3DNSITE: A networked interactive 3D visualization system to simplify location awareness in crisis management

Giovanni Pintore* CRS4 — Enrico Gobbetti CRS4 — Fabio Ganovelli CNR — Paolo Brivio CNR

Abstract

We report on the 3DNSITE system, a web-based client-server 3D visualization tool for streaming and visualizing large tridimensional hybrid data (georeferenced point clouds and photographs with associated viewpoints and camera parameters). The system is motivated by the need to simplify data acquisition and location recognition for crisis managers and first responders during emergency operations or training sessions. In this peculiar context, it is very important to easily share 3D environment data among people in a distributed environment, accessing huge 3D models with embedded photographs on devices with heterogenous hardware capabilities and interconnected on different network types. Moreover, since the specific end-users are not necessary skilled with virtual reality and 3D objects interaction, the navigation interface must be simple and intuitive. Taking into account these constraints, we propose a mixel object-based/image-based system, which enhances the current state-of-the-art by exploiting a multi-resolution representation for the 3D model and a multi-level cache system for both the images and 3D models structure. A novel low-degree-of-freedom user interface is presented to navigate in the scenario with touchscreen devices. The proposed implementation, included in a more general training and decision framework for emergency operations, is evaluated on real-world datasets.

CR Categories: I.3.2 [Computer Graphics]: Graphics Systems—Distributed/network graphics I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques I.3.7 [Computer Graphics]: Three-dimensional graphics and realism—Virtual reality I.3.8 [Computer Graphics]: Applications

Keywords: Virtual reality, 3D interaction, Input and interaction technologies, Visualization

1 Introduction

Modern communities have experienced a spate of catastrophic events in recent years. The combination of dense population concentrations (36.6M in Tokyo, 10.5M in Paris [UN 2009]) and complex and large architectural environments make it very hard to anticipate, prepare for and manage the impact of natural, industrial, or man-made disasters. In this ever-changing environment, it is essential for public authorities to design proper emergency plans, train security organizations and crisis managers through simulations, and to effectively handle crisis management procedures. To serve these critical needs new approaches and technologies are thus researched and developed.

*CRS4 Visual Computing Group, POLARIS Edificio 1, 09010 Pula, Italy. www: http://www.crs4.it/vic/ e-mail: {gianni}@crs4.it

Figure 1: ***Crisis management and simulation setup.*** *Top left: a control room virtual whiteboard supporting multitouch controls. Top right: a tablet employed during a training session. Bottom left: firebrigates truck hosting a field command post. Bottom right: on truck field command post control room.*

Context and objectives One of the directions of research among these new approaches is to aid remote navigation/location awareness in complex environments through the exploitation of 3D or near-3D data. Two data types are of particular importance to the security domain: extremely massive point clouds and geo-referenced (three-dimensional) photographs, since they can be acquired very rapidly and provide both measurable and visually recognizable description of a site. The basic idea behind the approach is that 3D data of some form (shapes of buildings/environments) is already widely available, and will become more common in the future thanks to the improvement in quality and reduction in cost of 3D acquisition technologies. With this information, that can go from complete 3D reconstruction of sites (e.g. as those acquired by aerial or terrestrial 3D laser scanners) to 3D calibrated photographs it is possible to present users with an easy to understand depiction of a natural or man-made environment. The joint visualization of 3D models, geo-referenced 3D aligned images and any other content that can be represented as a geographical position or direction, can thus complement traditional 2D maps in a number of tasks. The end-users of this kind of systems can be divided in three main categories: first responders, crisis managers, and their trainers. First responders are operational units from police, fire department, and medical services, that operate on the field during a crisis situation. Crisis managers are specialized strategists whose supervision is vital for managing, organizing, and coordinating the operations, especially in presence of large incidents. Their decisions must be carried out rapidly and carefully, defining the specific tasks of each first responder. In order to grant a higher level of skill and professionalism, trainers teach and prepare crisis managers and first responders

Figure 2: ***Screenshots of live navigation using the 3DNSITE viewer.*** *Interactive 3D navigation over the gas storage site of Geomethane in Manosque (France). This site has been used as training emergency scenario on the framework involving 3DNSITE (the 3DNSITE system is in charge to store, stream – server side – and visualize – client side – tridimensional hybrid data). The framework involves the Operational Center of the Fire and Rescue Services of Alpes de Hautes, the local Gendarmerie, and many real crisis managers and first responders. The company is an underground hydrocarbon storage site with a* 7.5 *million* m^3 *storage capacity. It was created in 1969 and is classified as a SEVESO (high-risk production site), thus it has been an excellent test case for our system.*

for real crisis events. First responders can be equipped with portable devices, such as touchscreen tablets, or similar (see Fig. 1 up right). Managers and trainers might have access to portable devices as well as large touchscreen displays, and use them as a kind of whiteboard (see Fig. 1 up left). In the typical scenario of a crisis evolution, the operators from the control rooms and the agents from the field need also to share the same data, thus the visualization system is required to run in a web-based environment [Marvie et al. 2011]. In addition, the data can be dynamic, since the real or simulated scenario evolves and agents need to constantly monitor the state of the crisis context. Specifically, the targeted datasets should be updated at runtime by adding new images which should be shown to the agents. Finally, the dataset itself could be interfaced and integrated with institutional or military information systems.

Challenges Within this peculiar context, several issues and research challenges should be faced. One of the first and most critical requirements in those systems is how to achieve efficient data distribution and good scalability. In fact, the same dataset must be browsed and shared between control rooms and several agents deployed on the field. Granting each actor a complete representation of the crisis scenario is also challenging, since each can be equipped with diversified hardware and network resources (see Fig. 1). Except for the main control room setup, agents usually work with portable devices. These clearly impose strong limitations in terms of 3D capabilities, memory, storage and network bandwidth. The control rooms themselves can be also, though not necessarily, located close the crisis site (e.g., firebrigates truck), experiencing a lack of resources in terms of space, electric power supplies, etc. Focusing the attention on the visualization side of the problem, though many solutions are presented in literature distinctly for 3D model and images, the combined visualization of 3D models and aligned images needs specific solutions, both in terms of hardware and network resources scalability and in terms of user interface easiness. The design of the user interface and interaction scheme poses other important issues. Direct experience with crisis context actors (e.g., police, fire departments, medical services) has put in evidence their attitudes and specific needs when exploring the set of images. These typical users are not used to virtual reality or 3D objects interaction and usually they prefer to interact with immediate and simple aided interfaces. For example, the actors ask for efficient ways to select and browse images represented in the 3D world, in place of a fully free 3D navigation through the 3D model. The navigating between views should also be fast and direct to prevent experiencing a "feeling lost" impression. An additional option explored in the emergency context is the opportunity to update the scenario according to the real, or the simulated, evolution of the crisis. To achieve this, one or more images should be inserted in the existent dataset, either if they are acquired directly by a mobile client, or if collected in the main control room from various sources.

Contributions Taking into account these constraints, we propose several solutions, both adapting already presented state-of-the-art methods and enhancing them, in order to simplify location recognition and visualize dynamically updated pictorial data inside a virtual 3D environment. We integrate these solutions into the 3DNSITE system, a specific client-server visualization tool included in a more general training and decision framework to support emergency operations. The focus of 3DNSITE is to handle, store, stream and visualize tridimensional hybrid data (e.g., point clouds, meshes, embedded 3D aligned photographs), in order to simplify location recognition for first responders, crisis managers and trainers. We achieve scalability over network, storage, and computational resources by combining a priority-based, multi-level cache system with a multiresolution, dynamic, hierarchical representation of the 3D model based on current state-of-the art solutions for points [Gobbetti and Marton 2004] and meshes [Cignoni et al. 2005]. This view-dependent adaptive approach for the 3D model ensures scalable performance in host-to-graphics and network communications, and completes the scenario depiction with a good approximation when no real photo is available from some point of view, without any need to synthesize artificial images. Following the same output-sensitive philosophy, we present also a novel low-degree-of-freedom user interface to easily navigate between 3D aligned photographs as well as their viewports. The navigation method is explicitly designed for touchscreen portable devices (e.g., tablets) and control room electronic whiteboards. This method is articulated in two steps. In the out-of-core pre-processing phase, starting from the 3D model and the related aligned images, all the viewports are organized and stored on a server, according to image-to-image distances in a 6D calibration space. Using extra image information (e.g., coming from EXIF metadata, or GPS/AGPS records taken during the data acquisition campaign), at preprocessing a geo-referenced frame is also calculated and stored both for the 3D model and the 3D embedded images. All of this data can be considered as a kind of documentation of the site and is added to the dataset to be simultaneously shared by the various client devices. Then, at runtime the client accesses the dataset through HTTP, using the pre-computed informations and the touchscreen input to predict and address the user navigation.

Our results analyze data gathered for different kinds of portable de-

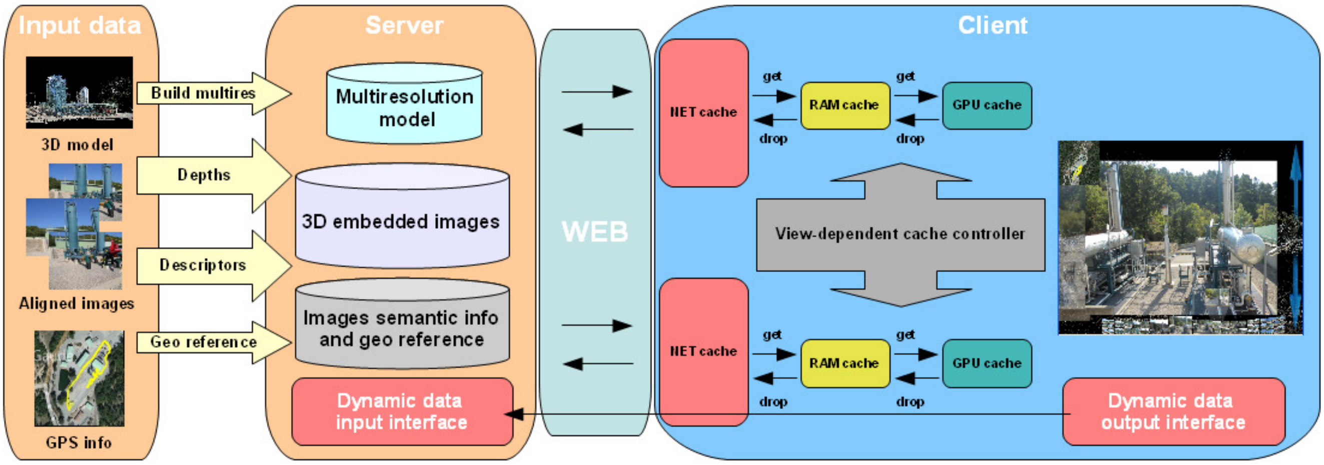

Figure 3: ***Architecture overview.*** *Starting from images already aligned with a 3D model and GPS data we perform an out-of-core pre-processing phase where the original 3D model is processed and stored in a multiresolution structure; images depths, descriptors, semantic distances and a geo spatial reference frame are computed. These informations are stored in a metadata index file and used both for 3D environment aided navigation and to set the images cache priorities according with the view paramaters. The 3D model and the geographical reference frame are considered as a kind of "skeleton" of the scenario, where pre-existent images and new images are 3D embedded. From the index file containing the list of repositories the clients access the data by HTTP protocol. The rendering and the caches are updated according with the current viewport parameters, using 3 cache levels: HTTP, RAM and GPU, at the same time the pre-computed informations are exploited to predict and address the user navigation. At run-time one or more images can by inserted in the existent dataset, acquired directly by a mobile client or collected in the main control room by various sources.*

vices, on which caching and multiresolution support prove to be mandatory to achieve interactivity (30fps vs 1fps). The effectiveness of the system is demonstrated on a number of real world scenarios of the global framework in which 3DNSITE is included (see Fig. 2).

2 Related work

Despite the growing need of decision support and crisis simulation systems (see [Boin 2009]), the research in this area had paid little attention to the use of IT-based simulation tools (see [Dugdale J. and N. 2010]) until recent years. Nowadays the increasingly complex nature of crisis management demands the support of Virtual Reality technologies, especially for the simulation of complex crisis and contingency scenarios that would be difficult to recreate and validate in real conditions. In the decision support and crisis management field only few training methods have been proposed and partially employed. Moreover, they have still not been fully translated into working software [Palen et al. 2007] [Lanfranchi and Ireson 2009]. Crisis managers in control rooms are often away from the crisis location and are not offered any real image of the disaster. At the same time, agents deployed on site at first need to quickly orient themselves and to obtain a clear understanding of the situation. Indeed, sharing the data in a collaborative network is an important part of the decision support and increases the efficiency of the actions [Carver and Turoff 2007] [Mouton et al. 2011] . Systems such as 3DNSITE try to match these requirements improving the virtual exploration, navigation, and location awareness through the exploitation of massive point clouds and three-dimensionalized photographs. Several solutions have been presented in literature such as maps browsing, 3D mobile navigation as well as photo-browsing, but only a few existing browsers support the joint navigation of mixed 2D and 3D datasets [Snavely et al. 2006], [Vincent 2007], [Snavely et al. 2008] [Kopf et al. 2010] [Goesele et al. 2010]. 3DNSITE uses a navigation paradigm derived from that of Google StreetView [Vincent 2007], Photosynth [Microsoft 2007] and, much earlier, Movie-Maps [Lippman 1980], where the scene is visualized from predefined points of view. 3DNSITE also computes smooth transitions between aligned photos obtaining an effect similar to Photo Tourism's [Snavely et al. 2006], but without adopting any proxy geometry. During the navigation, photographs are embedded on-the-fly in the 3D world by dynamically projecting images onto the 3D geometry, as suggested by [Brivio et al. 2012]. Several other projective approaches have been considered for this purpose [Pintus et al. 2011a], but experience with end-users of real-crisis context has suggested an approach where the images are projected preserving their original viewport, setting the user's virtual position exactly at the photographer's location. Unlike the mentioned 3D photo browsers, 3DNSITE enables the user to perform free-point-of-view browsing at interactive frame rates, exploiting the presence of dense point clouds to provide the user with a representation of the scene when no photograph is available. This feature, combined with state-of-the-art multi-resolution representations of the dataset, is another important key-strength of the system with respect to the other photo browsers. Since 3DNSITE is meant to run on portable devices with limited resources, the targeted large 3D models cannot be rendered in real-time. For this reason, 3DNSITE employs a level-of-detail data structure derived from state-of-the-art work on clustered multi-resolution structures for high resolution polygonal models [Yoon et al. 2004], [Cignoni et al. 2004], [Cignoni et al. 2005] and for massive point clouds [Gobbetti and Marton 2004], [Kasik et al. 2008]. Our implementation supports rendering at interactive pace by selecting a representation that fits the available device resources in function of the current view frustum. In addition, this data structure naturally lends itself to an efficient memory management which we also adopt for managing the photographs. This view-dependent multi-resolution system also ensures interactive frame rates when the dataset is massive and complex and the data are being streamed to limited portable devices. We apply the same view-dependent/output-sensitive philosophy to the image cache controller exploiting precomputed image descriptors to determine the caching priority of each image.

3 System overview

Fig. 3 provides a general overview of our system. The first step consists of an out-of-core pre-processing of the input data, whose output is stored on the server ready to be efficiently accessed by the various visualization devices. The original 3D model is processed and stored as a hierarchical multiresolution structure. In a parallel process, the viewports of the images aligned onto the 3D model are organized in a linear sequence reflecting their similarity in a 6D calibration space. Using images extra info (e.g., EXIF data) or GPS/AGPS tracks recorded during the data acquisition campaign, a geo spatial reference frame is calculated and stored for 3D model and the 3D embedded images. The 3D model and the geographical reference frame are considered as a kind of "skeleton" of the scenario, where pre-existent images and new images will 3D embedded. The server then contains two basic repositories: one for the 3D model and one for the images. This data is indexed in a metadata file, which also contains all the precomputed viewport informations and the geographical references of the images. At runtime, the clients access this data through the HTTP protocol, starting from the index file. To this purpose, a priority-based, multi-level cache system is employed to remotely access the high resolution images and the 3D model multiresolution structure. The various items of data are downloaded, cached into RAM, and eventually uploaded into the GPU according to a scalar priority value which aims at describing the relevance of the data with respect to the current virtual 3D view. The index metadata information is independently used by each host to determine the cache priorities, also taking in account the data intersection with the current view frustum. During the exploration of the dataset, the rendering and the caches are updated according to the current view parameters. At the same time the precomputed informations are exploited to predict, address, and aid the user 3D navigation. During the navigation, one or more images, acquired directly by a mobile client or collected in the main control room from various sources, can be added to the existent dataset. Either case, the 3DNSITE system processes them, regenerates the metadata file, and replaces the previous one on the server. Note that this operation is very fast and can thus be performed either locally on the server or remotely (e.g., the client itself can send to the server the new image and updated metadata file). Thanks to our web-based organization of the data, each host then gets updated sharing the new data.

4 Methods and tools

4.1 Server side pre-processing

We assume that the photographs are calibrated with the associated 3D model, such as those for example created with Structure from Motion pipelines [Snavely et al. 2006], [Pintus et al. 2011b] [Tuite et al. 2011] [Wan et al. 2012]. The first step is to transform the original 3D model (usually a point cloud) in a multi-resolution structure, in order to handle datasets that exceed the capacity of the client GPU RAM and efficiently scale over portable devices with limited bandwidth resources. The construction process creates a hierarchy over the samples of the datasets simply by reordering and clustering them into point clouds of approximately constant size arranged in a tree. In other words, the final multi-resolution model has exactly the same points of the input model, but grouped into patches and organized in a level of detail representation. The root of the level of detail tree represents the entire model with a single cloud. These patches at different resolution can be assembled in different combinations to produce the full model. The partitioning procedure takes as input an external memory array of uniformly distributed point samples, together with its bounding box, and recursively generates a hierarchical structure by space partitioning (top-down step), then constructing non-leaf cells by bottom-up recombination and simplification of lower level cells. Dependencies between mesh modifications are arranged in a DAG besides assigning model space errors and bounding volumes to cells. Variable resolution representations of the models are obtained by defining a *cut* of the DAG and merging all nodes above the cut. At run-time, selective refinement queries based on projected error estimation and regions of interest are performed on the multi-resolution hierarchy to rapidly produce view-dependent continuous model representations by combining precomputed patches. The benefits of this approach are that the workload required for a unit refinement/coarsening step is amortized on a large number of point primitives, and the small point clusters can be optimized off-line for best performance in host-to-graphics and network communications. This hierarchical data structure is split in a index tree and a point cloud (or triangles) repository. The access to this repository is made through an output-sensitive/view-dependent controlled cache system. The same view-dependent philosophy combined with the precomputed image descriptors is employed to control the accesses to the images cache system (see 4.2). In a second phase, each high-resolution image is compressed into JPEG format and the minimal image-space depth of its content is precomputed and stored. To determine this quantity, a depth buffer of the 3D model is rendered from the image viewpoint, as defined in the camera calibration. Image ordering and distances are also precomputed, as well as an abstract descriptor associated to each image. This descriptor, used to estimate good orderings and the semantic distances among images, is a weighted average of time-of-shot, image shot position, image shot orientation, color distribution, spatial-color-layout and image depth (6D space). The descriptors are exploited at run-time both to drive the images cache priorities and to aid the user interaction. High-resolution images and all the precomputed metadata are stored in their proper repository ready for compression and streaming over HTTP. In addition, using the images' extra info (e.g. EXIF data) or GPS/AGPS records taken during the data acquisition campaign, a geo spatial reference frame is calculated and stored for 3D model and the 3D

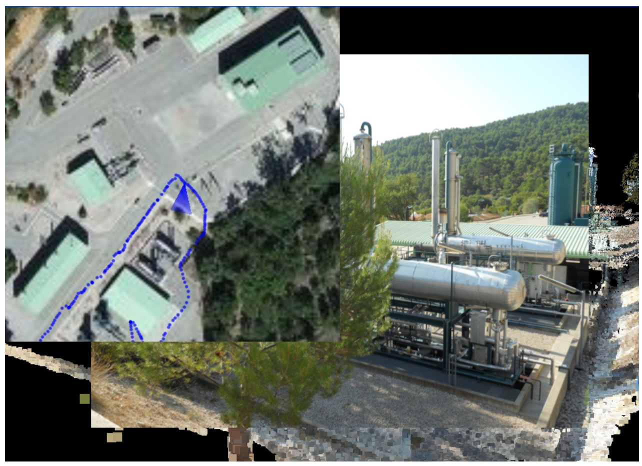

Figure 4: ***GPS track recorded at Geomethane site*** *The entire the dataset has been georeferenced by comparing the GPS record of the agent who took the pictures with the shot positions estimated by the structure-from-motion pipeline.*

embedded images (see Fig. 4). The GPS records are compared with the shot positions given by the input data and the geographical spatial reference frame is calculated using a $RANSAC$ method based on [Capel 2005], [Chum et al. 2003], [Torr and Zisserman 2000] and [Fischler and Bolles 1981]. The method achieves good results and an acceptable precision for this specific application.

4.2 Data distribution

3DNSITE employs a priority-based, multi-level cache system which strives to optimize data access and the allocation of network and hardware resources. The cache pipeline is composed of three levels: HTTP, RAM and GPU, with an additional fourth DISK level between HTTP and RAM provided when required. Such a cache system is required to manage thousands of items and frequent priority updates and locking, as well as the synchronization of different threads. Each cache level operates in its own thread allowing for blocking operations on files and sockets greatly simplifying the implementation. A priority is assigned to the 3D data blocks as determined by the multi-resolution 3D subsystem, which takes in account visibility, distance, resolution, and so on. The aligned images are loaded from the network through the cache system in accordance with the priority rule. Given a fixed size budget of RAM memory and GPU memory (depending of the hardware resources) the client starts to load the images giving the highest priority to the current camera viewport, and giving decreasing priority as the semantic difference with the current image increase (Fig. 5). The minimum amount of cache instances is two: one for the 3D data blocks and one for the images (see Fig. 3). However, more caches can be instantiated for additional image collections (e.g. thumbnails or temporally different image sets). Given the typical size of a dataset in this

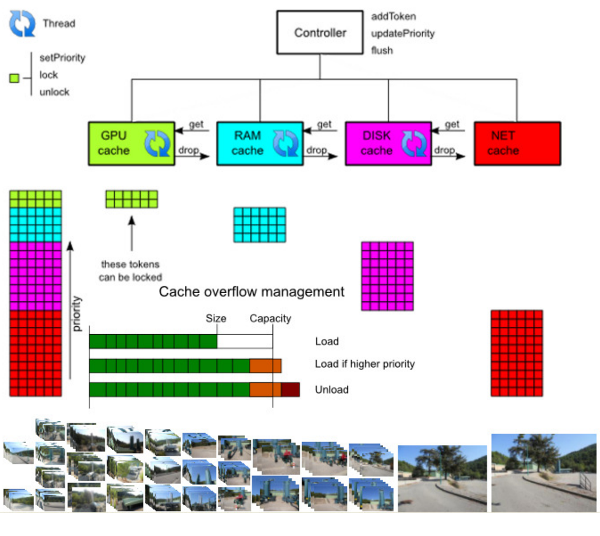

Figure 5: ***Multi-level priority-based cache system*** *The aligned images are loaded from the network through the cache system following a priority rule. Given a fixed size budget of system RAM and GPU memory (depending on the hardware resources) the client starts to load the images giving the highest priority to the current camera viewport, and while decreasing priority as the semantic difference with the current image increases.*

context, compression techniques are also necessary. In a cache system, decompression can be considered a part of the process of loading an item from a lower to a higher cache level. Using a compression scheme means trading some loading speed for storage space and bandwidth. 3D data blocks are stored remotely already in the form of Vertex Buffer Objects (VBOs). Compression/decompression of 3D models have also been considered, but available systems do not deliver enough deflating speed for our purposes. For the images, a JPEG compression scheme has been adopted. Though the JPEG decompression cannot be easily implemented on GPU level without specific hardware requisites, this system assures an advantageous compression ratio (e.g. 200:1) and a wide compatibility with different devices.

4.3 Client side 3D navigation

Figure 6: ***Highlight of the next image candidate*** *Live navigation in the Geomethane scenario. By touching a point in bottom left corner of the screen the most representative image for that spatial area is highlighted. With a double click the camera automatically moves to the point of view from which the image was taken.*

In our current implementation we use an persistent HTTP/1.1 connection to transfer the 3D data blocks, optionally employing HTTP pipelining. This method has been used successfully in [Bettio et al. 2007] [Gobbetti and Marton 2004] for the same purpose. The combination of these two techniques improves bandwidth usage and reduces network latency while keeping the protocol's API simple, since clients benefit from an underlying connection-based implementation hidden under a reliable connectionless interface. The pipelining approach allows multiple HTTP requests to be written together out to the socket connecting client and server without waiting for the corresponding responses. The client then waits for the responses to arrive in the order in which they were requested. Since each 3D data block consists of several thousands of points or triangles already precomputed in the preprocessing step, assembling the view-dependent representation at rendering time is extremely fast and results in very low CPU load. Each block is optimized, cached in the GPU through the multi-level cache system, and rendered with a single CPU call for maximum performance. The rendering algorithm selects the best representation according to the rendering budget and the availability of the blocks, thus guaranteeing a lower bound on the frame rate. The resulting technique has the following properties: it is fully adaptive and is able to retain all the original topological and geometrical detail, even for massive datasets; it is not limited to meshes of a particular topological genus or with a particular subdivision connectivity; it preserves geometric continuity of variable resolution representations at no run-time cost; it is strongly GPU-bound and is over one order of magnitude faster than existing adaptive tessellation solutions on current PC platforms, its patch-based structure successfully exploits on-board caching, cache coherent stripification, compressed out of core representation and speculative prefetching for efficient rendering on commodity graphics platforms with limited main memory; it enables high-quality, simplified representations to be constructed with a distributed out-of-core simplification algorithm. As already said in Subsection 4.2, the aligned images are loaded from the network through a cache system. The cache controller is output-sensitive, deciding the load priorities according with current viewport, image descriptor differences and hardware capabilities. The requests are stopped when the memory budget is filled and restart if the user changes his point of view and consequently the current viewport/image. To present the photographs embedded and projected in the 3D world during the navigation, we adopt a projective texturing approach, with an effect equivalent to cast the image as a slide from a virtual projector into the 3D scene, simulating the viewport from which the photograph has been originally taken the photo. This solution has been chosen after having considered different projection methods, meeting the end-users requirements which prefer to see the real images as they have been taken without any projection artifact. The image rectangle is defined as the section of the view frustum pyramid of the corresponding shot, cut at distance D from the camera and roughly corresponding to the precomputed minimum depth of the objects featured in the image. When the view-position discrepancy increases the texture projection is progressively disabled, and the scenario depiction is provided by the underlying 3D model. Thanks to the presence of an efficient multi-resolution structure the application exploits massive and highly detailed models for this purpose, providing a good and useful representation of the site also when no photograph is available from the current point of view. One the most important features required in such as context is a fast and simple interface to navigate in the 3D environment. Considering the users' skills and the time-critical situation, this interface must avoid making the user feel "lost" and must keep his focus on the real images taken from the field. Starting from an initial $current\ image$ the system places the camera on the related image viewport, and the user begins his navigation interacting with the touchscreen. Following the touch of the user on the screen, the possible nearest images according to the 6D metric defined at preprocessing time are highlighted (see Fig. 6). If the user is touching inside the current viewport itself the camera rotates only the view direction according to the movement, or zooms in to the current image if a double click is performed. Instead, if a double click is performed on an highlighted near image, the observer camera moves to new image viewport, which becomes the new current one. The $6D$ image descriptor driven the image proximity is defined by the equation (1),

$$\frac{\sum_1^n x_i f_i}{\sum_1^n f_i} \quad (1)$$

where the features f_i are: shot position, shot direction, time of shot, color distribution, spatial color layout, depth; the weights x_i are defined by the user at pre-processing time, according to the scenario requirements (see Section 5) and implicitly describing a metric for the image ordering. Considering the current image descriptor, all the other shots are ordered by increasing difference from this one and are rendered back-to-front in a dedicated, but hidden, OpenGL buffer. To identify the different shots in the 3D space we draw the projected viewports of the images as oriented quads in the same buffer, encoding the related id as a color. Moving the touch on the screen outside the current view results in probing this hidden frame buffer, returning the id of the touched viewport through its color value. Since several viewports could share the same pixel positions on the screen, the back-to-front method assure that the returned id is the one of nearest view to the current image. Unlike common 3D photo browsers, the navigation is not limited to the image viewports, but by keeping the touch pressed on the screen for few seconds is possible to "unlock" the aided navigation, allowing the user to then slide on the screen to freely move through the 3D point cloud. This feature is useful for specific parts of the site where no real photo is available, always having a representation of the scenario.

Figure 7: ***Navigation with a tablet in a training dataset*** *One the most important features required in such as context is a fast and simple interface to navigate in the 3D environment. Considering the users' skills and the time-critical situation, this interface must avoid making the user feel "lost" and must keep his focus on the real images taken from the field.*

5 Implementation and results

The $3DNSITE$ system has been released for Windows and Linux platforms and developed with C++, OpenGL and the Nokia Qt toolkit. The framework includes two modules: a pre-processor subsystem called *3Dnsite Generator* and a client module called *3Dnsite Viewer*.

5.1 Pre-processor module

The 3Dnsite Generator prepares the data to be stored in a server, offering a user interface in order to tune the data building process according to the specific requirements. The 3D models and the related aligned images can come from different sources. For instance, since active 3D range scanning technology is becoming a common resource, it is becoming quite common for companies to routinely laser scan their factories (e.g. power plants, complex pipelines, etc.). In addition, photographs provide a fast and easy way to document a site. On the other hand, cheaper passive methods also

Dataset	3D model Size MSamples	3D model Patches	3D model Time	Images Size Mpixels	Images Count	Images Time
Geomethane	7.5	463	2m22s	12	300	5m20s
Training building	0.5	30	10s	12	60	41s

Table 1: ***Dataset pre-processing stats.*** *The pre-processor transforms the original 3D model into a multi-resolution structure in order to handle datasets which exceed the capacity of the client GPU RAM and to enable network streaming. The measurements show that the time to build a new dataset is very short, allowing a trainer to quickly create new scenarios.*

are becoming quite popular. Thanks to recent computer vision advances in Structure from Motion (SfM), it is possible to extract more affordable, dense 3D samplings from large image datasets,

	hardware performances						Network		
User	Device	Multires fps	No multires fps	Connection	RAM MB	GPU MB	peak MB/s	startup s	average KB/s
Manager 1	whiteboard	172	16	ethernet	1024	256	11	0	700
Manager 2	laptop	70	9	wireless N	512	256	600	0	600
Manager 3	netbook	35	3	ethernet	512	256	3.6	3	400
Manager 4	netbook	27	3	ethernet	512	128	3.4	3	300
Agent 1	netbook	26	2	wireless N	512	128	0.600	6	220
Agent 2	tablet	26	1	wireless N	1024	256	0.593	6	470
Agent 3	tablet	24	1	wireless N	1024	128	0.535	8	260
Agent 4	tablet	23	1	wireless N	512	128	0.587	12	400
Agent 5	tablet	19	1	3G-HSDPA	1024	128	0.220	22	125
Agent 6	tablet	8	1	EDGE-GPRS	1024	128	0.050	34	36

Table 2: ***Client rendering performance.*** *For the performance measurements all the devices were set to a screen resolution of 1200x800 and used the Geomethane dataset (1.2 GB dataset based on 300 hi-res images). The average bandwidth is measured during navigation; at application startup there is a one-time bandwidth peak that depends on memory budget and images size. The support for the 3D multi-resolution system increases in importance when using portable devices, where no interaction is possible without it. The RAM MB and GPU MB columns show the memory budget for both system and GPU caches. For local applications these values can be set almost to the full memory capacity; however, this setting does not work well for a networked application, where a high GPU budget increases the average bandwidth required to fill the last cache stage. Due to the screen resolution employed (1200x800) the 3D embedded photos have been scaled to 5MPixels with variable JPEG compression rate (maximum 1:200).*

even those composed of heterogeneous pictures shot under uncontrolled conditions. Given a dense set of photographs, SfM algorithms produce medium quality colored point clouds, often sufficiently detailed for all operations that require location recognition. The way in which the 3D model is acquired also determines which techniques can be used to calibrate and align the images on the 3D model (e.g. [Pintus et al. 2011b]). When the images are used to compute a point-cloud 3D model, they are already calibrated and aligned to that model. The datasets presented here are acquired by a SfM pipeline, starting from a set of images and with the support of a GPS tracker for the spatial reference. To process them we have used a commodity desktop PC with an Intel Core2 Q6600 2.6GHz and 2GB ram. The first dataset presented in this paragraph is the gas storage site of Geomethane in Manosque (France). This site has been used as training emergency scenario involving the Operational Center of the Fire and Rescue Services of Alpes de Hautes,the local Gendarmerie and many real crisis managers and first responders (see Section 5.2). The 3D point cloud has 7.5Million samples and has been obtained from 300 12Mpixel photographs. As already discussed, the pre-processor transformed the original 3D model into a multi-resolution structure in order to handle datasets which exceed the capacity of the client GPU RAM and to enable network streaming (see Table 1). Due to the screen resolution employed (1200x800) the 3D embedded photos have been scaled to 5MPixels with variable JPEG compression rate (maximum 1:200). The images' metadata has been processed using an average depth for each image as reference plane. Different image depth calculations (available options are also minimum and maximum depth) result in different image projections at rendering time, whereas different feature weights result in different processing times and different behaviors of the user interface (see eq. 1). The 3Dnsite Generator lets the user to set up 6 different feature weights in a scale from 0 to 1: shot position, shot direction, time of shot, color distribution, spatial color layout and image depth. In the Geomethane dataset, the dominant values are shot position (0.8), shot direction (0.4) and image depth (0.3). The other features have been considered with values lower than 0.1. While the Geomethane scenario can be considered to be a type of site documentation, the second kind of data, proposed in Table 1, is a pure training scenario. In fact, one important application of these systems is to train agents on unseen sites, where they have to orient themselves and quickly find strategical locations. The pre-processing measurements show that the time to build a new dataset of this type is very short, thus allowing the trainer to quickly create new scenarios. Due to the spatially limited area the dominant values for the metadata creation are: shot direction (0.9), shot position (0.2) and image depth (0.4). Notably these training datasets are typically small places, but characterized by high-resolution photos with a large amount of data to stream.

5.2 Client module

The client setup tested has been developed essentially for two user profiles: a manager profile and an agent profile. The manager profile follows the operations from the operational headquarters using an electronic whiteboard connected to a commodity desktop (Fig. 1, top left) or from the field command post using a laptop (Fig. 1 bottom left), employing 3DNSITE to plan detailed operations and give orders to the agents. On the other hand, the agent profile has to perform specific tasks in several strategic locations, assuming he has never been in that place before. Many users from these two profiles have tested the system together with real trainers, driving the development of the navigation metaphor. Both profiles have found the interface comfortable and intuitive, and the agents in particular quickly familiarized themselves with the application thanks to its similarity with popular web-based 3D map navigators. The hardware employed for the test was: a commodity desktop PC with an Intel Core2 E6600 2.4GHz, 2GB RAM and an NVidia GeForce GTX560, connected to a 40-inch multitouch whiteboard; an Alienware M17xR3 laptop with an Intel Core i7 Processor 2630QM, 6GB RAM and an NVidia GeForce GTX460M; a Compag mini netbook with an Intel Atom processor and an NVidia ION GPU; an Acer Iconia 500 tablet with AMD Fusion C-60 and Radeon HD6290. The screen resolution on all the devices was set to 1200x800 and the Geomethane dataset was used to measure performance. As can be seen in Table 2, the support for the 3D multi-resolution system increases in importance when using portable devices, where no interaction is possible without it. Simultaneously, the multi-level cache scales well over the hardware computing power and network bandwidth, assuring the interactivity of the application even with 3G mobile bandwidth—a considerable feat especially considering the size and detail of the images (eg. 5Mpixels). The RAM MB and GPU MB columns show the memory budget for both system and GPU caches. For local applications these values can be set almost to the full memory capacity; however, this setting does not work well for a networked application, where a high GPU budget increases the average bandwidth required to fill the last cache stage. Therefore, a trade-off between memory setup and network capabilities is required. The most important difference between devices (and their available network bandwidth) is not in terms of interactivity, which is well supported by the scalability of the system, but in the terms of time needed to load the cache at the application startup. In this phase, the network bandwidth has the peak shown in the *Network bandwidth* column of Table 2; while in desktop and laptop cases the startup time is almost instantaneous, the time needed increases with the reduction of hardware and network capabilities. This latency time is completely led by the images download, despite the view-dependent adaptive cache control the minimum quantity to download to see an Hi-res image is the size of the photo itself (sometimes over 1MB). On the other side the 3D point cloud, due to the atomic nature of its elements, can be adapted more easily. The network setup on the Geomethane site was a WiFi-N provided by a mobile antenna in the fire truck command post. Although it is becoming quite common to have a high-speed wireless connection in crisis and emergency contexts, we have performed several tests using 3G and EDGE connections, clearly experiencing a loss of performances, but nevertheless preserving good interactivity.

6 Conclusions and future work

We have presented a web-based system to interactively navigate a complex 3D environment during the evolution/simulation of a crisis. This tool, called 3DNSITE, has been designed and developed to share and visualize three-dimensional hybrid data, integrating into a more general training and decision framework for emergency operations. Tested inside this main framework, 3DNSITE has been demonstrated to satisfy some important requirements for operation in crisis contexts. In particular, it achieves scalability over limited network and hardware resources while preserving a good interactivity. Although the individual methods employed have already been presented in computer graphics literature, their combination, enhancement and application in this peculiar web-based environment has resulted in very positive performance measurements and user experience. Scalability over network and hardware resources is achieved by the combination of a priority-based, multi-level cache system and a multi-resolution, dynamic, hierarchical representation of the 3D model. This output-sensitive approach for the 3D model enables best performance in host-to-graphics and network communications, and provides a good approximation of the scenario even with a when no real image is available from the specific point of view. Despite the fact that the models employed for the tests cannot be considered very large massive models in the customary realm of computer graphics, they are large enough to limit the interaction in remote operations supported by limited hardware and network resources, as show in Section 5.2. The strong network scalability characteristics of the multi-resolution method proposed is also promising for future enhancements and applications, since the 3D models are continuously increasing in size and complexity. The same view-dependent/output-sensitive philosophy is applied to the high-resolution image cache controller with the aid of precomputed 6D image descriptors, supporting a fast and efficient navigation interface. The opportunity to update the scenario according to the real or simulated evolution of the crisis is now supported within the limit of a few images. However, current work is focussed on obtaining a better dynamic and progressive dataset update for both the images and the 3D model.

Acknowledgments. This research is partially supported by EU FP7 grant 242341 (INDIGO).

References

BETTIO, F., GOBBETTI, E., MARTON, F., AND PINTORE, G. 2007. High-quality networked terrain rendering from compressed bitstreams. In *Proc. ACM Web3D International Symposium*, New York, NY, USA, ACM Press, 37–44.

BOIN, A. 2009. The new world of crises and crisis management: Implications for policymaking and research. *Review of Policy Research 26*, 4, 367–377.

BRIVIO, P., TARINI, M., CIGNONI, P., AND SCOPIGNO, R. 2012. Joint interactive visualization of 3D models and pictures in walkable scenes. In *Eurographics 2012 - Posters*, Eurographics Association.

CAPEL, D. 2005. An effective bail-out test for ransac consensus scoring. In *BMVC British Machine Vision Conf*, 629–638.

CARVER, L., AND TUROFF, M. 2007. Human-computer interaction: the human and computer as a team in emergency management information systems. *Commun. ACM 50*, 3 (Mar.), 33–38.

CHUM, O., MATAS, J., AND KITTLER, J. 2003. Locally optimized ransac. In *DAGM-Symposium*, 236–243.

CIGNONI, P., GANOVELLI, F., GOBBETTI, E., MARTON, F., PONCHIO, F., AND SCOPIGNO, R. 2004. Adaptive tetrapuzzles: efficient out-of-core construction and visualization of gigantic multiresolution polygonal models. *ACM Trans. Graph. 23*, 3, 796–803.

CIGNONI, P., GANOVELLI, F., GOBBETTI, E., MARTON, F., PONCHIO, F., AND SCOPIGNO, R. 2005. Batched multi triangulation. In *Proc. IEEE Visualization*, 207–214.

DUGDALE J., BELLAMINE-BEN SAOUD N., P. B., AND N., P. 2010. Simulation and emergency management. *Information Systems for Emergency Management*, 229–253.

FISCHLER, M. A., AND BOLLES, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (June), 381–395.

GOBBETTI, E., AND MARTON, F. 2004. Layered point clouds: a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models. *Computers & Graphics 28*, 6 (Dec.), 815–826.

GOESELE, M., ACKERMANN, J., FUHRMANN, S., HAUBOLD, C., KLOWSKY, R., STEEDLY, D., AND SZELISKI, R. 2010. Ambient point clouds for view interpolation. In *ACM SIGGRAPH 2010 papers*, ACM, New York, NY, USA, SIGGRAPH '10, 95:1–95:6.

KASIK, D., DIETRICH, A., GOBBETTI, E., MARTON, F., MANOCHA, D., SLUSALLEK, P., STEPHENS, A., AND YOON, S.-E. 2008. Massive model visualization techniques: course notes. In *ACM SIGGRAPH 2008 classes*, ACM, New York, NY, USA, SIGGRAPH '08, 40:1–40:188.

KOPF, J., CHEN, B., SZELISKI, R., AND COHEN, M. 2010. Street slide: browsing street level imagery. In *ACM SIGGRAPH 2010 papers*, ACM, New York, NY, USA, SIGGRAPH '10, 96:1–96:8.

LANFRANCHI, V., AND IRESON, N. 2009. User requirements for a collective intelligence emergency response system. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, British Computer Society, Swinton, UK, UK, BCS-HCI '09, 198–203.

LIPPMAN, A. 1980. Movie-maps: An application of the optical videodisc to computer graphics. *SIGGRAPH Comput. Graph. 14* (July), 32–42.

MARVIE, J.-E., GAUTRON, P., LECOCQ, P., MOCQUARD, O., AND GÉRARD, F. 2011. Streaming and synchronization of multi-user worlds through http/1.1. In *Proceedings of the 16th International Conference on 3D Web Technology*, ACM, New York, NY, USA, Web3D '11, 111–120.

MICROSOFT, 2007. Photosynth. `http://photosynth.net`.

MOUTON, C., SONS, K., AND GRIMSTEAD, I. 2011. Collaborative visualization: current systems and future trends. In *Proceedings of the 16th International Conference on 3D Web Technology*, ACM, New York, NY, USA, Web3D '11, 101–110.

PALEN, L., HILTZ, S. R., AND LIU, S. B. 2007. Online forums supporting grassroots participation in emergency preparedness and response. *Commun. ACM 50*, 3 (Mar.), 54–58.

PINTUS, R., GOBBETTI, E., AND CALLIERI, M. 2011. Fast low-memory seamless photo blending on massive point clouds using a streaming framework. *J. Comput. Cult. Herit. 4*, 2 (Nov.), 6:1–6:15.

PINTUS, R., GOBBETTI, E., AND COMBET, R. 2011. Fast and robust semi-automatic registration of photographs to 3d geometry. In *The 12th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, 9–16.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3d. In *ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, SIGGRAPH '06, 835–846.

SNAVELY, N., GARG, R., SEITZ, S. M., AND SZELISKI, R. 2008. Finding paths through the world's photos. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, SIGGRAPH '08, 15:1–15:11.

TORR, P. H. S., AND ZISSERMAN, A. 2000. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding 78*, 138–156.

TUITE, K., SNAVELY, N., HSIAO, D.-Y., TABING, N., AND POPOVIC, Z. 2011. Photocity: training experts at large-scale image acquisition through a competitive game. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM, New York, NY, USA, CHI '11, 1383–1392.

UN, 2009. United nations population division world urbanization prospects (the 2009 revision).

VINCENT, L. 2007. Taking online maps down to street level. *Computer 40* (December), 118–120.

WAN, G., SNAVELY, N., COHEN-OR, D., ZHENG, Q., CHEN, B., AND LI, S. 2012. Sorting unorganized photo sets for urban reconstruction. *Graph. Models 74*, 1 (Jan.), 14–28.

YOON, S., SALOMON, B., GAYLE, R., AND MANOCHA, D. 2004. Quick-vdr: Interactive view-dependent rendering of massive models. In *Visualization, 2004. IEEE*, IEEE, 131–138.