access to event information for which *in situ* or domain expert knowledge is essential in automated and manual image interpretation.

Our system is composed of free software and open source components, but the system structure is sufficiently generic to accommodate other proprietary or free software modules that implement the same functionality. The choice of the Google Earth™ client is optional and may be replaced with other client software that implements the `SuperOverlay` construct. We are currently evaluating the Java release of the open source World Wind client as an alternative. We believe that open source and public software components better address our long term goal to distribute geospatial processing capabilities to relevant end users and contributers within the emergency response domain, without the need for significant and recurring investment. This belief is in line with the mission of the UN SPIDER program whose goal is to "ensure that all countries have access to and develop the capacity to use all types of space-based information to support the full disaster management cycle" [37].

Future work on the system will focus on the implementation of relevant standards. For instance, we intend to include SensorML [38] in our `SuperOverlays` to support image data discovery services. Currently, we have implemented a simple mechanism for the triggering of image processing and geospatial querying via the Google Earth™ forwarding mechanism, using a fixed set of directives. We intend to extend this to protocols that follow the OGC Web Processing Service standard [39]. The feature capturing functionality will be tailored to support evolving mapping standards for legend hierarchies and related styling. Furthermore, we intend to test our system using

GPS-enabled portable devices, such as smart phones and PDAs, to support collection of *in situ* data in the crisis area in near real time. Data licensing and sensitivity issues would also need to be addressed with an enhanced system security configuration. At the same time, however, we will embark on further testing involving colleagues and partners who work in parallel application domains.

Acknowledgment

The authors would like to thank JRC colleagues A. Gerhardinger for the preprocessing of the satellite imagery, S. Eckert for providing access to the ground truth data, and C. Bielski for critically reviewing the content. The authors would also like to thank the Open Source communities maintaining the essential GDAL, PostgreSQL, PostGis, GeoServer, and Apache components, without which they would not have been able to create their system.

References

[1] Space and Major Disasters 2000 [Online]. Available: http://www.disasterscharter.org, International charter, [Online]. Available

[2] Geo: Group on Earth Observation [Online]. Available: http://www.earthobservations.org [Online]. Available

[3] B. McGuire, I. Mason, and C. Killburn, *Natural Hazards and Environmental Change (Key Issues in Environmental Change)*. London, U.K.: Arnold, 2002.

[4] P. Hoyois, J.-M. Scheuren, R. Below, and D. Guha-Sapir, Annual Disaster Statistical Review: Numbers and Trends 2006, Brussels, Belgium, Tech. Rep., 2007, CRED.

[5] S. Voigt, T. Kemper, T. Riedlinger, R. Kiefl, K. Scholte, and H. Mehl, "Satellite image analysis for disaster and crises-management support," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1520–1528, Jun. 2007.

[6] F. Bovolo and L. Bruzzone, "A split-based approach to unsupervised change detection in large-size multitemporal images: Application to tsunami-damage assessment," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1658–1670, Jun. 2007.

[7] P. Gamba, F. Dell'Acqua, and G. Trianni, "Rapid damage detection in the bam area using multitemporal sar and exploiting ancillary data," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1582–1589, Jun. 2007.

[8] L. Bruzzone and D. F. Prieto, "An adaptive parcel-based technique for unsupervised change detection," *Int. J. Remote Sens.*, vol. 21, no. 4, pp. 817–822, Mar. 2000.

[9] L. Bruzzone and D. F. Prieto, "Automatic analysis of the difference image for unsupervised change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1171–1182, May 2000.

[10] L. Bruzzone and D. F. Prieto, "A minimum-cost thresholding technique for unsupervised change detection," *Int. J. Remote Sens.*, vol. 21, no. 18, pp. 3539–3544, Dec. 2000.

[11] F. Bovolo and L. Bruzzone, "A detail-preserving scale-driven approach to unsupervised change detection in multitemporal sar images," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 12, pp. 2963–2972, Dec. 2005.

[12] F. Chaabane, A. Avallone, F. Tupin, P. Briole, and H. Maitre, "A multitemporal method for correction of toposheric effects in differential sar interferometry: Application to the gulf of corinth earthquake," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1605–1615, Jun. 2007.

[13] T. D. Groeve, L. Vernaccini, and A. Annunziato, B. V. de Walle and M. Turoff, Eds., "Global disaster alert and coordination system," in *Proc. 3rd Int. Conf. Information Systems for Crisis Response and Management (ISCRAM)*, Newark, NJ, May 13–17, 2006, pp. 1–10.

[14] P. Barbosa, J. Kucera, P. Strobl, A. Camia, G. Amatulli, and J. San-Miguel, "European forest fire information system (effis) rapid damage assessment: Appraisal of burnt area maps in southern europe using modis data (2003 to 2006)," in *Proc. 5th Int. Conf. Forest Fire Research*, Portugal, Nov. 27–30, 2006, Figueira da Foz, Elsevier.

[15] A. De Roo, C. Wesseling, and W. Van Deurzen, "Physically based river basin modelling within a gis: The lisflood model," *Hydrol. Process.*, vol. 14, no. 11, pp. 1981–1992, 2000.

[16] A. Berson, *Client/Server Architecture*. New York: McGraw-Hill, 1992.

[17] T. Mitchell, *Web Mapping Illustrated*. Sebastopol, CA: O'Reilly Media, Jun. 2005.

[18] I. Sommerville, *Software Engineering*, 8th ed. Harlow, U.K.: Addison-Wesley, 2007.

[19] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley, 1995.

[20] Kml 2.1 Reference [Online]. Available: http://code.google.com/apis/kml/documentation/kmlreference.html

[21] Geospatial Data Abstraction Library 2007 [Online]. Available: http://www.gdal.org/

[22] Open Geospatial Consortium [Online]. Available: http://www.opengeospatial.org

[23] Open Geospatial Consortium, Opengis Web Coverage Service 1.1.2 (wcs) 2007 [Online]. Available: http://www.opengeospatial.org/standards/wcs

[24] Open Geospatial Consortium, Opengis Web Map Service 1.3.0 (wms) 2004 [Online]. Available: http://www.opengeospatial.org/standards/wms

[25] D. Stanfill, "Using image pyramids for the visualization of large terrain datasets," *Int. J. Imag. Syst.*, vol. 3, 1991.

[26] Kml 2.1 Tutorial [Online]. Available: http://code.google.com/apis/kml/documentation/kml_21tutorial.html#super overlays

[27] G. Lemoine and D. Brunner, "Integration of full resolution image coverages using superoverlays," presented at the Freie Und Open Source Software Fuer Geoinformationssysteme (FossGIS), Berlin, Germany, Mar. 13–15, 2007 [Online]. Available: http://www.fossgis.de/wiki/images/a/a5/Lemoine_Brunner_FossGIS2007_Berl in.pdf

[28] G. Lemoine, D. Brunner, and F.-X. Thoorens, "Superoverlay deployment in grid-enabled image processing," presented at the Int. Geoscience and Remote Sensing Symposium (IGARSS), Boston, MA, Jul. 7–11, 2008.

[29] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman, Eds. San Francisco, CA: Morgan Kaufmann, Nov. 1998.

[30] Open Geospatial Consortium, Opengis Web Feature Service 1.1.0 (wfs) 2006 [Online]. Available: http://www.opengeospatial.org/standards/wfs

[31] Open Geospatial Consortium, Opengis Styled Layer Descriptor 1.1.0 (sld) 2007 [Online]. Available: http://www.opengeospatial.org/standards/sld

[32] R. Welch and M. Ehlers, "Merging multiresolution spot hrv and landsat tm data," *Photogramm. Eng. Remote Sens.*, vol. 53, no. 3, pp. 301–303, 1987.

[33] J. Grodecki and G. Dial, "Block adjustment of high-resolution satellite images described by rational functions," *Photogramm. Eng. Remote Sens.*, vol. 69, no. 1, pp. 59–70, Jan. 2003.

[34] M. Pesaresi, "Texture analysis for urban pattern recognition using fine-resolution panchromatic satellite imagery," *Geograph. Environ. Model.*, vol. 4, no. 1, pp. 43–63, May 2000.

[35] Google Lat Long Blog—Imagery for Sichuan, China Earthquake 2008 [Online]. Available: http://google-latlong.blogspot.com/2008/05/imagery-for-sichuan-china-ea rthquake.html

[36] National Oceanic and Atmospheric Administration's National Geodetic Survey (Noaa)—Hurricane Ike 2008 [Online]. Available: http://ngs.woc.noaa.gov/ike/

[37] United Nations Platform for Space-Based Information for Disaster Management and Emergency Response (Spider) 2007 [Online]. Available: http://www.unoosa.org/oosa/unspider/index.html

[38] Open Geospatial Consortium, Opengis Sensor Model Language (Sensorml) 2008 [Online]. Available: http://www.opengeospatial.org/standards/sensorml

[39] Open Geospatial Consortium, Web Processing Service (Wps) 2008 [Online]. Available: http://www.opengeospatial.org/standards/wps

Dominik Brunner (S'07) received the Diploma in technical computer science from the University of Applied Sciences Hof, Germany, in 2004.

From 2004 to 2006, he was a software engineer at SAP AG, Walldorf, Germany. Since 2006, he has been with the European Commission—Joint Research Centre, Ispra, Italy, and also with the Remote Sensing Laboratory, Department of Information Engineering and Computer Science, University of Trento, Italy. His research interests include the field of remote sensing, pattern recognition, and image processing in support to damage assessment and emergency response, in particular the analysis of VHR SAR data of urban areas.

Guido Lemoine (S'92–M'95–SM'08) is an agricultural engineer by training (1987, Soil Science, Wageningen University, The Netherlands). His main expertise is in applied remote sensing, first developed as a research topic (microwave backscattering of soils) and later as a commercial activity in a co-founded remote sensing and GIS consultancy. Since he joined the European Commission's Joint Research Centre, Ispra, Italy, in 1997, he has further developed his remote sensing and informatics expertise in agricultural statistics and subsidy control applications, in fisheries monitoring, and since 2006, in civil security applications. His current work focuses on integration of very high-resolution optical and SAR data in crisis response, collaborative geospatial analysis, and fast computing methods for near real-time mapping of crisis event impact.

Francois-Xavier Thoorens is a telecommunication engineer with a specialization in image processing (Telecom, Paris Sud 2004). Since 2004, he has been a GIS and satellite SAR image analysis specialist in the fisheries monitoring team of the European Commission's Joint Research Centre, Ispra, Italy. He is the lead developer of the SUMO vessel detection software that is widely used in European maritime surveillance projects. His research interests include the use of remote sensing techniques for crisis response and security applications in a maritime context, such as environmental pollution, illegal trafficking, and vessel surveillance.

Lorenzo Bruzzone (S'95–M'98–SM'03) received the laurea (M.S.) degree in electronic engineering (*summa cum laude)* and the Ph.D. degree in telecommunications from the University of Genoa, Genoa, Italy, in 1993 and 1998, respectively.

From 1998 to 2000, he was a Postdoctoral Researcher at the University of Genoa. In 2000, he joined the University of Trento, Italy, where he his currently a Full Professor of telecommunications. He teaches remote sensing, pattern recognition, and electrical communications. He is the Head of the Remote Sensing Laboratory, Department of Information Engineering and Computer Science, University of Trento. His current research interests are in the area of remote-sensing image processing and recognition (analysis of multitemporal data, feature extraction and selection, classification, regression and estimation, data fusion, and machine learning). He conducts and supervises research on these topics within the frameworks of several national and international projects. He is Evaluator of project proposals for many different governments and scientific organizations. He is the author (or coauthor) of 73 scientific publications in referred international journals, more than 130 papers in conference proceedings, and seven book chapters. He is a referee for many international journals and has served on the scientific committees of several international conferences.

Dr. Bruzzone is a member of the Managing Committee of the Italian Inter-University Consortium on Telecommunications and a member of the Scientific Committee of the India-Italy Center for Advanced Research. He ranked first place in the Student Prize Paper Competition of the 1998 IEEE International Geoscience and Remote Sensing Symposium (Seattle, WA, July 1998). He was a recipient of the Recognition of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING Best Reviewers in 1999 and was a Guest Editor of a Special Issue of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING on the subject of the analysis of multitemporal remote-sensing images (November 2003). He was the General Chair and Co-chair of the First and Second IEEE International Workshop on the Analysis of Multi-Temporal Remote-Sensing Images (MultiTemp), and is currently a member of the Permanent Steering Committee of this series of workshops. Since 2003, he has been the Chair of the SPIE Conference on Image and Signal Processing for Remote Sensing. From 2004 to 2006 he served as an Associated Editor for the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS and currently is an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. Since 2009, he as been a member of the Administration Committee of the IEEE Geoscience and Remote Sensing Society. He is also a member of the International Association for Pattern Recognition and the Italian Association for Remote Sensing (AIT).