



**INDIGO**  
Crisis Management Solutions

**3DNSITE SERVER**



---

## Content

<b>CONTENT.....</b>	<b>2</b>
<b>1INTEGRATION INTO THE INDIGO SYSTEM.....</b>	<b>3</b>
<b>2PRE-REQUISITES.....</b>	<b>3</b>
<b>33D MODEL RECONSTRUCTION.....</b>	<b>3</b>
<b>4IMAGES AND 2D TO 3D CALIBRATION.....</b>	<b>3</b>
<b>5DATASET BUILDING.....</b>	<b>3</b>
<b>6INSTALLATION AND RUN.....</b>	<b>4</b>
6.1 Installation.....	4
6.2 Run the application .....	4
6.3 Manage a dataset processing .....	4
6.4 Output and Alignment .....	6

## 1 Integration into the Indigo system

The 3DNSITE system is a client/server 3D visualization tool to handle, stream and visualize large three-dimensional hybrid data (complex 3D model with 3D aligned images).

The specific data is prepared usually before the crisis happens through the **3DNSITE Generator** tool and usually contains important details not available in the main 3D map of the INDIGO system.

To acquire the image collections no specific hardware is required (eg. Differently from Google Street View – mobile stations with coupled cameras and depth sensors or omnidirectional video cameras with tracking systems) but photos taken with generic cameras can be included, as well as images coming from different sources (archives, etc.).

## 2 Pre-requisites

We assume that a three dimensional description of the site is available **before** the accident and those models are reconstructed from various means.

We assume that we can have access:

- to a 3D model, represented as a point cloud, similar to those that can be acquired with 3d scanning (or that can be generated from other means )
- to a set of photographs aligned within the same 3D reference (systems such as Photosynth, Arc3d, Bundler, etc. can be used for the alignment)

Alternatively the **3DNSITE Generator** can directly generate the 3D model and the related aligned images/viewports from photographs exploiting an embedded SfM pipeline embedded in.

## 3 3D model reconstruction

The 3D model can come in any form, as long as it can be rendered in a depth-coherent way. Specifically, the system supports point clouds, triangle soups, and indexed triangle meshes. 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.

## 4 Images and 2D to 3D calibration

The **3DNSITE GENERATOR** assumes that the photographs are calibrated with the associated 3D model.

When the images are used to compute a point-cloud 3D model, they are already calibrated and aligned to that model.

If the 3D model and the photographs are produced by separate acquisition pipelines (e.g. a photographic campaign and laser scanning session), we have to align the pictures with the 3D model as a separate task.

## 5 Dataset building

A polygonal or point cloud 3D model and a calibrated image set are given as input.

In the pre-processing phase, the 3D model is converted to a multi-resolution format, then image thumbnails, minimum depths, ordering and semantic distances are computed.

Given the input datasets composed of a 3D model and a calibrated image collection, the pre-processor is run over it and the results are stored on the server.

The pre-processing consists in data formatting and the computation of various metadata which will be used at rendering time. These include construction of thumbnails from images, compression of hi-res images in JPEG format, and the construction of the 3D multi-resolution structure. Additionally, for each image, the minimal image-space depth of its content is computed 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 computed.

An abstract descriptor is associated to each image in order to estimate good orderings and the semantic distances among images.

## 6 Installation and run

The beta-release of the code consists of a generator of a 3DNSITE dataset ready for browsing.

A simple GUI meant to demonstrate and test the algorithms will be provided.

### 6.1 Installation

The application is provided as Windows *.msi* installation wizard.

### 6.2 Run the application

Launch the **3DNSITE generator** and load the 3D model, the calibrated images and the related cameras using the GUI (see Fig. 1). Importers from Photosynth (url), Bundler (.out) and 3DNSITE format (.pxi) are available from the **File** menu.

### 6.3 Manage a dataset processing

During the data processing, an image is associated to various high-level descriptors. These are used for defining total orderings over the images, and distances between consecutive images (for each ordering). The distance between images will be used in the viewer to manage the transition between one point of view and the next and the proximity of thumbnails in the bottom image bar.

Using the *Parameters computation* tab (see Fig. 2) the 3DNSITE generator can extract and use the following image descriptors: time-of-shot (extracted from the file Exif), image shot position and orientation (i.e. a 3D translation vector and a 2D rotation vector), as given by the calibration, color distribution, as a 16 entry histogram of colors, spatialcolor-layout, as the 4x4 down-sampled image (color computations are performed in LAB color-space) and the depth, the average distance from the point of view of each photograph and the corresponding 3D points.

Launching the processing an index file is created containing references to the model and the images, associating them with their depth, order, and semantic information.

An utility to load a geo-referenced map or aerial photograph to align the hybrid data is available from the *Output index (pxi)* tab (see Fig. 3) pushing the button **Geo Utilities**.

File						
Data overview Parameters computation Optional settings Output index (pxi)						
	Hires images	Lowres thumbnails	Shots	Depths	Order	Semantic
1	C:/Programmi/VC...		<vcgCamera Rot...			
2	C:/Programmi/VC...		<vcgCamera Rot...			
3	C:/Programmi/VC...		<vcgCamera Rot...			
4	C:/Programmi/VC...		<vcgCamera Rot...			
5	C:/Programmi/VC...		<vcgCamera Rot...			
6	C:/Programmi/VC...		<vcgCamera Rot...			
7	C:/Programmi/VC...		<vcgCamera Rot...			
8	C:/Programmi/VC...		<vcgCamera Rot...			
9	C:/Programmi/VC...		<vcgCamera Rot...			
10	C:/Programmi/VC...		<vcgCamera Rot...			
11	C:/Programmi/VC...		<vcgCamera Rot...			
12	C:/Programmi/VC...		<vcgCamera Rot...			
13	C:/Programmi/VC...		<vcgCamera Rot...			
14	C:/Programmi/VC...		<vcgCamera Rot...			
15	C:/Programmi/VC...		<vcgCamera Rot...			
16	C:/Programmi/VC...		<vcgCamera Rot...			

3D model

Figure 1: Data overview tab

File		
Data overview Parameters computation Optional settings Output index (pxi)		
<div> <div> <b>Thumbnails</b> <div> <input type="radio"/> do not compute           <input type="radio"/> use same of hires           <input checked="" type="radio"/> downsample hires         </div> </div> <div> <b>Depth</b> <div> <input type="radio"/> do not compute           <input checked="" type="radio"/> compute from 3D model <input type="text" value="C:/Programmi/VCG/"/> <input type="button" value="..."/> <div> <input type="checkbox"/> minimum           <input type="checkbox"/> maximum           <input checked="" type="checkbox"/> average         </div> </div> </div> </div>		
<div> <div> <b>Order</b> <div> <div>time of shot</div> <div>direction of shot</div> <div>position of shot</div> </div> </div> <div> <div>depth of shot</div> <div>color distribution</div> <div>color spatial distribution</div> </div> </div>		
<div> <div> <b>Semantic</b> <div> <div>time of shot</div> <div>direction of shot</div> <div>position of shot</div> </div> </div> <div> <div>depth of shot</div> <div>color distribution</div> <div>color spatial distribution</div> </div> </div>		
<div> <div>Features computation</div> <div> <div></div> <div>0%</div> </div> <div> <input type="button" value="Compute all"/> </div> </div>		

Figure 2: Parameters computation tab

## 6.4 Output and Alignment

The last tab contains the commands to output the PXI file (that will be loaded by the 3DNsite Viewer and the commands to add a new picture to an existing PXI).

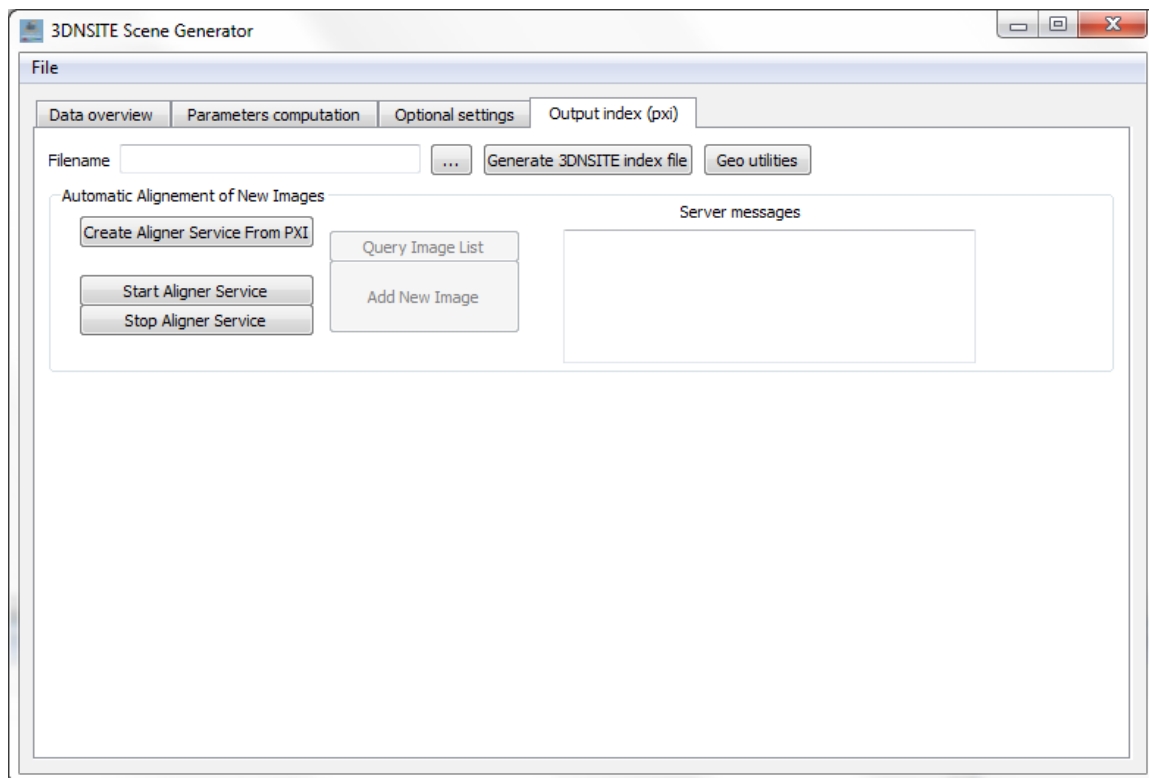
The automatic alignment of a new image needs an ad hoc data structure which is computed from the PXI dataset by clicking on “Create Aligner From PXI” button. This operation needs to be done only once for each new PXI dataset and it takes an amount of time proportional to the number of images.

The data consists of:

- An .xml file for each image, named [image\_name].xml containing the data of the camera
- A .sift file for each image, named [image\_name].sift containing the sift descriptors of the image

With off-the-shelf personal computers this time sums up to a few seconds (from 5 to 10) per image. The progress of the construction can be monitored in the Server Messages text area. When the creation is terminated the aligner service can be started by clicking the corresponding button “Start Aligner Service”. When the service starts, it loads the data computed at the creation and set itself on a listening state, where it waits for new images to be aligned. Again, this operation takes a time proportional to the number of images. When the service has started and is running, the buttons “Query Image List” and “Add New Image” become enabled. “Query Image List” simply asks the server the list of names of images that are currently stored.

The “Add New Image” opens a Dialog to choose the image the user wants to add to the dataset and launch the alignment process. The success of the automatic alignment of a new image depends on how many similar images are present in the database. If the algorithm succeeds, the PXI file is rewritten including the new image. The result is communicated in the Server Messages text area.



**Figure 3: Dataset Generation and Updating**