



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Benchmark Imagery for Assessing Geospatial Semantic Content Extraction Algorithms

Final Report for LLNL and LANL Joint Tasks
LL11-GS-PD3SE / LA11-GS- PD3SE

*W. T. White III,¹ P. A. Pope,² J. W. Goforth,¹
L. R. Gaines,² L. Prasad,² S. M. Hojnacki,¹
D. M. Ranken,² and R. S. Roberts¹*

¹Lawrence Livermore National Laboratory (LLNL)

²Los Alamos National Laboratory (LANL)

15 October 2013

Submitted to:

The National Nuclear Security Agency (NNSA)

Office of Proliferation Detection R&D, NA-22

James Peltz, Program Manager

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor Los Alamos National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC, or Los Alamos National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government, Lawrence Livermore National Security, LLC, or Los Alamos National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and by Los Alamos National Laboratory under Contract DE-AC52-06NA25396.

30 September 2013

The authors wish to acknowledge funding and support from the Office of Nuclear Nonproliferation Research and Development, NA-22, in the National Nuclear Security Agency (NNSA) of the Department of Energy (DOE). Without the forward-looking support from NA-22 — particularly from Dr. David LaGraffe, Mr. James Peltz and Dr. Alexander Slepoy — this project would not have been possible.

It is our hope that the Benchmark Imagery Suite will be useful to algorithm developers and that users improve it so that it becomes a standard tool for testing and evaluating software to interpret overhead images of industrial facilities. Users with questions or comments are encouraged to contact either of the principal investigators. We welcome your remarks.

Paul A. Pope, PhD
Principal Investigator
Los Alamos National Laboratory
papope@lanl.gov
505-667-9135

W. Travis White, III, PhD
Principal Investigator
Lawrence Livermore National Laboratory
white23@llnl.gov
925-423-1067

[THIS PAGE INTENTIONALLY LEFT BLANK.]

Table of Contents

Preface	i
Cover letter.....	ii
Table of Contents	iii
List of Tables	iv
Table of Figures	v
Executive Summary	1
The Benchmark Imagery Project	1
Benchmark Imagery Suite v1.1a.....	1
Path Forward.....	2
Organization of the Report.....	2
I. Introduction.....	3
II. Initial conditions: Industrial Classes, Facilities and Labels.....	5
Selection of Industrial Classes	5
Choice of Facilities	6
Constrained Vocabulary.....	6
III. Real Images.....	7
Requirements and Scope of Imagery	7
Acquiring Real Images	8
Example Images.....	8
IV. Lidar Images	14
Obtaining Lidar Data	15
Processing the Lidar Data	16
V. Synthetic Images.....	17
Model-building	19
Using Commercial 3-D Components.....	20
Creating 3-D Components from 2-D Images.....	20
Base Terrain Preparation.....	21
Initial Facility Layout	21
Image Generation.....	23
Automating the Layout Process	23
B*-Tree Floorplanner.....	24
KML Generator.....	24

Results of Using an Automated Layout Tool	24
VI. Ground Truth	25
Real Images: Selective Point Annotation (All Object Classes)	25
Real Images: Exhaustive Point Annotation (Buildings Only)	26
Real Images: Exhaustive Areal Segmentation (All Object Classes) — Posterization.....	27
Synthetic Images: Exhaustive Areal Segmentation (All Object Classes) — Posterization.....	29
VII. Demonstration of Testing an Algorithm with Benchmark Imagery	32
Define V&V Methodology.	34
Obtain algorithm to test.	35
Train & Test algorithm using the Benchmark Imagery Suite.....	36
Interpret Results from the V&V Tests	41
Remarks and Suggestions about the Benchmark Imagery Suite	42
VIII. Conclusion	43
Summary of Work Accomplished	43
Lessons Learned.....	43
A Path Forward.....	43
IX. References.....	45
X. Appendix A — Organization of the Benchmark Imagery Suite v1.1a	48
XI. Appendix B — Photointerpretation Key.....	53
General Guidance.....	53

List of Tables

Table 1. <i>Chisnell and Cole’s Industrial Classes</i>	5
Table 2. <i>Constrained vocabulary of labels in the Benchmark Imagery Suite</i>	7
Table 3. <i>Aerial Images Included in the Benchmark Imagery Suite v1.1a</i>	13
Table 4. <i>List of facilities in Benchmark Imagery Suite v1.1 for which lidar data are included</i> .	14
Table 5. <i>List of 12 real-image files with exhaustive point-wise annotation</i>	27
Table 6. <i>List of 50 real-image files with exhaustive areal segmentation/annotation (“Posterized”)</i>	28
Table 7. <i>Color lookup table for the constrained vocabulary used to posterize real images</i>	28
Table 8. <i>Color look-up table for the posterized image shown above in Figure 22</i>	32
Table 9. <i>List of real image names (.tif) and their corresponding point annotation files (.csv) used for training the BldgFinder algorithm (where Bldg = “Building”)</i>	38
Table 10. <i>Performance table results of applying the BldgFinder algorithm (where Bldg = “Building”), trained using the information listed in Table 9, to the real images listed under “Name,” with their associated sparse point annotations serving as ground truth</i>	39
Table 11. <i>List of real image names (.tif) and their corresponding exhaustive point annotation files (.csv) used for training the BldgFinder algorithm (where Bldg = “Building”)</i>	39
Table 12. <i>Performance table results of applying the BldgFinder algorithm (where Bldg = “Building”), trained using the information listed in Table 11, to the real images listed under “Name,” with their associated exhaustive point annotations serving as ground truth</i>	39
Table 13. <i>List of real image names (.tif) corresponding with the areal (polygonal) annotation files (_annotations.shp) used for training the BldgFinder algorithm (where Bldg = “Building” and “Industrial Building”)</i>	40
Table 14. <i>Performance table results of applying the BldgFinder algorithm (where Bldg = “Building” and “Industrial Building”), trained using the information listed in Table 13, to the real images listed under “Name,” with their associated posterization polygons serving as ground truth</i>	40
Table 15. <i>Performance table results of applying the BldgFinder algorithm (where Bldg = 'bldg', 'hangar', 'hangarsmall', 'reactor', 'reactorbldg', 'smallbldg', and 'stackbldg'), trained using the information listed in Table 13, to the “two types four sites” synthetic images listed under “Name,” with their associated posterization polygons serving as ground truth</i>	40
Table 16. <i>Performance table results of applying the BldgFinder algorithm (where Bldg = 'bldg', 'hangar', 'hangarsmall', 'reactor', 'reactorbldg', 'smallbldg', and 'stackbldg'), trained using the information listed in Table 13, to the “arid_nuc” synthetic images listed under “Name,” with their associated posterization polygons serving as ground truth</i>	41

List of Figures

Figure 1. <i>Key terminology used in this report.</i>	3
Figure 2. <i>Simple schematic illustration of an iterative approach to assisting a subject matter expert by extracting semantics from imagery.</i>	3
Figure 3. <i>Illustration of a heat processing facility, HP0058_C8, GSD: 0.3m.</i>	9
Figure 4. <i>Illustration of a chemical processing facility, CP28_VDE, GSD 0.5m.</i>	10
Figure 5. <i>Illustration of a heavy manufacturing facility, HM45_V3F, GSD: 0.3m.</i>	10
Figure 6. <i>Example of a mechanical processing facility, MP032_VBQ; GSD: 0.3m.</i>	11
Figure 7. <i>Example of a semiconductor industry facility, SI0018_U4F; FSD = 0.15m.</i>	12
Figure 8. <i>Example of a rasterized lidar image. Site CP2_U3Z. Left: Elevations are displayed as gray scale values, with black being lowest and white highest. Right: Rasterized lidar image overlaid on the corresponding GeoTIFF Image.</i>	14
Figure 9. <i>Graphical User Interface for the USGS’s EarthExplorer showing an area of interest for site CP2_U3Z (red square) superimposed upon an image of the surrounding region.</i>	15
Figure 10. <i>Graphical User Interface for the USGS’s EarthExplorer showing that an area of interest (orange square) straddles two separate tiles of USGS data (yellow and green boxes).</i>	15
Figure 11. <i>Synthetic image of a coal-fired power plant in a coastal environment.</i>	19
Figure 12. <i>Synthetic image of a nuclear power plant in an arid environment.</i>	19
Figure 13. <i>Coal-fired power plant, HP52.</i>	22
Figure 14. <i>Image of an actual nuclear plant, HP104.</i>	22
Figure 15. <i>Image of an actual nuclear plant, HP119.</i>	22
Figure 16. <i>Portion of an annotated image showing the labels superimposed on the scene, Heavy Manufacturing Plant, HM5.</i>	26
Figure 17. <i>Example of a portion of an exhaustively annotated image (buildings only), HM2_VA6.</i>	27
Figure 18. <i>Posterized image of site CP2_U3Z. (See Fig 8.)</i>	28
Figure 19. <i>Shape file representation of a posterized real image, CP2_U3Z.</i>	29
Figure 20. <i>Example of an object, “Hangar-2” with multiple layers of abstraction in its name, increasing in specificity from top to bottom.</i>	30
Figure 21. <i>A synthetic image of a nuclear power plant in an arid environment.</i>	31
Figure 22. <i>Posterized version of Figure 21.</i>	31
Figure 23. <i>Vectorized version of Figure 22.</i>	31
Figure 24. <i>Algorithm verification and validation cycle proposed by the NA-22 GSV Working Group.</i>	34
Figure 25. <i>Illustration of comparison of BldgFinder candidate polygons to ground truth (in this case, point annotations) to create an error matrix.</i>	36

Benchmark Imagery for Assessing Geospatial Semantic Content Extraction Algorithms — Final Report

W. T. White III,¹ P. A. Pope,² J. W. Goforth,¹ L. R. Gaines,² L. Prasad,² S. M. Hojnacki,¹
D. M. Ranken,² and R. S. Roberts¹

15 October 2013

¹Lawrence Livermore National Laboratory (LLNL)

²Los Alamos National Laboratory (LANL)

Executive Summary

THE BENCHMARK IMAGERY PROJECT

Benchmark Imagery for Assessing Geospatial Semantic Content Extraction Algorithms (the “Benchmark Imagery Project”) was a three-year endeavor funded by the Enabling Capabilities (EC) Team of DOE/NNSA’s Office of Proliferation Detection R&D, NA-22, and carried out through a collaborative effort between Los Alamos National Laboratory (LANL) and the Lawrence Livermore National Laboratory (LLNL). The primary objective was to create a set of benchmark imagery for validating algorithms designed to extract semantic content from geospatial images of industrial facilities. The goal was not to develop algorithms; rather, it was to provide information that could be used to do so. The secondary objective was to demonstrate how this information could be used for algorithm validation.

This report summarizes the results of the Benchmark Imagery Project.

BENCHMARK IMAGERY SUITE v1.1A

The set of imagery, metadata and ancillary data collected under the project constitute the Benchmark Imagery Suite v1.1a. The suite includes aerial visible-band images of 190 US industrial facilities, synthetic images of 38 fictitious electrical power plants and lidar elevation rasters of 16 of the real facilities. The classes of industries in the suite include the following:

- Chemical Processing (27 facilities) — *e.g.*, oil refineries.
- Heavy Manufacturing (31 facilities) — *e.g.*, steel foundries and metal-casting plants.
- Heat Processing (37 facilities) — *e.g.*, nuclear and coal-fired electrical power plants.
- Mechanical Processing (83 facilities) — *e.g.*, aluminum processing plants and water treatment plants.
- Semiconductor Industry (12 facilities) — *e.g.*, computer chip companies.

The related data in the suite include the geographical locations and industry types of each facility. Spectrally, most of the images are three-band red-green-blue (RGB); however, some also include a fourth, near-infrared band. Spatially, each image covers an area of approximately 1 to 1.6 km², with a ground sample distance (GSD) of 0.3 to 1m. All images are orthorectified GeoTIFF [1] images taken at a near-nadir viewing angle.

Ancillary data in the Benchmark Imagery Suite v1.1a are intended to facilitate using the data in algorithm testing. They include object-annotation files for all 190 of the aerial photographs, and

posterized versions of all of the synthetic images and 51 of the aerial images. (A posterized image is a color-coded silhouette that identifies key objects in the scene.) The real images are organized by industry type. In the top-level directory for each industry type is a catalog of the sites therein plus a KML file that interfaces to Google Earth to show the user where the various facilities of that type are. The catalog is a browser-compatible (.html) file that lists key parameters of each image, such as spatial sampling parameters, image size, and number of spectral bands and date of acquisition.

All of the aerial GeoTIFF images in the suite are openly available and were downloaded through the USGS web site (<http://www.usgs.gov/pubprod/aerial.html#aerial>). The choice to use USGS data was driven by a goal of having openly distributable images, free of classification or licensing issues. The Benchmark Imagery Suite v1.1a is unclassified but is for OFFICIAL USE ONLY. To obtain a copy of the data, interested parties should contact

Mr. James Peltz
Program Manager
james.peltz@nnsa.doe.gov
202-586-7564

PATH FORWARD

The Benchmark Imagery Suite v1.1a is only one step towards compiling a comprehensive set of imagery, metadata, test methods and test results for algorithms that infer meaning from overhead geospatial imagery. The next major phase of its life cycle is for others to start using and revising it in a controlled way. Opportunities to expand upon the existing suite include both (a) refining the existing data (such as with more exhaustive annotation, more posterized images, more lidar imagery, or more documented tests of algorithms) and (b) broadening the scope of the imagery in the suite (such as with more industry classes, increased lexicon, more imaging modalities, more test metrics, or sites with multiple over flights). It is hoped that others will use the suite and improve it so that it becomes a standard benchmarking tool and inspires the creation of effective new algorithms.

ORGANIZATION OF THE REPORT

The body of this report has 8 sections: Background, Initial Conditions: Industrial Classes, Facilities and Labels, Real Images, Lidar Images, Synthetic Images, Ground Truth, Demonstration of Testing an Algorithm with Benchmark Imagery, and Conclusion. Two appendices discuss the organization of the suite and provide criteria used for annotating the imagery.

I. Introduction

An important portion of nuclear non-proliferation information is derived from overhead imagery. As the quantity and quality of overhead imagery grows, the data stream is eventually going to exceed unaided human capacity to handle. Since image interpretation ultimately is word-based in nature, tools that automatically label and interpret images in words are needed (Figure 1 and Figure 2). In parallel, testing standards must be developed for these emerging tools, including suites of test images and methodologies for using the test images. In short, new tools for the automatic extraction of semantic content from geospatial imagery of industrial facilities are needed to better support NA-22's mission in nuclear non-proliferation [2]. Collecting an initial suite of test images and demonstrating their utility are the underlying *raison d'être* for — and goal of — this project.

DEFINITION GEOSPATIAL SEMANTIC CONTENT EXTRACTION

“Inference (interpretation) of the implied significance of objects or activities from images that include information about their layout and location”

Geospatial (adj.): of or relating to the absolute or relative position of things on the Earth's surface.

Semantic (adj.): of, pertaining to, or arising from, the different meaning of words or other symbols.

Figure 1. Key terminology used in this report.

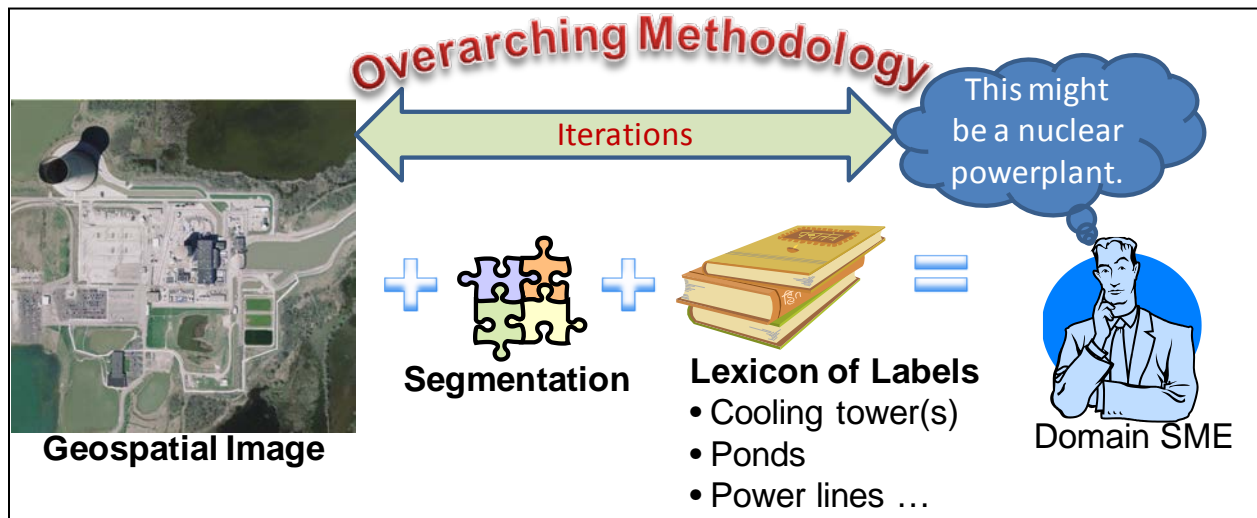


Figure 2. Simple schematic illustration of an iterative approach to assisting a subject matter expert by extracting semantics from imagery.

The actual process of automatic image interpretation, illustrated by the simple cartoon in Figure 2, is non-trivial. The overarching methodology must contain lists of rules, objects, and relationships among those objects — an ontology [3] — that can be mapped onto the input image(s). The components of the ontology may or may not be well specified or complete. For example, what are the rules that enable a human analyst to determine the boundaries of a facility

from looking only at an image or set of images? The lexicon of labels that a human image analyst would use (pond, tower, pile, ...) is often different than the labels that an image-processing algorithm would use (polygon, edge, point, ...). How should one design a segmentation algorithm to align with the lexicon of labels that a human would assign to objects in the scene? How does the vocabulary of labels limit possible interpretations of the image? What range of features must the geospatial image include for an algorithm to function? These are but a few of the issues that arise and that are active areas of computer vision research as applied to automated interpretation of overhead imagery.

From the preceding discussion, one can easily see that algorithms that support automated image interpretation may apply only over a narrow range of conditions. When applied to images that do not meet those conditions, the algorithms break. One desired property of a suite of benchmark imagery is that it should contain enough variety to test algorithms to their breaking points. This would help users understand the range of validity of the algorithms. Furthermore, it seems intuitively correct that the training set and the test set of images should be different when evaluating an algorithm.

An implicit assumption in making a benchmark imagery suite is that its existence will stimulate creation of new algorithms. This is based on historical precedents associated with land-based terrestrial imagery following the creation of other computer-vision data bases such as the Cal Tech 101 and 256 data bases [4]. It is the Field of Dreams concept: "Build it, and they will come." [5] To stimulate as many new algorithms as possible, the suite should be available to academics, because much of the creativity in geospatial image processing is concentrated within the academic community. This leads to an indirect requirement for the suite that it be affordable (free) and widely distributable (unclassified and unlicensed).

Another key issue is what should be the balance among actual photographs and synthetic images in the Benchmark Suite? Image analysts work with actual photographs, but algorithm developers can benefit from using real, synthetic and composite images. Each type of image has its own advantages. Real images automatically incorporate all the physics of the scene and all the engineering characteristics of sensors. Composite and synthetic images enable the content in a scene to be entirely configurable and the ground truth to be perfectly known. Creating facilities, objects, and environment in a synthetic image can be dictated by end users and their needs. This is very important when testing algorithms on situations where imagery is limited or where control is needed over the scene. Facilities can be generated in different stages of construction. Images can be "captured" at different angles and different time of day. Effects such as fog, clouds, and lighting can be simulated. Sensor and mission characteristics such as over-exposure, distortion, blur, and a variety of viewing geometries can be modeled. Also, synthetic images may be more distributable than real images, because they tend to be less sensitive in nature (unclassified). Therefore, the Benchmark Imagery Suite should contain a mix of real and composite/synthetic images.

The work in this project was accomplished iteratively among the following sequence of steps:

1. Select the types of industrial facilities, the vocabulary, the rules
2. Select the Real Images
3. Create the Synthetic/Composite Images
4. Collect the Lidar data
5. Create the Ground Truth Set
6. Demonstrate Algorithm Testing with the Benchmark Imagery Suite

II. Initial conditions: Industrial Classes, Facilities and Labels

In a semantic process, one must define terms up front [6]. For the current project, the industrial classes of interest, candidate facilities, image features and their definitions were specified prior to obtaining significant numbers of images.

SELECTION OF INDUSTRIAL CLASSES

We defined the industrial classes of interest by following a seminal paper on photointerpretation from Chisnell and Cole [7], who divided all industries into six categories and provided a vocabulary of names for industrial objects. Table 1 lists their major industry groups.

Industrial Class	Description	Example
<i>Extraction</i>	Exploits the natural resources of the earth and its waters with the minimum handling required to accumulate raw materials in a form suitable for transportation or processing.	Quarry
<i>Mechanical Processing</i>	Engages in the sizing, sorting, separating, or otherwise changing the physical form or appearance of raw materials.	Sawmill
<i>Chemical Processing</i>	Utilizes chemicals or chemical processes to separate or rearrange the chemical constituents of raw materials.	Refinery
<i>Heat Processing</i>	Utilizes heat as the primary process to refine, separate, or re-form raw materials, or derive energy from them.	Smelter
<i>Heavy Fabrication</i>	Forms and assembles the parts of finished products which individually are large and heavy.	Shipyard
<i>Light Fabrication</i>	Forms and assembles the parts of finished products which individually are not extremely heavy and are rarely very bulky.	Boat repair yard

Table 1. *Chisnell and Cole's Industrial Classes*

Since their paper first appeared (1958), new technologies and, arguably, new industry types have emerged. To the list of Chisnell and Cole's industry types, we have added "Semiconductor Industry," which did not exist at the time that Chisnell and Cole wrote their work. This class is intended to cover not only the processing of semiconductor materials but the fabrication of integrated circuits, printed circuits and related items. We then prioritized the order in which we would collect examples from each industry type. The current version of the Benchmark Imagery Suite includes the following classes:

- Chemical Processing (27 facilities) — *e.g.*, oil refineries.
- Heavy Manufacturing (31 facilities) — *e.g.*, steel foundries and metal-casting plants.
- Heat Processing (37 facilities) — *e.g.*, nuclear and coal-fired electrical power plants.
- Mechanical Processing (83 facilities) — *e.g.*, aluminum processing plants and water treatment plants.
- Semiconductor Industry (12 facilities) — *e.g.*, computer chip companies.

In consultation with the NA-22 program office, Chisnell and Cole's Extraction and Light Fabrication classes are not included in the current Benchmark Imagery Suite.

CHOICE OF FACILITIES

Next, a list of candidate industrial facilities and their locations (i.e., a gazetteer) was prepared, compiled from open-source information, including two EPA databases, an OSHA database, the Energy Justice database and the North American Industry Classification System (NAICS) by the US Census Bureau [8]. A separate gazetteer was formed for each industrial type used in the Benchmark Imagery Suite, with a unique alphanumeric code assigned to each site. Each industrial facility was given a unique benchmark ID indicating whether it was chemical processing (CP), heavy manufacturing (HM), heat processing (HP), mechanical processing (MP) or semiconductor industry (SI). This set of gazetteers, which contains the name of each facility, its location and other site-specific data, are working documents that are not included with the Benchmark Imagery Suite. Minimum and maximum latitude and longitude values were calculated for each site, based on the nominal geographic coordinates and size of each facility. Extremely large facilities (many square miles) and small facilities (one or two buildings) were excluded.

CONSTRAINED VOCABULARY

The Benchmark Imagery Suite uses a simplified vocabulary derived from Chisnell and Cole. The list of labels in this work is limited to objects that a human could identify without significant specialized knowledge. Modifiers that imply hidden knowledge are dropped from labels. For example, with regard to standing water, we include the simple term “pond,” but Chisnell and Cole list a variety of types of pond that imply specialized knowledge of functionality such as “flocculation clarification basins” in a water treatment plant. Table 2 below lists the version of the constrained vocabulary used in the Benchmark Imagery Suite. During the course of the work the vocabulary evolved slightly, with some of the original definitions being slightly revised and a few additional terms put into the list. Appendix B gives the photointerpretation key created for the Benchmark Imagery Project with photographic examples of each of the terms in Table 2.

<i>Image Feature</i>	<i>Definition</i>
Building	A structure consisting at a minimum of a foundation, a roof, and supports and which is not a <i>house</i> or an <i>industrial building</i> .
Conveyor	A structure (<i>e.g.</i> , belt, duct, pipe, etc.) used to move material (<i>e.g.</i> , a conveyor belt, pipeline, or steam pipe), electricity (<i>e.g.</i> , conduit), or people (<i>e.g.</i> , sky bridge).
Cooling Tower	A type of cooling unit which is typically a very tall, hyperboloid-shaped structure.
Cooling Unit	A piece of equipment used to dissipate or remove heat from other equipment or a building by dumping it into the surrounding environment through the use of an intermediate fluid (<i>e.g.</i> , water, refrigerant).
Crane	Permanently mounted equipment affixed to a facility which is used to lift, and possibly move, heavy objects — specifically, an “Overhead Crane.”
Electrical Substation	A collection of structures used for the transmission, transformation, distribution or switching of electric power.

<i>Image Feature</i>	<i>Definition</i>
House	A structure consisting at a minimum of a foundation, a roof, and supports, which is (usually) small in areal extent and is typically located within a residential area.
Industrial Building	A structure consisting at a minimum of a foundation, a roof, and supports, which is (usually) large in areal extent and located within the bounds (<i>e.g.</i> , fence-line) of an industrial facility.
Kiln	Equipment (<i>e.g.</i> , furnace or oven) for drying, baking, or burning material.
Pile	A purposefully-placed heap of solids with some angle of repose.
Pond	A container of any shape (<i>e.g.</i> , natural, round, rectangular, square, etc.), installed either above or below ground, which uses either soil or concrete or metal to effect the containment, and which has an open view to the sky, such that the liquid within often appears dark in tone (since the liquid may contain significant solids).
Rail Line	A linear structure consisting of a foundation (<i>e.g.</i> , compacted material) upon which rests horizontally oriented bars of metal, to facilitate the movement of locomotives, railcars, etc.
Stack	A vertical structure used to convey exhaust; usually relatively tall in height compared to surrounding structures.
Tank	A non-mobile (<i>i.e.</i> , fixed), enclosed structure (<i>i.e.</i> , with a lid and walls), of wide variety in shape (<i>e.g.</i> , cylindrical, spheroidal, etc.), and oriented either horizontally or vertically, for holding solid, liquid, or gas.
Tower	A vertical piece of equipment (<i>e.g.</i> , watch tower, fractionating tower, cracking unit, etc.), usually relatively tall in height compared to surrounding structures.
Water Tower	A tank, elevated via placement on top of a tower.

Table 2. *Constrained vocabulary of labels in the Benchmark Imagery Suite.*

III. Real Images

In this report, “real” images refer to images captured with an imaging sensor, as opposed to computer-generated “synthetic” images. This section of the report focuses on the real images in the Benchmark Imagery Suite.

REQUIREMENTS AND SCOPE OF IMAGERY

The images in the Benchmark Imagery Suite must meet several conditions. They must be overhead images of industrial facilities. They must be geospatial in nature. They must have adequate lateral spatial resolution to enable detection of the key features listed in Table 2 — nominally with ground sample distances (GSD) of one meter or less. They must cover a large enough area to show the environment around each facility — on the order of 1 to 2 km². They must be accessible and affordable to users. There must be adequate numbers of images in the set. Previous work in computer vision has suggested that many tens or even hundreds of images are needed for each modality represented in a benchmark suite (visible-band, infrared, broad-band,

multispectral, hyperspectral, radar, and lidar, for instance), in order to provide the richness of features that algorithm developers might need [9].

To collect such a wide assortment of images of a set of facilities captured with multiple overhead imaging modalities would have required a large effort. The Benchmark Imagery Project, however, was a modestly sized project whose intent was to make an initial collection of images, not a complete set representing all imaging modalities with all possible variations in the scene. Therefore, with few exceptions (see Section IV Lidar), we chose to use only visible-band conventional digital images. Such overhead imagery is the most readily accessible, affordable and distributable of all of the imaging modalities.

ACQUIRING REAL IMAGES

A ready source of visible-band overhead imagery is the USGS National Map Viewer, <http://nationalmap.gov/viewer.html> (formerly called the Seamless Data Warehouse <http://seamless.usgs.gov/>). These large-area aerial images, which are GeoTIFF [1], carry terrestrial coordinates of the scene embedded in their headers and often have a GSD as small as tens of cm. Although restricted to images over the United States, the images are in the public domain and can be obtained free of both cost and licensing restrictions. This potentially permits wide-spread sharing of the benchmark imagery. In this regard, we followed the precedent set by the Overhead Imagery Research Dataset (OIRDS) project [10].

To obtain a desired image from the USGS data bank, one must specify the geographic region of interest by providing the maximum and minimum values of latitude and longitude. This can be done either manually through a graphical user interface or automatically in batch mode. The latter approach allows one to download multiple images with a single command file. We wrote a software tool to read the geographic information of each site from the gazetteers mentioned previously and submit a batch download request to the USGS server. We downloaded well over three-hundred images this way and organized them by industry type in separate folders. We manually inspected each downloaded image and rejected obviously incorrect pictures. The most common errors were blank or grossly misshapen images or images that did not contain the intended industrial plant. The default size of each real image was 1km × 1km. A few were slightly larger in order to capture all of a facility plus its surroundings.

EXAMPLE IMAGES

The pictures on the following pages illustrate the types of images downloaded for the Benchmark Imagery Suite. Table 3 at the end of this section gives the list of all facilities by index number and type.



Figure 3. *Illustration of a heat processing facility, HP0058_C8, GSD: 0.3m.*



Figure 4. Illustration of a chemical processing facility, CP28_VDE, GSD 0.5m.



Figure 5. Illustration of a heavy manufacturing facility, HM45_V3F, GSD: 0.3m.



Figure 6. Example of a mechanical processing facility, MP032_VBQ; GSD: 0.3m.

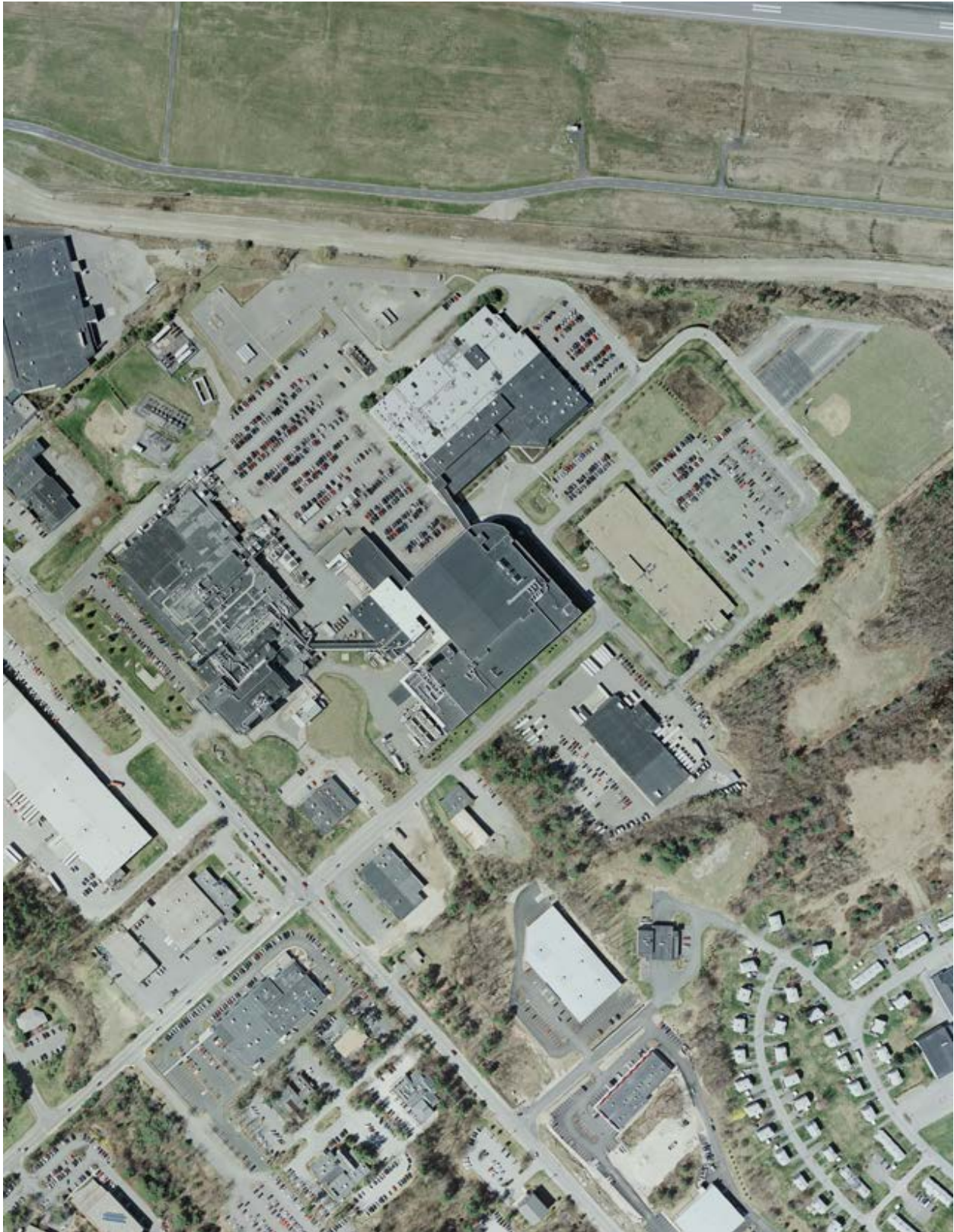


Figure 7. Example of a semiconductor industry facility, SI0018_U4F; FSD = 0.15m.

Chemical Processing 27 Facilities	Heavy Manufacturing 31 Facilities	Heat Processing 37 Facilities	Mechanical Processing 83 Facilities		Semiconductor Industry 12 Facilities
CP2_U3Z	HM2_VA6	HP0003_V80	MP001_P16	MP058_P12	SI0002_V58
CP3_U5W	HM5_VGO	HP0031_U44	MP002_V6I	MP059_P14	SI0005_U62
CP5_VGA	HM6_V6L	HP0034_U4X	MP004_P11	MP060_VC4	SI0006_U62
CP7_VD6	HM7_U52	HP0039_VJ4	MP006_P18	MP062_U30	SI0013_U62
CP10_U60	HM9_V77	HP0040_VCY	MP007_P17	MP063_P14	SI0015_U5H
CP14_P15	HM10_V77	HP0044_VDN	MP008_P11	MP064_P15	SI0018_U4F
CP15_VDA	HM17_V5Z	HP0047_C87	MP009_P18	MP066_P12	SI0019_V7Q
CP23_V24	HM25_VGA	HP0049_C87	MP010_P15	MP067_P14	SI0020_U4H
CP28_VDE	HM28_VGA	HP0050_VGA	MP011_VGA	MP068_P16	SI0021_U3B
CP32_VIO	HM31_C87	HP0052_V2V	MP012_P16	MP069_VD2	SI0024_U6D
CP35_P17	HM36_VGA	HP0056_C87	MP013_P15	MP073_P15	SI0027_U62
CP38_P16	HM41_V5Z	HP0058_C87	MP014_P16	MP074_P14	SI0030_U62
CP39_VIY	HM45_V3F	HP0060_VIT	MP015_U3J	MP077_C87	
CP42_V24	HM52_VBV	HP0062_V5D	MP016_P16	MP078_VHD	
CP43_P16	HM53_VBJ	HP0068_C88	MP017_P16	MP081_P13	
CP44_P17	HM56_VBV	HP0069_U3J	MP018_C87	MP082_P18	
CP45_VD0	HM59_VBU	HP0078_V80	MP019_P18	MP083_U5R	
CP51_U62	HM63_C87	HP0082_V6X	MP020_P11	MP084_P18	
CP58_VD6	HM64_VAY	HP0086_C87	MP022_P16	MP087_P14	
CP6_VDA	HM74_VAT	HP0087_VCA	MP024_VC4	MP089_P15	
CP61_VBI	HM75_VAT	HP0104_P16	MP025_U4K	MP090_P15	
CP78_U42	HM77_VAW	HP0107_U30	MP027_P18	MP092_P16	
CP81_VD6	HM78_VC3	HP0108_VDA	MP029_V78	MP093_P15	
CP85_U3L	HM79_VBY	HP0109_VEN	MP030_VC4	MP094_P17	
CP86_VCP	HM80_VBG	HP0110_V5Q	MP031_P15	MP095_P17	
CP93_V7Z	HM81_VAW	HP0112_VF7	MP032_VBQ	MP096_V77	
CP98_U43	HM83_C87	HP0115_VCS	MP035_P16	MP097_P18	
	HM86_VC5	HP0117_VBO	MP038_P18	MP098_P17	
	HM92_VCK	HP0119_VAS	MP039_P18	MP099_P13	
	HM103_U4X	HP0121_P18	MP040_P17	MP100_P17	
	HM104_U4X	HP0122_P18	MP041_P18	MP101_VE6	
		HP0130_U30	MP043_P16	MP102_V6K	
		HP0142_C88	MP044_P10	MP104_VFI	
		HP0148_V5D	MP045_P10	MP105_P18	
		HP0150_VG9	MP046_P10	MP106_P18	
		HP0155_V7Q	MP047_P10	MP107_V5V	
		HP0156_VH7	MP048_P10	MP108_CM2	
			MP049_P10	MP109_P13	
			MP053_P17	MP111_P18	
			MP054_P16	MP113_CM0	
			MP055_P17	MP114_P12	
			MP057_P16		

Table 3. Aerial Images Included in the Benchmark Imagery Suite v1.1a.

IV. Lidar Images

The Benchmark Imagery Suite includes a small amount of lidar elevation data. Lidar, which stands for “Light Detection And Ranging” or “Laser Imaging, Detection And Ranging” depending upon whom one asks [11], gives 3-dimensional geospatial data. Unlike conventional 2-dimensional imagery, a set of lidar data enables one to determine not only the ground coordinates of objects in a scene but also their elevations (height above the reference geoid). In particular, for scenes with known or slowly varying ground elevation, a lidar data set enables one to determine height of objects above or below the surrounding terrain, which is useful for testing algorithms that detect structures like buildings, towers, fences and power lines.

<i>Chemical Processing</i>	<i>Heat Processing</i>	<i>Semiconductor Industry</i>
CP2_U3Z	HP0003_V80	SI0019_V7Q
CP51_U62	HP0034_U4X	
CP61_VBI	HP0044_VDN	
CP81_VD6	HP0078_V80	
CP93_V7Z	HP0087_VCA	
CP98_U43	HP0115_VCS	
	HP0117_VBO	
	HP0119_VAS	
	HP0155_V7Q	

Table 4. List of facilities in Benchmark Imagery Suite v1.1 for which lidar data are included.

The lidar datasets cover the same ground as sixteen of the real images (see Table 4). The data have been converted from LAS point-cloud format [12] to RAS raster-image format. The grid of pixels in each rasterized lidar image exactly matches that of the associated RGB image, making the lidar image like an additional layer of the visible-band image. Instead of giving spectral information, each pixel of the rasterized lidar image contains a single scalar value denoting the elevation of whatever object occupies the ground coordinates of that pixel. Figure 8 below shows an example of a rasterized lidar image, first by itself and then overlaid on the corresponding GeoTIFF aerial image.

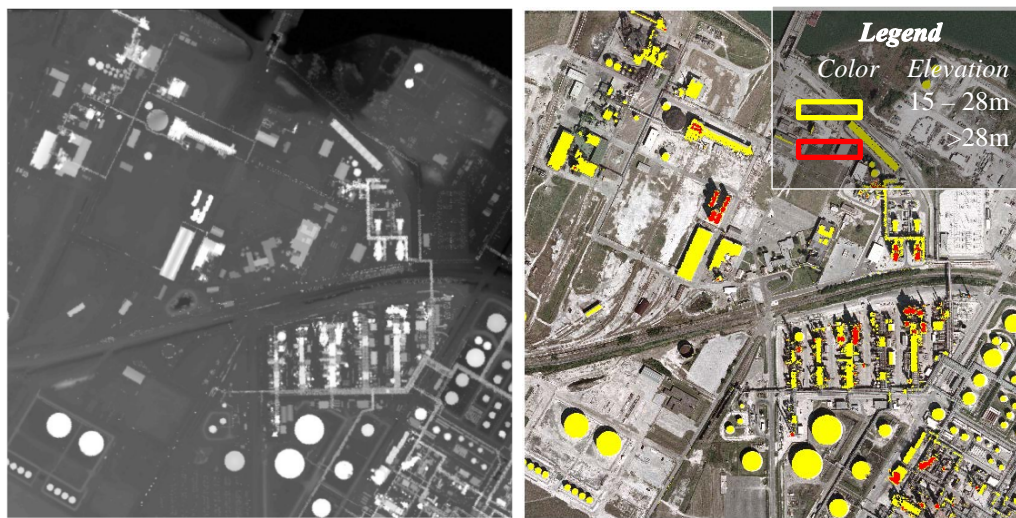


Figure 8. Example of a rasterized lidar image. Site CP2_U3Z. Left: Elevations are displayed as gray scale values, with black being lowest and white highest. Right: Rasterized lidar image overlaid on the corresponding GeoTIFF Image.

OBTAINING LIDAR DATA

The USGS web server provides a national lidar data set for the United States. Coverage, though not complete, is extensive. Almost 40% of the sites with visible-band imagery in the Benchmark Imagery Suite v.1.1 have lidar data sets available. All of the lidar data in the current suite come from the USGS server via the Earth Explorer tool (<http://earthexplorer.usgs.gov>), a client/server interface that provides access to USGS’s Earth Resources Observation and Science (EROS) Center archive.

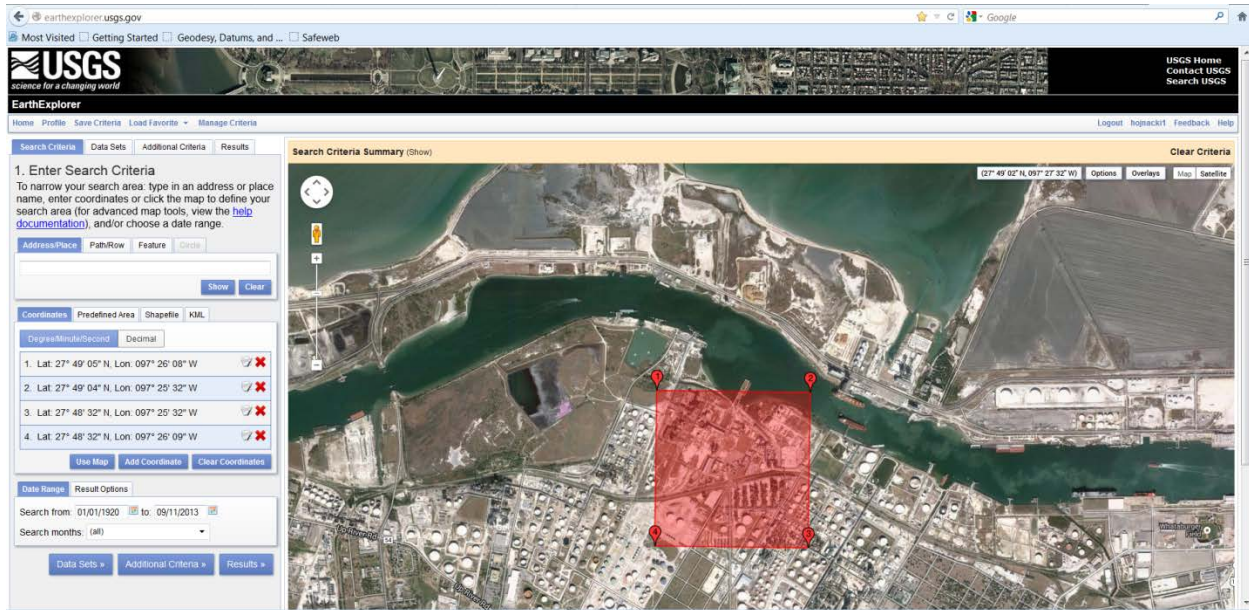


Figure 9. Graphical User Interface for the USGS’s Earth Explorer showing an area of interest for site CP2_U3Z (red square) superimposed upon an image of the surrounding region.

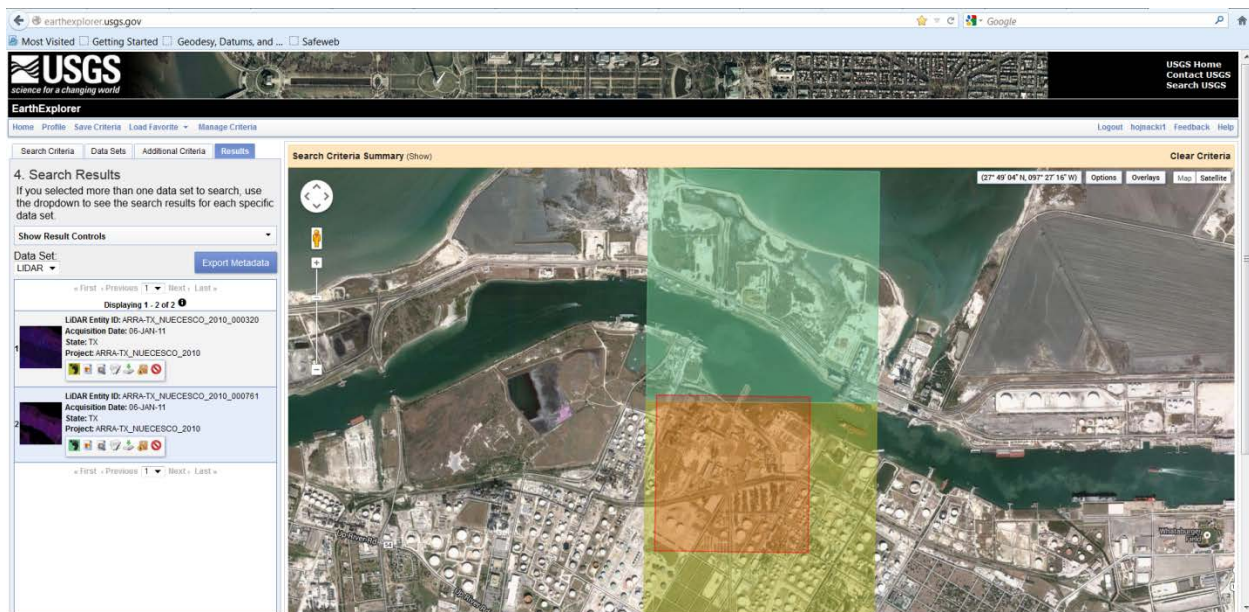


Figure 10. Graphical User Interface for the USGS’s Earth Explorer showing that an area of interest (orange square) straddles two separate tiles of USGS data (yellow and green boxes).

To find lidar datasets that covered the same ground area as the visible imagery in the Benchmark Imagery Suite, we first obtained the geographic bounding box coordinates of each GeoTIFF image in the suite from its header. We then used the bounding box coordinates to create a Keyhole Markup Language (KML) polygon file. (KML is a tag-based language for displaying geographic data on computer-based two-dimensional maps and three-dimensional Earth browsers, such as the Google Map component used in Earth Explorer.) The KML polygon file was uploaded to Earth Explorer to define an area of interest to search for lidar data. Figure 9 on the previous page shows the USGS Earth Explorer user interface during the process of entering a search request for a given site. In this example the site is CP2_U3Z, the same site shown in Figure 8. The Data Search component is located on the left side of the screen, and the Google Map component is on the right side. In the figure, the bounding box for CP2_U3Z has just been uploaded and is shown in red.

The geographic data on the USGS server are organized in tiles that are approximately contiguous, square zones. Each tile of lidar data consists of sets of (x, y, z) points representing easting, northing, and elevation measured from a reference geoid. In addition, projection coordinate system information is stored with each tile.

When responding to a query for lidar data, Earth Explorer searches its database for all tiles that overlap the specified bounding box of interest and that contain lidar pulse returns falling within the bounding box. It then returns a list of such tiles and graphically displays them. Figure 10 on the previous page shows the search results for the particular bounding box for CP2_U3Z. As shown in the figure, there are two tiles of lidar data (highlighted in yellow and green) that overlap the region of interest in this example (highlighted in orange).

Ideally, one would like to have lidar data that have been collected at the same time and cover the same ground area as the visible imagery in the Benchmark Imagery Suite. However, the lidar data available in Earth Explorer were not collected at the same time as the visible imagery. For many of the scenes, multiple lidar tiles were needed to completely cover the corresponding ground area. For some scenes, the tiles overlapped and/or were from lidar campaigns that occurred during different years. Such spatial and temporal differences in tiles can lead to differences between the lidar imagery and the visible imagery and even to differences between lidar data sets. In the Benchmark Imagery Suite, no attempt was made to reconcile differences that might have occurred as a result of time lapses between tiles of data.

When choosing which data to download, tiles from the same lidar campaign were selected where possible. In addition, data from recent collects were preferred over data from older collects to better match the man-made objects in the scenes and to take advantage of advances in lidar precision and accuracy in the more recent collects. In all cases, a set of tiles was selected that, taken together, completely covered the bounding box area.

PROCESSING THE LIDAR DATA

After collecting the lidar tiles, software was written to automate the following steps for each visible image scene:

First the point clouds from multiple tiles were combined. Next, each point in the point cloud was compared to the coordinates of the bounding box in the GeoTIFF header of the corresponding visible image. Points falling outside the bounding box were removed.

Likewise, anomalous points were identified and eliminated. Anomalous points included noisy points with values outside the range of realistic elevation values for objects in the scene. Anomalously high points can be caused by birds, low-flying aircraft or atmospheric aerosols; anomalously low points (i.e., negative elevation values) may be caused by excessively long return times resulting from multiple reflections.

Next, the raster size and pixel scale of the visible image were read from the GeoTIFF header. This information was used to create a regular grid and the point cloud was interpolated to fit the regular grid. Three interpolation algorithms were tested: natural, cubic, and v4. The “natural” interpolation algorithm subjectively produced the best results. Of the three algorithms tested, it was also the most used in the literature. The resulting rasterized data (interpolated and gridded elevation data) was flipped vertically to match the orientation of the GeoTIFF image (see Figure 8). Finally, the elevation data with corresponding easting and northing grid was saved as a matrix file. The data were also saved as indexed tiff images with corresponding color maps.

To verify that the rasterized lidar data matched the resolution of the visible imagery, the rasterized lidar elevation image was overlaid on the corresponding visible image. Since the site CP2_U3Z is at sea level and the ground area is fairly flat, resulting lidar elevation values are close to height above the ground. To show the overlap, lidar points with elevation values between 15m and 28m were mapped to yellow, and points with elevation above 28m were mapped to red and then overlaid on the corresponding visible image. The result is shown on the right side of Figure 8. The rasterized lidar elevation data lines up well with the visible image data.

In creating the rasterized lidar elevation images, we encountered many situations where the coordinate system used in the lidar data differed from that used in the GeoTIFF visible-band imagery. Such occurrences were due to the many different mapping conventions commonly available and frequently used in remote sensing. When the coordinate systems differ, the overlap between the lidar image and its corresponding GeoTIFF visible-band image can be artificially degraded by many meters. While converting from one coordinate system to another is generally straight-forward and would virtually eliminate the inter-image displacement artifacts, we chose not to do that task in this first version of the Benchmark Imagery Suite. Instead, we required that the lidar and the GeoTIFF coordinate systems had to be the same. Based on that requirement, only 16 of the visible band images had lidar data that merited inclusion in the Benchmark Imagery Suite v1.1. The task of reconciling differences in coordinate systems in order to increase the amount of lidar data in the suite has been left for future work.

V. Synthetic Images

The synthetic images created for the Benchmark Imagery Suite are simulated digital photographs of 3-dimensional computer models of industrial facilities. The models are placed on geospatially accurate 3-dimensional backgrounds built by draping a 2-dimensional orthorectified image of a wide scene over a digital elevation model (DEM) of the underlying terrain. The computer program SceneWorks creates the images. It merges the 3-D models with the 2-D background image and DEM, sets the synthetic camera parameters and lighting, adds selected features if desired (*e.g.*, clouds, motion smear) and takes a simulated photograph. Like the real images, the synthetic images in the Benchmark Imagery Suite are RGB visible-band, and they use values of

areal extent ($1 - 2 \text{ km}^2$) and GSD ($0.3 - 0.4\text{m}$) that are similar to those of the real images. There are 38 synthetic images in the Benchmark Imagery Suite v1.1a, all of which are stored in GeoTIFF format.

SceneWorks, the view rendering engine used in this project, is a real-time 3-D visualization framework developed by LLNL that is used to support the Army, Navy, Air Force, NGA, and DHS [13], [14]. It leverages the video-game industry and related advances in off-the-shelf video cards. Images take approximately $1/30^{\text{th}}$ of a second to generate, permitting real-time 3-D interaction between the user and the scene. Real-time interaction is a tremendous benefit when setting up the scene, placing components, and tweaking properties such as lighting and materials.

Because the synthetic images in the Benchmark Imagery Suite are a mixture of computer-generated features and real photographs, they are technically composite images, not purely synthetic. However, the key industrial components in each scene, their surface textures, the lighting and the shadows from those components are generated numerically. In this document, we use the terms *synthetic image* and *composite image* interchangeably when referring to the set of computer-generated images in the Benchmark Imagery Suite.

A rich assortment of synthetic images is desired in a benchmark suite in order to test algorithms over a wide range of inputs and to help prevent algorithms from being mistakenly “tuned” to specific conditions. Parameters that can be varied synthetically include the type of facility, the detailed components of the facility, the layout of the facility, the background scene, the base terrain, the environment, the lighting, the material properties of each object in the scene, the weather, and all of the camera viewing properties (position, orientation, imaging mode, spectral sampling, spatial sampling, image quality, ...). However, there is a trade-off between having a large variety of images and the effort required to create them.

The types of industrial facility included in the suite are a coal-fired power plant and a nuclear power plant — heat processing facilities. The components and their arrangement within each facility are patterned after overhead photographs of real-world power plants and are reviewed with subject matter experts. Each power plant is placed in one of four backgrounds: arid, rural/forest, coastal and urban/suburban, giving a total of eight different combinations of facility and background. Each of the two types of power plant has a specific set of synthetically generated industrial features that are retained from one background to another, but the arrangement of those features changes from background to background.

In addition to the eight synthetic images described above, we created 10 images that were variations of the nuclear power plant on the arid background. The objective was to provide a set of images for systematically investigating the impact of facility rearrangement on a segmentation/annotation algorithm, keeping all other parameters (*e.g.*, background, lighting, time of day) constant. To help do this, we wrote a computer tool to assist with creating facility layouts. Our approach was to adapt an existing computer code that automatically arranged electronic components on a circuit board, subject to a set of placement rules, B*-Tree Floorplanner [15]. The computer-generated facility layout specifications were passed to the synthetic scene generation software which, in turn, placed the components of the facility on the background and created a synthetic image. The image was then checked for errors and the layout was corrected and the image redrawn as needed. Once we had the 10 variations of the original scene, we created 2 more variants of each of the 10 new images by exploiting reflection

operations. In one variant, we “flipped” the placement of the facility components about a horizontal axis. In the other variant we “flopped” the placement of the facility components about a vertical axis.

Figures 11 and 12 below show examples of two synthetic images, a coal-fired power plant in a coastal environment and a nuclear power plant in an arid environment.



Figure 11. Synthetic image of a coal-fired power plant in a coastal environment.



Figure 12. Synthetic image of a nuclear power plant in an arid environment.

In the remainder of this section we provide the details of generating synthetic images used in the Benchmark Imagery Suite.

- Model Building
- Base Terrain Preparation
- Initial Facility Layout
- Image Compositing
- Automating the layout process

Using the SceneWorks software we imported the terrain based data (overhead imagery and elevation data). Then we placed the 3-D components into the scene and configured the lighting, shadows, and other environmental parameters. Next we positioned the virtual camera (altitude and view angle) and selected the image size (in pixels). Then the pre-processed, synthetic image was captured in SceneWorks.

MODEL-BUILDING

By a *model* we mean a computer object that represents a physical structure such as a building. The models are 3D spatially accurate geometric meshes that have been textured with RGB images. When a desired model is not already in the SceneWorks library, we obtain it in one of two ways: we buy it, or we fabricate it by starting with two-dimensional overhead images.

SceneWorks, requires three-dimensional models. Creation of highly detailed facility components should be avoided whenever possible. Making such components is very time consuming, and there are tremendous existing resources on the internet where 3D models of entire facilities can be purchased for only a few hundred dollars (see Turbosquid.com [16]). There are also free resources (see Trimble 3D Warehouse [17]) where 3D models can be downloaded, but in our experience the quality of these models is insufficient.

Using Commercial 3-D Components

SceneWorks reads models in text based, OSG format. When an existing model is not available directly in that format, it can typically be downloaded and converted to OSG by the AutoDesk software application 3DSMax [18] with the OSG exporter plug-in [19] installed.

Complete facilities should be broken into components, which can be also done with 3DSMax. Each component should be moved to the origin with the ground at its base (i.e. ‘ground-clamped’) and the origin at its center. There is no need to create multiple copies of identical components. For example, if the complete facility has two cooling towers, only one cooling tower needs to be exported as an individual component. It can be replicated in SceneWorks.

3D models have textures which have three different material lighting properties: ambient, diffuse and specular. Sometimes the material lighting properties of commercially acquired 3D components make the lighting to look unrealistic compared to real overhead images. These components can be changed in either 3DSMax before export, or by modifying the exported OSG file directly. We prefer modifying the OSG file because this can be done in bulk using python scripts to change all textures in all models in a given directory at once. Results can be viewed in the context of the scene in SceneWorks almost instantly, with no modification of any other files. Visible overhead images typically have very harsh lighting compared to the lighting typically found on 3D models. Model lighting can be corrected to be more realistic simply by increasing the diffuse lighting, and reducing the ambient lighting.

Creating 3-D Components from 2-D Images

Some hard-to-find 3D components can be created very quickly from 2D objects. Components which have no height such as railroad tracks, basins and parking lots are easily created by texturing a basic 2D shape with imagery. These objects do not need 3D geometry because they are flat; therefore they do not cast significant shadows and do not have significant perspective.

The procedure starts with finding an example of the object in high-resolution orthorectified imagery. This imagery is acquired from the USGS, as our other imagery used in creating base terrains. The object’s dimensions are measured in an online program such as Google Earth [20], or from the image geospatial data directly using a GIS program like Global Mapper [21]. Once the dimensions are known, then the object is created in using a basic 3D modeling program, like Trimble Sketchup [22]. Even though the object is 2D, it must exist in a 3D world, so it must have some nominal height. Using Sketchup we extrude the object vertically (along the Z axis) so that the height is approximately 1/2 meter. SceneWorks will automatically disable shadow casting on objects that are less than 1 meter in height to prevent any visual artifacts.

The top of the 3D object should be textured with the image of the object in Sketchup. Before that, the texture image must be cropped out of the large orthorectified satellite image. We are

using a free Photoshop-like tool called “GIMP” [23]. Once the texture image of the object is cut out and saved in PNG format [24] using GIMP, the object can then be textured on the top face of the 2D object with the texture image. The 3D object is then exported from Sketchup as a DAE (Collada) [25] file. This file can then be converted to OSG using the OSGConv tool [26], which is included with SceneWorks.

BASE TERRAIN PREPARATION

The four different environments we selected to use as background images include sites in arid, coastal, rural/forest, and urban/suburban areas. The arid environment was previously selected from a region near Kettleman City, CA. We found our coastal background images by performing a broad area search of the United States coast line. We selected a region near Galveston, TX. For the forest background we searched the main forested areas of the continental U.S. and chose a region near Ellington, MO. For the urban background, we searched the Environmental Protection Agency website for areas that would provide an adequate clearing for a facility in an urban setting, and we selected an area near Denver, CO.

When selecting a location for building a base terrain there are several desirable and undesirable characteristics that must be considered. The location must contain a flat clearing large enough for multiple facility types (at least 100m x 100m) and free of any discernible objects that would prevent a facility from being placed (built) there. The clearing should have no trees, buildings, or vehicles. Base imagery may contain shadows being cast from nearby trees and small buildings, but the shadows simulated in SceneWorks should match the underlying imagery. Finding the date that the imagery was taken can help in generating shadows that match, as SceneWorks generates shadows from facility 3D models using accurate solar ephemeris modeling.

Google Earth is very useful tool for seeking out and evaluating different locations. It displays high resolution imagery and DEM, and it is quickly navigated. Once a potential location is found in Google Earth and appears to be viable, the source data (DEM and imagery) are acquired and evaluated. Most imagery in Google Earth is commercial and restricted in distribution. We acquire our source data using the USGS National Map Viewer and Download Platform [27]. Using the USGS website we acquire a 5km x 5km region of imagery and DEM centered on the desired location of the facility. For examples of images reviewed and more information on selecting base imagery, please see Goforth, *et al.* [28].

The source data (imagery and DEM) must be converted into a format that optimized for 3D real-time rendering. Using SceneWorks’ terrain generator, the input files are selected and the 3D terrain is generated in only several minutes. This 3D real-time format is an Open Scene Graph (OSG) terrain format, IVE. SceneWorks also uses the geospatial metadata from the input files to configure its internal GIS engine and coordinate system. The SceneWorks scene map is now saved and ready for facility components to be added.

INITIAL FACILITY LAYOUT

Our approach for facility layout is to select one or more real facilities from our benchmark imagery set as a template for our layout and then to review that layout with one or more subject matter experts. There is a moderate amount of effort in creating a realistic layout of a facility.



Figure 13. Coal-fired power plant, HP52.



Figure 14. Image of an actual nuclear plant, HP104.



Figure 15. Image of an actual nuclear plant, HP119.

The synthetic images in the Benchmark Imagery Suite include two types of facilities — a nuclear power plant and a coal-fired power plant. These two facilities are each placed in 4 different environments: arid, coastal, urban/suburban, and rural/forest. The placement of 2 types of facilities in 4 different environments yielded a total of 8 synthetic images. The facilities must be modified slightly for each environment. For example, cooling towers are not needed when a power plant is positioned on the ocean in a coastal environment.

The coal-fired power plant layout is based on Image HP52. Comparison of synthetic and real images can be seen in Figures 11 and 13. The nuclear power plant layout is based on Benchmark Images HP104 and HP119. Comparison of synthetic and real images can be seen in Figures 12, 14 and 15.

IMAGE GENERATION

Once the facility layout is established, the 3D models are placed in the scene using the SceneWorks STAGE tool. The models must be ground clamped to ensure there are no gaps between the building and ground. If model is not flush with the ground, the object will appear to be floating above the ground, and the shadows will not cast directly from the base of the object. Then simulated time of day and date are set manually so that shadows match the underlying base imagery. This is done with the real-time shadow rendering capability of SceneWorks.

In SceneWorks, the simulated camera is positioned with nadir view at an altitude high enough to allow for a large amount of lens magnification using the zoom feature. This removes a large amount of perspective, simulating an image taken from a high-altitude platform. At this point the scene has been constructed and the settings configured, so generating the composite image is trivial. The image-capture button is pressed, and the composite image is automatically generated and saved to disk as a TIFF image.

Using GIMP, we manually post-processed each image to enhance the brightness, contrast, saturation and lightness. Post-processing improved the realism of the image by visually matching the colors and exposure of the composite image to other similar images in the benchmark imagery set. After post-processing, we converted the output image from TIFF to GeoTIFF. To do so, we used SceneWorks to return the UTM coordinates of the four corners of the original overhead background image. Then we computed the GSD of each pixel and, using open-source “listgeo” and “geotifcp” tools [1], added the geospatial data (GSD, UTM extents) to the TIFF file as GeoTIFF metadata.

AUTOMATING THE LAYOUT PROCESS

The layout of an industrial plant is the conceptual skeleton around which synthetic scenes of the plant are built. Experience gained creating the initial facility layouts for the Benchmark Imagery Suite showed that having a faster way to identify and arrange the components of a given plant could potentially increase the rate at which synthetic images were generated. We therefore experimented with a way to automate the layout process [29].

In seeking a workable automatic layout scheme, we attempted to limit the effort and yet make it useful for the current generation of the Benchmark Imagery Suite. A key self-imposed requirement was that the process — whatever it turned out to be — must add more usable images to the set of synthetic scenes. When we began the effort, we did not know the details of the obstacles we might face or how quickly (or if) we could pull together working software. One of our goals in undertaking this experiment in automation was simply to identify key issues from which we might benefit in future synthetic scene simulations.

Facility layout is a complex problem with many objectives and constraints. Designers have been using computer algorithms to assist them in optimizing facility layouts since the early 1960s [30]. Although a large body of computerized tools and knowledge now exists, layout optimization remains an active area of operations research. (All the more reason for us to avoid writing our own layout software!)

To limit the scope of this effort, we chose simply to rearrange the components of our existing synthetic images on their backgrounds. This enabled us to focus on automating only one class of

facility layout issues — namely, topological/geometrical constraints. Examples of such constraints included things like the following:

- Proximity — *e.g.*, minimum and maximum separation between a nuclear containment vessel and a steam turbine.
- Connectivity — *e.g.*, a conveyor belt transporting waste ash from a coal combustion chamber to an ash pile.
- Size — *e.g.*, minimum and maximum linear extent of the footprint of a single component, group of components or an entire facility.

B*-Tree Floorplanner

One algorithm for performing automated layout is B*-Tree Floorplanner [15], which is readily available through the University of Michigan freeware package Parquet. [31] The algorithm is used for laying out electronic components and common lines (buses) in very large scale integrated (VLSI) circuits. It arranges a list of user-specified rectangular boxes and their dimensions into a given area, minimizing both dead space and bus overlap. The inputs are a set of macros and (optionally) a fixed outline. The macros define the boxes and their connectivity, and the fixed outline defines the boundaries of the area to be filled. The output is a floor plan — a layered arrangement of the boxes and their interconnections (buses) within the required outline. If the fixed outline is omitted from the input (“bus-driven floor planning” or BDF mode), the software solves the classic packing problem of minimizing the area of a given set of non-overlapping rectangles in a plane [32]. Results are local solutions to minimizing a cost function, and they vary from run to run, even with nominally identical input.

Instead of using the boxes to represent VLSI electronic components, we used the boxes to represent industrial components in a large facility. Each box contained just one feature (*e.g.*, building, pond, or cooling tower) plus a buffer zone. We operated the software in the BDF mode and produced ten different layouts of the nuclear power plant in an arid environment. Each run of the algorithm maintained the same salient components and constraints (adjacency and connectivity).

KML Generator

To interface the resulting layouts to SceneWorks, we developed a computer code, KML Generator, that converted the output of B*-Tree Floorplanner into KML model layouts. KML Generator took the footprint size and important information about the models of the individual facility components (component label, centroid offset, shape and base rotation), and generated a KML shape file that could be displayed in Google Earth, thereby enabling us to visualize the layouts easily.

After checking each layout, we read the KML files into SceneWorks. SceneWorks, in turn, placed the models of the various industrial components onto the background using the coordinates and orientations specified in the KML file.

Results of Using an Automated Layout Tool

We did achieve our goal of increasing the number of synthetic images with the automated layout software. The process of learning to use that software, not surprisingly, was more time consuming than we could have estimated. This effectively reduced the time available for

generating more images. Issues like correctly specifying sizes of the boxes to B*-Tree proved to be trickier than anticipated. When done incorrectly, layout errors would appear, but often they could not be detected until the synthetic image of the full 3-D scene was finished. Common errors were collisions between components in the 3-D rendering, and transposition of coordinate-axes. These were easily repaired by manually adjusting the input to the B*-Tree Floorplanner software and tweaking the KML Generator data as needed. As a result of the experiment, we identified a process that should work more smoothly in the future but that presently still requires human inspection of results and manual correction of errors.

VI. Ground Truth

Ground truth for the Benchmark Imagery Suite consists of annotated overlays that assign labels to some or all of the industrial features in the scenes. The words used for the labels come from the constrained vocabulary defined previously. (See Appendix B.) For the real images, the annotation is based on human inspection and interpretation of objects in the images. For the synthetic images, the annotation is based on the perfect knowledge of what the models in the scene represent. Two classes of annotation are provided: point labels and area labels. The former are labels that have been tagged to a single pixel in a single object in the scene, one label per object. The latter are images in which all pixels in the scene have been classified in accordance with the constrained vocabulary, the images have been segmented, and each type of object has been color coded. This section describes the four cases of annotation available in the Benchmark Imagery Suite:

- Real Images: Selective Point Annotation (All Object Classes)
- Real Images: Exhaustive Point Annotation (Buildings Only)
- Real Images: Exhaustive Areal Annotation (All Object Classes) — Posterization
- Synthetic Images: Exhaustive Areal Annotation (All Object Classes) — Posterization

REAL IMAGES: SELECTIVE POINT ANNOTATION (ALL OBJECT CLASSES)

100% of the real images in the Benchmark Imagery Suite were annotated point-wise. To do the labeling, we divided the images among six annotators, with at least two people being responsible for each image. To speed up the annotation process, we used a point-and-click software tool developed by LANL, Geospatial Image ANalysis Tool (GIANT). Although other tools were considered (Annotate, Annotation Tool, Daeja View One, ERDAS, Fototagger, ImageJ, Intelligent Scissors, Label Me, Light Room Abode Photoshop, Photostuff, Snagit, Szoter, Text-Image Linking Environment [TILE]), GIANT had the best mix of affordability and user features of interest. The tool allowed users to point to an object in the image and apply a label from a drop-down menu that contained the controlled vocabulary. Objects were labeled by opening an image in GIANT, using the mouse to select a single pixel within the boundary of each object to be labeled, and saving the resulting geospatial annotation file (.gaf). GAF files are IDL binary structure files with a list of (x, y) pixel coordinates and a label next to the coordinate pair. GIANT provides a tool to convert this binary format into a comma separated value file format which can be imported into other software for further processing, if desired. This tool was used to provide these data in CSV format within the Benchmark Imagery Suite.

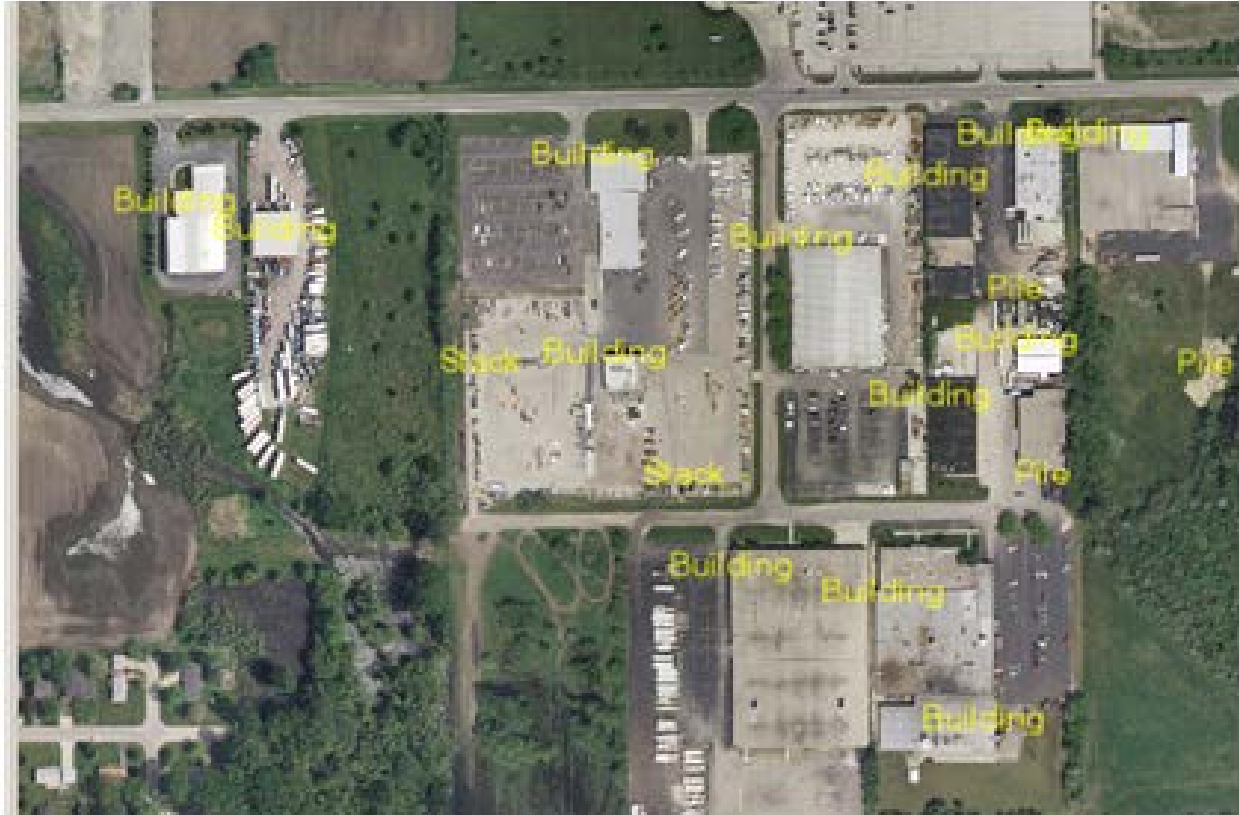


Figure 16. Portion of an annotated image showing the labels superimposed on the scene, Heavy Manufacturing Plant, HM5.

Annotators were instructed to label 30 to 50 objects of their choosing in each image. Thus, most pixels in the image had no label attached to them. (The pixel classification was selective, not exhaustive, across an image.) After all of the images were annotated this way, there were two to three sets of annotations for each image. The annotations for each separate image were then merged by one of the annotators, and the final annotated images were checked for correctness by a trained image analyst. No attempt was made to segment the annotated objects in the images by drawing borders around them. A final quality assurance pass was made to eliminate double labeling of objects and to center the position of each label on its corresponding object. Figure 16 shows a portion of an image that has been selectively annotated point-wise with GIANT.

REAL IMAGES: EXHAUSTIVE POINT ANNOTATION (BUILDINGS ONLY)

12 of the real images were annotated exhaustively in a point-wise manner. In these images, only the buildings were labeled. Other objects were ignored. The motivation to label the images exhaustively arose as a result of previously creating a simple “building finder” algorithm with which to experiment on the set of real images. Results of testing the algorithm on the selectively annotated images suggested that using sparse, point-wise annotation was not an effective way to evaluate that particular algorithm. We concluded that exhaustive annotation was required. As a first cut at exhaustively annotating the real images, we attempted to label all the buildings in a subset of the real images. Buildings less than the size of a trailer home were ignored. Multiple annotations were placed for objects with multi-faceted roofs, due to changes in lighting which generally cause segmentation algorithms to create multiple polygons for any single roof. In addition, we added two new classes of building object, “house” and “industrial building.” The

simple term “building” was then redefined to mean any building other than a house or an industrial building. Fig. 17 shows an example of this type of annotation for a residential neighborhood in the vicinity of site HM2_VA6.



Figure 17. Example of a portion of an exhaustively annotated image (buildings only), HM2_VA6.

Exhaustive annotation of buildings by hand proved to be a slow, tedious process, and we limited this exercise to 12 Heavy Manufacturing (HM) sites, as shown below in Table 5. For each of the 12 images, the list of locations of the labeled objects is included in the Benchmark Imagery Suite in a subfolder of the corresponding image as a file in comma-separated-value (CSV) format. (See the subfolder “Exhaustive_Point_Annotation” in each of the 12 folders listed below.)

HM2_VA6	HM28_VGA	HM36_VGA	HM41_V5Z
HM52_VBV	HM53_VBJ	HM77_VAW	HM80_VBG
HM82_VAY	HM86_VC5	HM92_VCK	HM104_U4X

Table 5. List of 12 real-image files with exhaustive point-wise annotation.

REAL IMAGES: EXHAUSTIVE AREAL SEGMENTATION (ALL OBJECT CLASSES) — POSTERIZATION

52 of the real images in version 1.1a of the Benchmark Imagery Suite have been exhaustively segmented, or “posterized.” (A posterized image is one in which each pixel has been classified and labeled by color coding it according to its class membership.) A group of students and a post-doc from Old Dominion University under the leadership of Prof. Khan Iftokharuddin manually segmented and labeled the 52 real images with GIMP by color coding each identified object. [33] All object classes from the constrained vocabulary were used for selecting labels.

Figure 18 shows an example of a posterized real image for site CP2_U3Z, the same site discussed in the previous section on lidar. (See Figure 8 for comparison.) Table 6 lists all of the images that ODU segmented and color coded. Table 7 provides the color code associated with each object class in the constrained vocabulary.

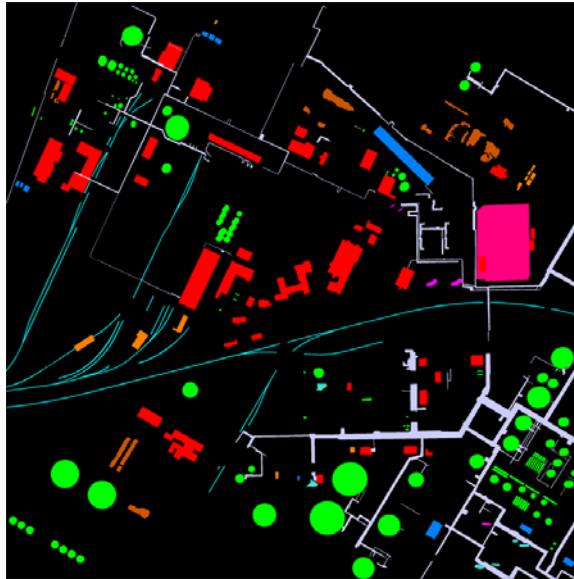


Figure 18. Posterized image of site CP2_U3Z. (See Fig 8.)

CP2_U3Z	HP0003_V80	SI0005_U62
CP3_U5W	HP0031_U44	SI0006_U62
CP5_VGA	HP0034_U4X	SI0013_U62
CP7_VD6	HP0040_VCY	SI0019_V7Q
CP14_P15	HP0044_VDN	SI0020_U4H
CP35_P17	HP0047_C87	SI0021_U3B
CP38_P16	HP0049_C87	SI0027_U62
CP39_VIY	HP0050_VGA	SI0030_U62
CP42_V24	HP0056_C87	
CP43_P16	HP0058_C87	
CP44_P17	HP0060_VIT	
CP51_U62	HP0062_V5D	
CP58_VD6	HP0068_C88	
CP61_VBI	HP0078_V80	
CP78_U42	HP0087_VCA	
CP81_VD6	HP0107_U30	
CP86_VCP	HP0110_V5Q	
CP93_V7Z	HP0112_VF7	
CP98_U43	HP0115_VCS	
	HP0117_VBO	
	HP0119_VAS	
	HP0121_P18	
	HP0142_C88	
	HP0148_V5D	
	HP0155_V7Q	

Table 6. List of 50 real-image files with exhaustive areal segmentation/annotation (“Posterized”).

<i>Object_Name</i>	<i>Color_Name</i>	<i>Red</i>	<i>Green</i>	<i>Blue</i>
Building	Orange	255	127	0
Conveyor	Periwinkle	204	204	255
Cooling Tower	Violet	143	0	255
Cooling Unit	Azure	0	127	255
Electrical Substation	Rose	255	0	127
House	Yellow	255	255	0
Industrial Building	Red	255	0	0
Kiln	Tan	210	180	140
Overhead Crane	Chartreuse	223	255	0
Pile	Burnt Orange	204	85	0
Pond	Blue	0	0	255
Rail line	Cyan	0	255	255
Stack (Industrial Chimney)	Magenta	255	0	255
Tank	Green	0	255	0
Tower	Turquoise	64	224	208
Water Tower	Spring Green	0	255	127

Table 7. Color lookup table for the constrained vocabulary used to posterize real images.

Each posterized real image was vectorized and saved as a shape file in various formats (see Figure 19). Both the posterized annotation image (a raster file) and the vectorized annotation image provide a way for segmentation/annotation algorithms to quantify the position and shape/orientation similarity of their labeled image segments to ground truth.

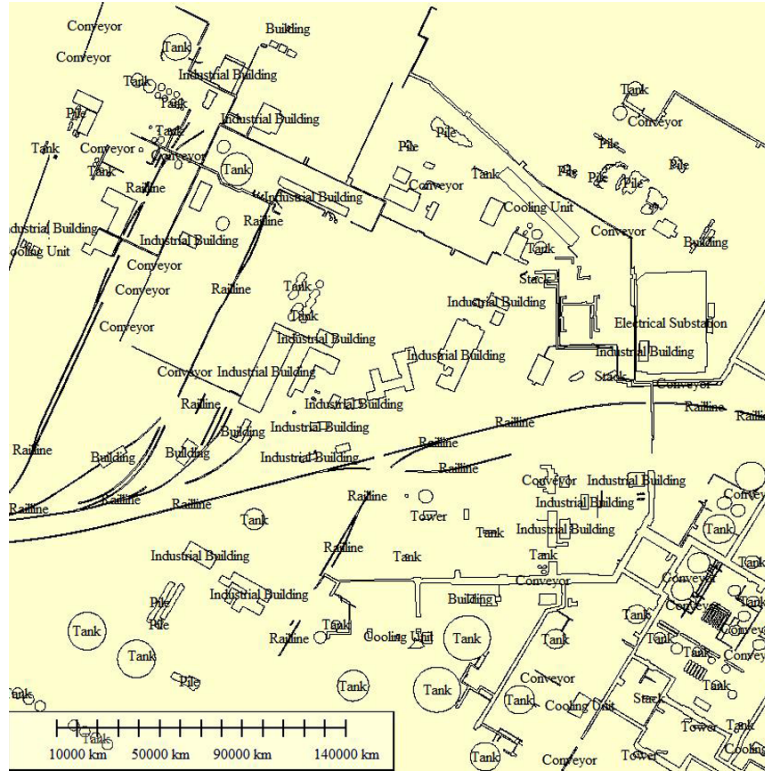


Figure 19. Shape file representation of a posterized real image, CP2_U3Z.

SYNTHETIC IMAGES: EXHAUSTIVE AREAL SEGMENTATION (ALL OBJECT CLASSES) — POSTERIZATION

All 38 of the synthetic images have posterized versions for use as ground truth. There are two minor differences between the posterized synthetic images and the posterized real images. The first difference is an invisible one that has to do with the way the pixel classification and segmentation are imposed. For the real imagery, a human must visually identify the boundaries of each image segment containing an object to be labeled, and then that person must fill the interior region of the segment with a color that corresponds to the class of the object being labeled. In doing this, there is room for subjective errors in determining where the segmentation boundary lies. In the posterized synthetic image, on the other hand, the human plays no direct role in estimating where the edges of an object lie. Instead, SceneWorks will automatically posterize the image with a single button click. The code for posterization in SceneWorks is based on a few key OpenGL calls which are triggered through the scene graph. The posterization is achieved through force overriding the Open Scene Graph (OSG) node of each object to disable texturing of the object. Lighting and shading are also disabled in the node so that there are no shadows. A color is selected for the object by iterating through the objects and selecting a different hue in the RGB color space for each object. Finally, the object is colored by replacing the color arrays on the node and all of its children. The base terrain is rendered all black. The result is automated, perfect segmentation. The color key for each image is a CSV file.

A second difference between the posterized synthetic images and the posterized real images is in the labels and associated color map that are used to identify objects. Every object in a synthetic image that must have a unique position must be handled as a unique entity and therefore requires a unique name within SceneWorks. If there are, for example, two identical looking but separately placeable cooling towers in a scene, they must be given separate names. The constrained vocabulary is too general to identify specific objects when there are multiple, disconnected copies of a given type of object. Therefore, the objects in our synthetic images have multiple levels of abstraction. For example “Hangar-2” would be colored a single color and would be keyed with three different levels of specificity, as shown in Figure 20.

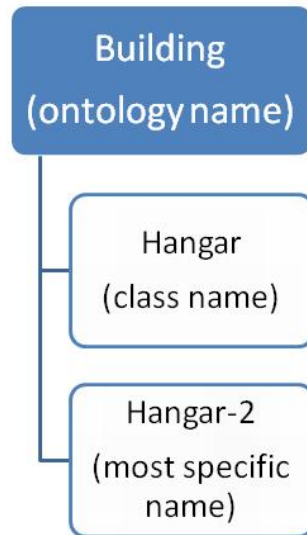


Figure 20. Example of an object, “Hangar-2” with multiple layers of abstraction in its name, increasing in specificity from top to bottom.

Having different levels of object abstraction allows for algorithms to use the posterized images at different levels of detail. For example, if two buildings are connected to each other, some algorithms just mark the complete structure as a “building,” while others segment those two buildings apart and define the specific type of building (e.g. Hangar-1 and Hangar-2.)

To allow for posterization abstraction levels, the objects are named in SceneWorks using the STAGE tool according to the following naming convention: object class first, then the specific object number (*i.e.*, “OBJECT_CLASS-OBJECT_NUMBER”). This enables the posterization engine to create multiple levels of abstraction. A posterized synthetic image has complete ground-truth detail, but it may be abstracted out (made more general) by using the color-key CSV file to associate all objects of a higher classification, as in representing all buildings by a single color instead of rendering the scene with a unique color for each individual building object/model.

As was true for real images, posterized synthetic images have an associated vectorized representation, a shape file. By providing both a rasterized ground truth image and a vectorized version, the Benchmark Imagery Suite accommodates labeling algorithms that produce labeled segments in either of 2 common image representations.

The following figures and table show an example of a synthetic image, its ground truth image in both posterized and vectorized form and the corresponding color look-up table.



Figure 21. A synthetic image of a nuclear power plant in an arid environment.

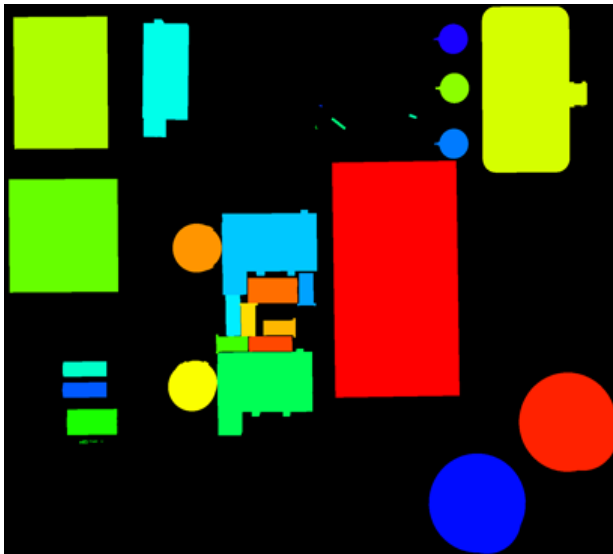


Figure 22. Posterized version of Figure 21.

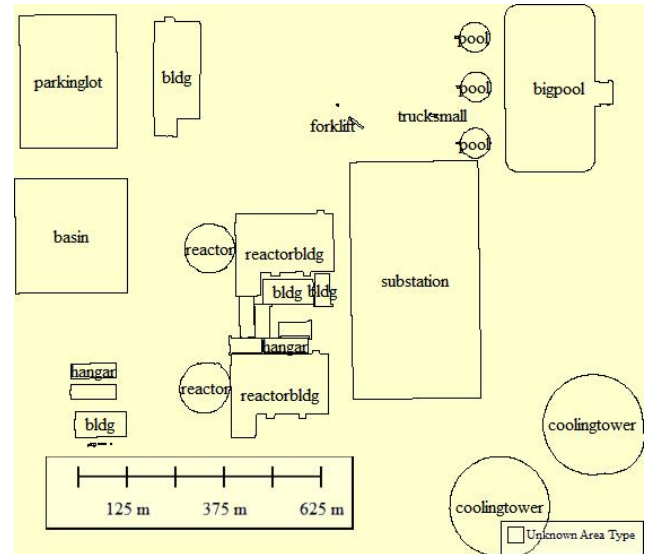


Figure 23. Vectorized version of Figure 22.

<i>Object #</i>	Color Indices			Labels		
	<i>Red</i>	<i>Green</i>	<i>Blue</i>	<i>Ontology Name</i>	<i>Object Class</i>	<i>Object Name</i>
1	0	255	234	Building	bldg	bldg1-1
2	25	255	0	Building	bldg	bldg2-1
3	255	110	0	Building	bldg	bldg2-2
4	64	255	0	Building	bldg	bldg3-1
5	0	162	255	Building	bldg	bldg3-2
6	255	183	0	Building	bldg	bldg3-3
7	255	221	0	Building	bldg	bldg3-4
8	0	255	200	Building	hangar	hangar-1
9	0	89	255	Building	hangar	hangar-2
10	255	72	0	Building	hangar	hangar-3
11	0	238	255	Building	hangar	hangar-4
12	0	255	85	Building	reactorbldg	reactorbldg-1
13	0	200	255	Building	reactorbldg	reactorbldg-2
14	255	34	0	Cooling Tower	coolingtower	coolingtower-1
15	0	12	255	Cooling Tower	coolingtower	coolingtower-2
16	255	0	0	Electrical Substation	substation	substation-1
17	102	255	0	Pond	basin	basin-1
18	213	255	0	Pond	bigpool	bigpool-1
19	0	123	255	Pond	pool	pool2-1
20	25	0	255	Pond	pool	pool2-2
21	140	255	0	Pond	pool	pool2-3
22	251	255	0	Tank	reactor	reactor-1
23	255	149	0	Tank	reactor	reactor-2
24	0	51	255	N/A	forklift	forklift-1
25	0	255	46	N/A	forklift	forklift-2
26	174	255	0	N/A	parkinglot	parkinglot-1
27	0	255	12	N/A	rack	rack-1
28	0	255	123	N/A	truck	truck-1
29	0	255	162	N/A	trucksmall	trucksmall-1

Table 8. Color look-up table for the posterized image shown above in Figure 22.

VII. Demonstration of Testing an Algorithm with Benchmark Imagery

One of the requirements of this project is to demonstrate the use of the Benchmark Imagery Suite in verifying or validating a semantic extraction algorithm. To meet that requirement, we have tested several algorithms. Our objective is not to create and validate algorithms but simply to show how the imagery might be used in the validation process. We broadly envision two types of semantic extraction algorithm. For convenience we refer to them as “Level 1 algorithms” and “Level 2 algorithms.” (Note: this terminology should not be confused with the use of “Level 1” or “Level 2” in referring to quality of data or information as may be used by others. See for example [34].) Level 1 algorithms denote processes that detect human recognizable features in a

scene and assign a label to each feature. An example is an algorithm that segments and automatically annotates an image in such a way that industrial components such as individual buildings, piles or electrical substations are located and labeled in an image of, say, a coal-fired electrical power plant. Level 2 algorithms are those that analyze the labels and the spatial (or spatio-temporal) inter-relationships among them to infer that they represent (in this example) a power plant. In a nutshell, Level 1 algorithms use images as their input. Level 2 algorithms use the output of Level 1 algorithms as their input.

Early in the project, we chose to focus on Level 1 algorithms. We expected that it would be straightforward to obtain automated annotation software and to demonstrate the use of the Benchmark Imagery Suite with it. However, that proved not to be the case. We therefore wrote a simplistic algorithm to locate and label buildings in overhead images of industrial facilities, and we tested that algorithm with the Benchmark Imagery Suite. From that initial exercise, we drew two top-level conclusions. First, the simplistic algorithm we wrote, while useful for demonstrating a testing methodology, was not suitable for further development. Second, we determined that exhaustively annotating the images, rather than selectively annotating point-wise, would be highly desirable.

We undertook exhaustive annotation of the real imagery. The task was very labor intensive, and we recognized that we would have to restrict exhaustive annotation to only a subset of the real images. We performed two types of exhaustive annotation. In one set of images we labeled every kind of building in the scene pointwise. In another set we segmented and color coded every object in the scene that was defined in the constrained vocabulary of Table 2 in Section II. This process we called “areal annotation” (as opposed to “point annotation”). At the conclusion of the exhaustive annotation process we had a total of four types of annotation, as discussed in the previous section on Ground Truth: (a) selective point annotation of all the real images (30 to 100 objects labeled per image), (b) exhaustive point annotation of some of the real images (12 images), (c) exhaustive areal annotation of some of the real images (51 images), and (d) exhaustive areal annotation of all of the synthetic images (38 images).

In the final year of the project, we used a new algorithm for finding buildings, an adapted version of LANL’s patented RADIUS algorithm [38]. We ran a variety of tests to demonstrate using the various types of annotation to test the new building finder algorithm. These tests are discussed in the remainder of this section of the document. In addition, in the process of preparing to test the RADIUS-derived algorithm, we found it necessary to create shape files from the well segmented, areal annotated images. In addition to providing training and test images, the resulting shape files were suitable for use in testing Level 2 algorithms. Although we attempted towards the end of the project to exchange files with other researchers who were developing a web based package for extracting semantic inferences on shape files [35], there were interfacing difficulties and insufficient time remaining on the Benchmark Imagery Project to see this effort through. It remains a topic of future interest.

We now discuss the major steps involved in using our imagery for testing algorithms.

1. Define V&V methodology
2. Obtain algorithm to test
3. Train & Test algorithm using the Benchmark Imagery Suite
4. Interpret results from the V&V tests

DEFINE V&V METHODOLOGY.

In previous years, NA-22 sponsored a GeoSpatial Validation (GSV) working group that recommended a protocol for validating geospatial algorithms. [36] We have chosen to use the V&V methodology from that group.

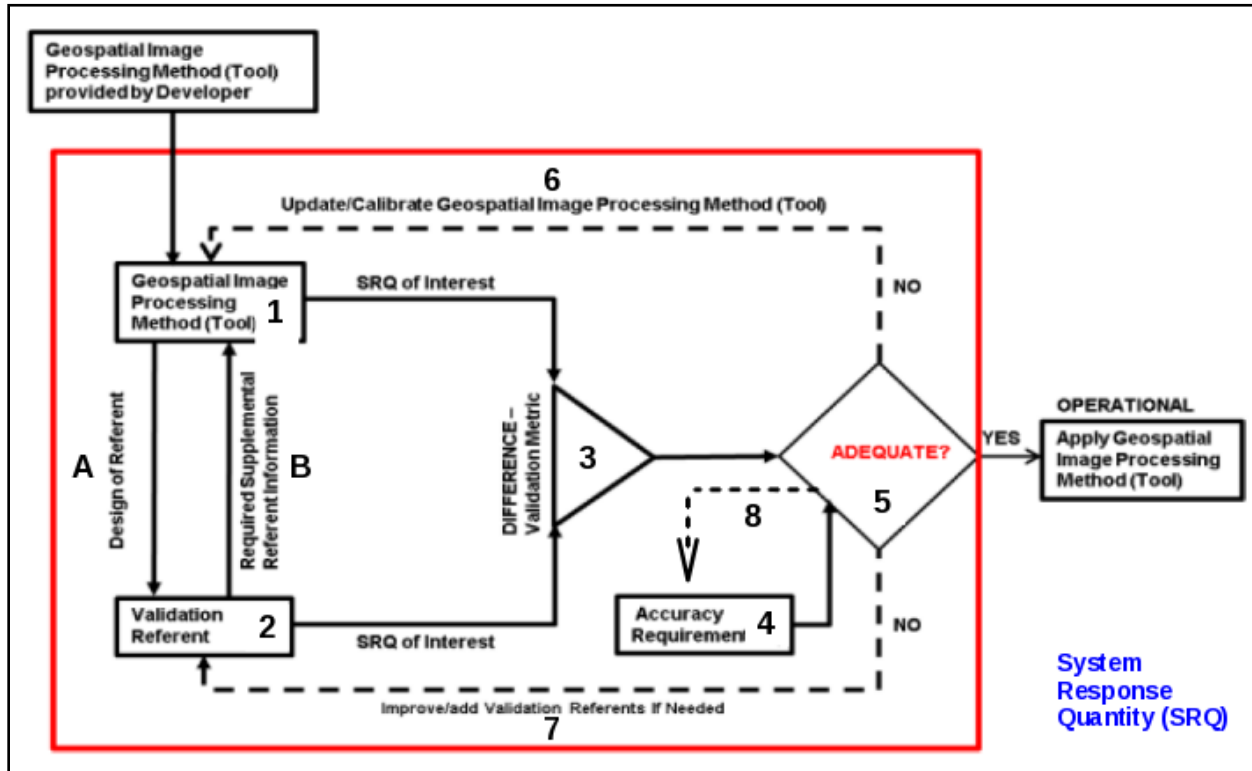


Figure 24. Algorithm verification and validation cycle proposed by the NA-22 GSV Working Group.

Shown in the figure, an algorithm (Box 1) is supplied by a developer and placed into validation testing (red box). It is then given input from a source of ground truth, or “Validation Referent” (Box 2). The algorithm calculates a result that can be measured, a “System Response Quantity (SRQ).” A geospatial SRQ could be, for example, a list of all buildings in a set of images and the corresponding geocoordinates and areal footprints. The SRQ computed by the algorithm is then compared to the pre-tabulated, true value (Box 3). In this report, we refer to the differences between the algorithmic outputs and the ground truth values as “validation metrics.” In the example just mentioned, validation metrics could include the difference in the number of buildings identified vis-à-vis the actual number or the position and size errors of the detected buildings. The validation metrics from Box 3 are then compared with a set of Accuracy Requirements (Box 4). If the validation metrics for the algorithm are within acceptable limits, then the algorithm is considered to be validated for the SRQs being tested (Box 5). Otherwise, the algorithm is rejected.

As drawn, the V&V process flow applies to an R&D environment, where a failure of validation might not indicate a problem only with the algorithm. Rather, an unsuccessful validation test could be a result of errors either in the ground truth, the accuracy requirements, or the algorithm, requiring improvements in any or all of them. The three dashed lines numbered 6, 7 and 8 in

Figure 24 accommodates feedback to support the R&D validation cycle. This situation applies to the Benchmark Imagery Project, which is a research and development effort.

(Note that in situations in which a set of validation benchmarks and acceptance requirements have been adopted as standards, a failure to validate is attributed entirely to having a bad algorithm, and the algorithm is flatly rejected.)

OBTAIN ALGORITHM TO TEST.

At the beginning of this project, we planned to use existing commercial software such as ERDAS Imagine, Arc GIS or eCognition for demonstrating the use of the benchmark imagery for V&V [37]. Upon investigation, we concluded that such software was not of interest, due to the high purchase price, steep learning curve and lack of automation. We sought to find more-automated software if possible. As the project developed, it became apparent that automated or semi-automated algorithms for identifying and classifying industrial facilities from overhead images were not available. Therefore, to demonstrate the process of using our current suite of images for V&V, we created our own algorithm, a simplistic “building finder” (BldgFinder). Since the purpose of the demonstration was not for us to develop an algorithm but to illustrate the use of our benchmark images in the V&V process, it was not required that the algorithm be robust. Additionally, this work establishes a benchmark so that the performance of other algorithms can be compared against the performance described here.

The semantic aspect of the algorithm was that it attempted to match a word from the constrained annotation vocabulary (Table 2) with objects in images. “Building” was chosen as the particular word to match, because buildings were a key industrial feature of all images in our data, whether real or synthetic. The geospatial aspect of the algorithm was that it used both the shape characteristics of the polygons created through segmentation and the spatial resolution (i.e., the ground sample distance [GSD]) of each image to calculate the area of the segmentation polygons.

The BldgFinder algorithm was developed based on LANL’s RADIUS image segmentation technology [38] and a pre-defined “buildingness” metric. The algorithm is a binary classifier; therefore, its performance was evaluated in terms of detecting the object of interest (i.e., “building”) as well as objects which were not of interest, (i.e., “non-building”), and its ability to avoid confusion between these two classes [39]. The “buildingness” metric was calculated based on several polygon shape characteristics (e.g., perimeter complexity) and the ground sample distance (GSD) of the image being segmented. The threshold value for determining whether a polygon represented a building (buildingness > threshold), called a “candidate polygon,” or was not a building (buildingness < threshold), called a “non-candidate polygon,” was set based on training data. Only candidate and non-candidate polygons which overlapped the ground truth were used in assessing algorithm performance. A wrapper written around the algorithm permitted batch operation of the code for efficiently testing large numbers of images. Further details of the BldgFinder algorithm are not provided here because the secondary objective of the Benchmark Imagery project was not to develop a semantic capability with a high probability of detection and low false alarm rate. Rather it was to illustrate use of the benchmark imagery suite, and inherently establishes a nominal performance benchmark for other algorithms to compare against.

TRAIN & TEST ALGORITHM USING THE BENCHMARK IMAGERY SUITE

The V&V demonstration consisted of two main activities: training and testing. Five tests were performed in order to demonstrate use of the benchmark imagery suite for training and testing a semantic algorithm (i.e., the BldgFinder algorithm), and to establish a nominal performance benchmark for other algorithms to compare against. Imagery and ground truth used to train and test the BldgFinder algorithm came from the Benchmark Imagery Suite v1.1. Error matrices (a.k.a., “confusion matrix,” or “contingency table”) [40] were created by comparing the BldgFinder classification results to the ground truth information, as illustrated on the next page (Figure 25). Next, commonly used performance metrics (e.g., overall accuracy) [41] were calculated from the error matrix values as a means to assess the algorithm’s performance. Finally, all of this information was then aggregated into a performance table associated with each test.

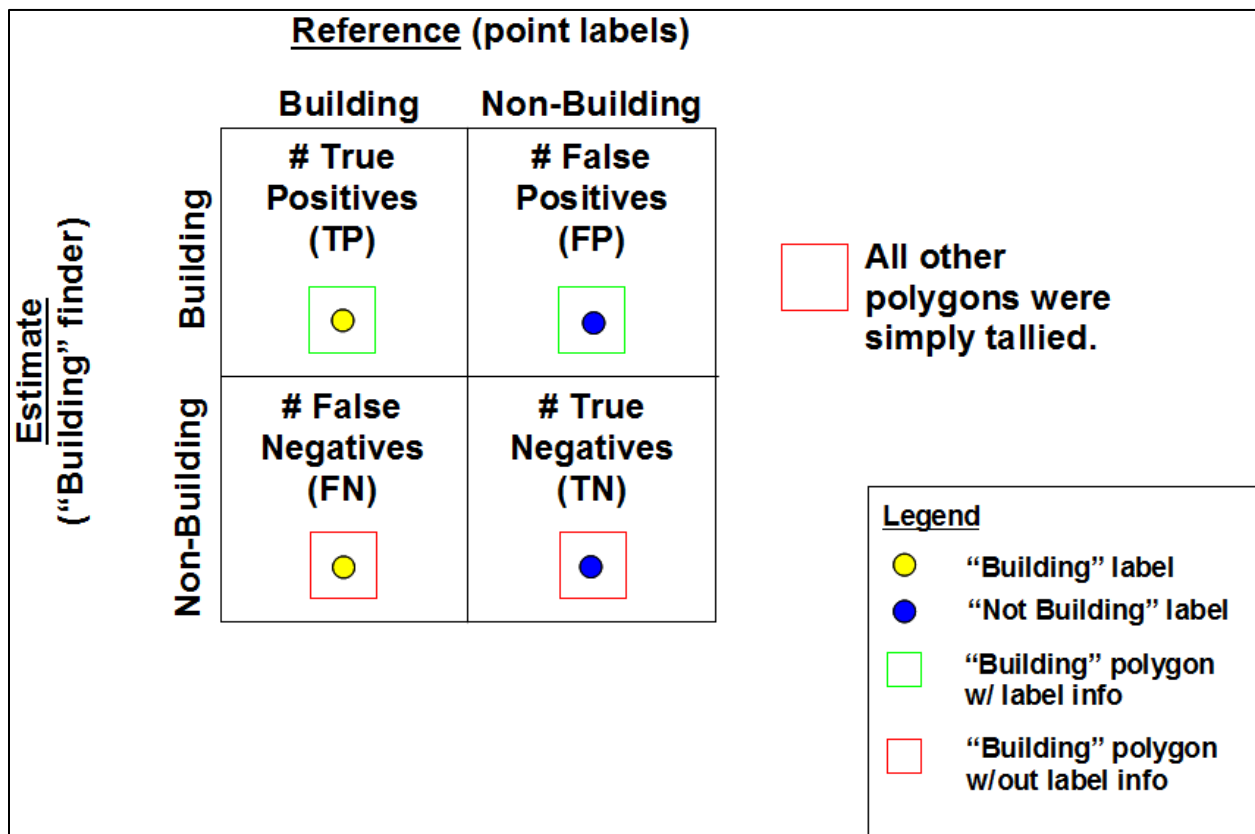


Figure 25. Illustration of comparison of BldgFinder candidate polygons to ground truth (in this case, point annotations) to create an error matrix.

The “ground truth” information consisted of both point and polygon information. Two types of point annotation were available, “sparse” and “exhaustive.” An object label (text string) was attached to each point location. The sparse point annotations utilized the controlled vocabulary from the photo-interpretation guide (Appendix B [42]), and were available for every real image, while the exhaustive point annotations used a subset of this vocabulary consisting of building types only (i.e., “building,” industrial building,” and “house), and were available only for twelve of the real images. Polygonal truth information consisted of object boundaries (represented as polygons) and their labels (represented as a text string attached to each polygon via a polygon’s

unique ID). This information was derived by performing a raster-to-vector conversion of the manual annotation masks (for the real imagery) or the posterization masks (for the synthetic imagery), referred to in this report as “manual polygons” and “posterization polygons,” respectively.

Each row of a performance table contains the error matrix results, performance metrics, and other information obtained by running the BldgFinder algorithm on a single image. The final row contains the cumulative result. There are sixteen columns in each performance table. The definition of each column is discussed next [40].

Name is the image name used for testing. If real imagery was used, the name is the alphanumeric site name and the image number, separated by a backslash (“\”). If synthetic imagery was used, then the name is simply the synthetic image name.

All Markups (AM) is the total number of object polygons, that is, the sum of all true building and true non-building points or polygons.

Building Markups (BM) is the total number of true building points or polygons.

%Bldgs is the ratio of Building Markups (BM) to All Markups (AM) expressed as a percentage (i.e., $\%Bldgs = BM/AM$).

True Positives (TP) is number of correctly detected building polygons as compared to the total number of true building points or polygons. For the point annotations, a true building polygon is the same as a candidate building polygon if the point annotation falls within the candidate building polygon. A correct detection is a candidate building polygon which significantly overlaps a true building polygon. Significant overlap was defined as a situation where the candidate polygon covered at least 30% of the area of the true building polygon. A true building polygon was only used once in calculating TP. That is, if multiple candidate building polygons overlapped a single true building polygon, then only the candidate building polygon with the highest overlap was used in calculating TP. Unused candidate polygons were simply tallied (see “Building Candidate (BC)”). Note that true positives (TP) plus false negatives (FN) equals the total number of true building polygons (i.e., the number of building markups (BM); $BM = TP+FN$).

False Positives (FP) is the number of incorrectly detected building polygons as compared to the total number of true building points or polygons. The definition of significant overlap and the single use of a true building polygon to calculate FP were the same as those used to calculate TP.

True Negatives (TN) is the number of correctly detected non-building (non-candidate) polygons as compared to the total number of true non-building points or polygons. The definition of significant overlap and the single use of a true non-building polygon to calculate TN were analogous to those used to calculate TP.

False Negatives (FN) is the number of incorrectly detected non-building (non-candidate) polygons as compared to the total number of true non-building points or polygons. The definition of significant overlap and the single use of a true non-building polygon to calculate FN were analogous to those used to calculate TP.

Building Candidates (BC) is the total number of candidate building polygons which were not accounted for as being a TP, FP, TN, or FN.

Overall Accuracy (OA) is calculated as $(TP+TN)/(TP+TN+FP+FN)$.

User's Accuracy for buildings (UAB) (a.k.a. “Precision” or “Positive Predictive Value”) is calculated as $TP/(TP+FP)$.

User's Accuracy for non-buildings (UANB) (a.k.a. “Negative Predictive Value”) is calculated as $TN/(TN+FN)$.

Producer's Accuracy for buildings (PAB) (a.k.a. “Recall” or “Sensitivity”) is calculated as $TP/(TP+FN)$. Note that this is the same as TP/BM since $TP+FN$ equals BM (i.e., $BM = TP+FN$).

Producer's Accuracy for non-building (PANB) (a.k.a. “Specificity”) is calculated as $TN/(TN+FP)$.

False Alarm Rate (FAR) is calculated as $FP/(FP+TN)$.

False Discovery Rate (FDR) is calculated as $FP/(FP+TP)$.

In the first test, the BldgFinder algorithm was trained by using a subset of the sparse point annotations created through manual markup of the real imagery (Table 9). Then the trained algorithm was tested on a separate set of imagery, using the corresponding sparse point annotations as truth (Table 10).

HM86_VC5\87432373
HM92_VCK\61736136
HM104_U4X\74599487
HM28_VGA\20407034
HM36_VGA\96388801
HM41_V5Z\79179280
HM83_C87\41129690
HM17_V5Z\43410911
HM5_VGO\99648675
HM6_V6L\53663767
HM7_U52\37899384
HM10_V77\35858425
HM59_VBU\33312430
HM63_C87\51214994
HM64_VAY\08084046
HM75_VAT\08995856

Table 9. List of real image names (.tif) and their corresponding point annotation files (.csv) used for training the BldgFinder algorithm (where Bldg = “Building”)

Name	AM	BM	%Bldgs	TP	FP	TN	FN	BC	OA	UAB	UANB	PAB	PANB	FAR	FDR
HM77_VAW\03996037	109	60	55.0%	54	37	12	6	16624	60.6%	59.3%	66.7%	90.0%	24.5%	75.5%	40.7%
HM82_VAY\08709672	11	8	72.7%	8	0	3	0	22113	100.0%	100.0%	100.0%	100.0%	100.0%	0.0%	0.0%
HM56_VBV\71522146	39	34	87.2%	25	5	0	9	16299	64.1%	83.3%	0.0%	73.5%	0.0%	100.0%	16.7%
HM45_V3F\13388096	42	30	71.4%	28	6	6	2	36098	81.0%	82.4%	75.0%	93.3%	50.0%	50.0%	17.6%
HM79_VBY\22002232	120	84	70.0%	72	23	13	12	15372	70.8%	75.8%	52.0%	85.7%	36.1%	63.9%	24.2%
HM31_C87\27779561	45	17	37.8%	13	17	11	4	15017	53.3%	43.3%	73.3%	76.5%	39.3%	60.7%	56.7%
HM25_VGA\29311232	122	68	55.7%	53	33	21	15	19163	60.7%	61.6%	58.3%	77.9%	38.9%	61.1%	38.4%
HM70_VC3\32899678	22	11	50.0%	11	6	5	0	16654	72.7%	64.7%	100.0%	100.0%	45.5%	54.5%	35.3%
HM103_U4X\51202931	26	24	92.3%	21	2	0	3	28889	80.8%	91.3%	0.0%	87.5%	0.0%	100.0%	8.7%
HM80_VBG\54787849	34	12	35.3%	10	18	4	2	19876	41.2%	35.7%	66.7%	83.3%	18.2%	81.8%	64.3%
HM2_VA6\64476554	54	13	24.1%	13	18	23	0	30313	66.7%	41.9%	100.0%	100.0%	56.1%	43.9%	58.1%
HM52_VBV\13363470	33	21	63.6%	18	6	6	3	16356	72.7%	75.0%	66.7%	85.7%	50.0%	50.0%	25.0%
HM9_V77\75747777	43	24	55.8%	14	8	11	10	61812	58.1%	63.6%	52.4%	58.3%	57.9%	42.1%	36.4%
HM81_VAW\79992491	24	11	45.8%	6	8	5	5	16327	45.8%	42.9%	50.0%	54.5%	38.5%	61.5%	57.1%
HM74_VAT\83434180	84	44	52.4%	43	33	7	1	15346	59.5%	56.6%	87.5%	97.7%	17.5%	82.5%	43.4%
HM53_VBJ\92937432	82	41	50.0%	37	34	7	4	20703	53.7%	52.1%	63.6%	90.2%	17.1%	82.9%	47.9%
Cumulative	890	502	56.4%	426	254	134	76	366962	62.9%	62.6%	63.8%	84.9%	34.5%	65.5%	37.4%

Table 10. Performance table results of applying the BldgFinder algorithm (where Bldg = “Building”), trained using the information listed in Table 9, to the real images listed under “Name,” with their associated sparse point annotations serving as ground truth.

Name
HM28_VGA\20407034
HM92_VCK\61736136
HM104_U4X\74599487
HM41_V5Z\79179280
HM86_VC5\87432373
HM36_VGA\96388801

Table 11. List of real image names (.tif) and their corresponding exhaustive point annotation files (.csv) used for training the BldgFinder algorithm (where Bldg = “Building”)

A second test was performed by using the real imagery which had exhaustive point annotations (a total of twelve real images). Half (six images) were used for training (Table 11) and the other half (six images) were used for testing (Table 12).

Name	AM	BM	%Bldg	TP	FP	TN	FN	BC	OA	UAB	UANB	PAB	PANB	FAR	FDR
HM77_VAW\03996037	576	201	34.9%	153	246	128	48	4713	48.9%	38.3%	72.7%	76.1%	34.2%	65.8%	61.7%
HM82_VAY\08709672	31	7	22.6%	6	18	5	1	3430	36.7%	25.0%	83.3%	85.7%	21.7%	78.3%	75.0%
HM80_VBJ\54787849	395	82	20.8%	59	176	136	23	2715	49.5%	25.1%	85.5%	72.0%	43.6%	56.4%	74.9%
HM2_VA6\64476554	86	24	27.9%	16	22	39	8	2630	64.7%	42.1%	83.0%	66.7%	63.9%	36.1%	57.9%
HM52_VBV\71522146	227	70	30.8%	55	92	64	15	1947	52.7%	37.4%	81.0%	78.6%	41.0%	59.0%	62.6%
HM53_VBJ\92937432	466	144	30.9%	112	155	166	32	4328	59.8%	41.9%	83.8%	77.8%	51.7%	48.3%	58.1%
Cumulative	1781	528	29.6%	401	709	538	127	19763	52.9%	36.1%	80.9%	75.9%	43.1%	56.9%	63.9%

Table 12. Performance table results of applying the BldgFinder algorithm (where Bldg = “Building”), trained using the information listed in Table 11, to the real images listed under “Name,” with their associated exhaustive point annotations serving as ground truth.

A third test was performed by using a subset of the areal (polygon) annotations created through manual markup of the real imagery (i.e., manual polygons) to train the BldgFinder algorithm (Table 13). Then the trained algorithm was tested on a separate set of imagery, using their corresponding, manually-defined polygonal annotations (manual polygons) as truth (Table 14).

Name
CP61_VBI\14832847
CP58_VD6\23623821
HP0062_V5D\37992661
SI0027_U62\49362802
HP0060_VIT\60281917
SI0005_U62\60978124
HP0155_V7Q\74045776
HP0117_VBO\78086512
CP42_V24\79212324
CP98_U43\94448391
HP0148_V5D\94942228

Table 13. List of real image names (.tif) corresponding with the areal (polygonal) annotation files (_annotations.shp) used for training the BldgFinder algorithm (where Bldg = “Building” and “Industrial Building”).

Name	AM	BM	%Bldg	TP	FP	TN	FN	BC	OA	UAB	UANB	PAB	PANB	FAR	FDR
SI0030_U62\02725713	563	187	33.2%	119	282	94	68	11696	37.8%	29.7%	58.0%	63.6%	25.0%	75.0%	70.3%
CP93_V7Z\22319506	210	30	14.3%	27	138	42	3	15034	32.9%	16.4%	93.3%	90.0%	23.3%	76.7%	83.6%
HP0040_VCY\32675142	386	65	16.8%	50	152	169	15	12011	56.7%	24.8%	91.8%	76.9%	52.6%	47.4%	75.2%
CP2_U3Z\44608466	351	57	16.2%	53	142	152	4	13986	58.4%	27.2%	97.4%	93.0%	51.7%	48.3%	72.8%
CP81_VD6\52063269	83	19	22.9%	9	27	37	10	8061	55.4%	25.0%	78.7%	47.4%	57.8%	42.2%	75.0%
HP0078_V80\62728946	100	41	41.0%	35	28	31	6	8361	66.0%	55.6%	83.8%	85.4%	52.5%	47.5%	44.4%
SI0013_U62\74196305	43	28	65.1%	26	10	5	2	8883	72.1%	72.2%	71.4%	92.9%	33.3%	66.7%	27.8%
HP0058_C87\97213252	154	33	21.4%	23	74	47	10	26982	45.5%	23.7%	82.5%	69.7%	38.8%	61.2%	76.3%
Cumulative	1890	460	24.3%	342	853	577	118	105014	48.6%	28.6%	83.0%	74.3%	40.3%	59.7%	71.4%

Table 14. Performance table results of applying the BldgFinder algorithm (where Bldg = “Building” and “Industrial Building”), trained using the information listed in Table 13, to the real images listed under “Name,” with their associated posterization polygons serving as ground truth.

A fourth test was performed by applying the BldgFinder algorithm, as trained in the third test, to the synthetic imagery from the “two types four sites” category, using the corresponding polygon annotations (posterization polygons) as truth (Table 15).

Name	AM	BM	%Bldgs	TP	FP	TN	FN	BC	OA	UAB	UANB	PAB	PANB	FAR	FDR
arid_coal	210	19	9.0%	13	22	169	6	5460	86.7%	37.1%	96.6%	68.4%	88.5%	11.5%	62.9%
arid_nuc	38	15	39.5%	12	5	18	3	4666	78.9%	70.6%	85.7%	80.0%	78.3%	21.7%	29.4%
denver_coal	155	24	15.5%	7	16	115	17	12111	78.7%	30.4%	87.1%	29.2%	87.8%	12.2%	69.6%
denver_nuc	46	15	32.6%	15	8	23	0	15440	82.6%	65.2%	100.0%	100.0%	74.2%	25.8%	34.8%
forest_coal	1133	41	3.6%	14	30	1062	27	3601	95.0%	31.8%	97.5%	34.1%	97.3%	2.7%	68.2%
forest_nuc	40	15	37.5%	12	6	19	3	8723	77.5%	66.7%	86.4%	80.0%	76.0%	24.0%	33.3%
ocean_coal	597	39	6.5%	12	27	510	48	11494	87.4%	30.8%	91.4%	20.0%	95.0%	5.0%	69.2%
ocean_nuc	55	15	27.3%	13	7	33	2	11607	83.6%	65.0%	94.3%	86.7%	82.5%	17.5%	35.0%
Cumulative	2274	183	8.0%	98	121	1949	106	73102	90.0%	44.7%	94.8%	48.0%	94.2%	5.8%	55.3%

Table 15. Performance table results of applying the BldgFinder algorithm (where Bldg = 'bldg', 'hangar', 'hangarsmall', 'reactor', 'reactorbldg', 'smallbldg', and 'stackbldg'), trained using the information listed in Table 13, to the “two types four sites” synthetic images listed under “Name,” with their associated posterization polygons serving as ground truth.

Finally, a fifth test was performed by applying the BldgFinder algorithm, as trained in the third test, to the synthetic imagery from the “arid_nuc” category, using the corresponding polygon annotations (posterization polygons) as truth (Table 16).

Name	AM	BM	%Bldg	TP	FP	TN	FN	BC	OA	UAB	UANB	PAB	PANB	FAR	FDR
arid_nuc_gen1	27	15	55.6%	15	3	9	0	3648	88.9%	83.3%	100.0%	100.0%	75.0%	25.0%	16.7%
arid_nuc_gen2	26	17	65.4%	15	5	4	2	5516	73.1%	75.0%	66.7%	88.2%	44.4%	55.6%	25.0%
arid_nuc_gen3	31	22	71.0%	10	6	3	12	5423	41.9%	62.5%	20.0%	45.5%	33.3%	66.7%	37.5%
arid_nuc_gen4	24	15	62.5%	14	6	3	1	5323	70.8%	70.0%	75.0%	93.3%	33.3%	66.7%	30.0%
arid_nuc_gen5	26	15	57.7%	11	5	6	4	5396	65.4%	68.8%	60.0%	73.3%	54.5%	45.5%	31.3%
arid_nuc_gen6	30	21	70.0%	12	6	3	9	5329	50.0%	66.7%	25.0%	57.1%	33.3%	66.7%	33.3%
arid_nuc_gen7	30	17	56.7%	13	4	9	4	5397	73.3%	76.5%	69.2%	76.5%	69.2%	30.8%	23.5%
arid_nuc_gen8	42	26	61.9%	15	7	9	11	4866	57.1%	68.2%	45.0%	57.7%	56.3%	43.8%	31.8%
arid_nuc_gen9	27	16	59.3%	11	5	6	5	5361	63.0%	68.8%	54.5%	68.8%	54.5%	45.5%	31.3%
arid_nuc_gen10	30	18	60.0%	12	4	8	6	5618	66.7%	75.0%	57.1%	66.7%	66.7%	33.3%	25.0%
Cumulative	293	182	62.1%	128	51	60	54	51877	64.2%	71.5%	52.6%	70.3%	54.1%	45.9%	28.5%

Table 16. Performance table results of applying the BldgFinder algorithm (where Bldg = 'bldg', 'hangar', 'hangarsmall', 'reactor', 'reactorbldg', 'smallbldg', and 'stackbldg'), trained using the information listed in Table 13, to the "arid_nuc" synthetic images listed under "Name," with their associated posterization polygons serving as ground truth.

INTERPRET RESULTS FROM THE V&V TESTS

In this section, we use the testing results from the five tests to draw inferences about the benchmark imagery and about the algorithm. Only general trends are discussed here because the number of comparisons which could be performed is large, given the number of tests and especially the number of performance metrics.

The first observation is that the PAB was high ($\sim > 75\%$) for all tests when using real imagery (Tables 10, 12, and 14). This suggests that, while nascent, the BldgFinder algorithm shows promise as a means to automatically detection buildings within synoptic imagery.

The second observation is that the FAR was high ($\sim > 40\%$) when real imagery was used (Tables 10, 12, and 14) and when the "arid_nuc" synthetic imagery was used (Table 16), but was relatively quite low (6%) when the "two types four sites" synthetic imagery was used (Table 15). This suggests that the "two types four sites" synthetic imagery requires modification so that algorithm performance is more similar to the results achieved by the other tests. Adding more clutter would be a logical first step.

The third observation is that the availability of exhaustive annotations did not improve the algorithm's performance. It was expected that using exhaustive point annotations would lead to higher accuracy. The PAB dropped from 85% to 76%, although the FAR did decrease from 66% to 57% (Tables 10 and 12).

The fourth observation is that, for the "two types four sites" synthetic imagery, the PAB was high ($> 80\%$) when using the "nuc" synthetic imagery, but was low ($< 68\%$) when using the "coal" synthetic imagery. This result suggests that the BldgFinder algorithm performance varies strongly with industry type.

The fifth observation is that, for the "arid_nuc" synthetic imagery, the PAB varied from a low of 45% to a high of 100%. It was anticipated that this test would show that algorithm performance is dependent upon where the objects are placed relative to the background, since the size, number, and shape were expected to be consistent across all ten synthetic images. However, this was found not to be the case (i.e., both AM and BM are not a constant value in Table 16). This

result lead to discovery of a problem with the posterization process, which is the cause of inconsistency in the size and number of posterization polygons. This is an example of how testing an algorithm can provide insight into how the benchmark imagery suite can be improved.

A sixth and final observation is that the number of building candidates (BC) was always very high (two or more orders of magnitude) when compared to the number of building markups (BM). This is due to the fact that the buildingness metric is purely spatial nature; therefore, it includes road fragments, shadows, parking lots, sidewalks, and other man-made structures that are rectilinear. While the ground truth in the benchmark imagery suite meets the need to train on the spatial attributes of buildings, currently there is no more information available that can be reliably used to narrow down building candidate polygons. The spectral (i.e., natural color) attribute of buildings is highly variable, and their aspect ratio is likewise highly variable (there are long skinny buildings as well as square-like ones). A promising cue is height above ground level, which can be provided by lidar. The current literature concerning the combination of natural color and lidar modalities indicates that polygons associated with roads, shadows, parking lots, and other low-relief features would likely be eliminated from the set of candidate polygons [41], thereby reducing the BC count.

In the V&V demonstration performed with the Benchmark Imagery Suite v1.1, we created and tested a nascent building-finder algorithm. Our validation metrics were also simple. Furthermore, we had no user-supplied accuracy requirements on the algorithm. In the future, other characteristics such as annotation confidence, image quality, amount of clutter, etc. might also prove useful for conducting V&V testing [10].

So far, we have emphasized how the V&V demo led to recommendations about the Benchmark Imagery suite. Before we close this section, though, we comment briefly on the algorithm that we used. We attribute much of the inaccuracy of the building finder algorithm first to the limited spectral resolution (3-bands) of the training data and second to the fact that the algorithm only made use of the visual portion of the electromagnetic spectrum. Buildings are not spectrally unique in the visual wavelength region of the electromagnetic spectrum. This leads to inaccurate segmentation. Next, the threshold value for the buildingness parameter was not varied. Often in defining a detection algorithm one varies the threshold to determine the value that optimizes the ratio of the probability of detection to the probability of a false alarm. Plotting the probability of detection against the false alarm rate results in a Receiver Operating Characteristic (ROC) Curve, a commonly used performance diagnostic for comparing algorithms. That was not done in this study, as our objective was not to optimize the performance of the algorithm but to use it as a vehicle for stepping through the process of V&V

REMARKS AND SUGGESTIONS ABOUT THE BENCHMARK IMAGERY SUITE

The images and annotations provided in the Benchmark Imagery suite proved useful for purposes of making a decision about the validity of a semantic algorithm.

The current set of images appeared to fully exercise the algorithm.

Exhaustive point annotation does not necessarily improve detection statistics of the algorithm.

Objects should be segregated into those that make up the “figure” (i.e., are part of the industrial facility) and those that make up the “ground” (i.e., are part of the surroundings)

VIII. Conclusion

SUMMARY OF WORK ACCOMPLISHED

This project has collected a suite of geospatial images for use in validating or verifying algorithms that extract semantic content from overhead images of industrial plants, and it has demonstrated using those images to test the performance of algorithms for finding buildings in a scene. (An algorithm that detects, locates and labels specific human recognizable features in an image — buildings for instance — is an example of a semantic extraction algorithm.) The suite of benchmark images contains 190 aerial RGB GeoTIFF images of real facilities, 38 synthetic RGB GeoTIFF images of fictitious electrical power plants, and 16 lidar data sets. Each of the latter data sets corresponds to one of the real aerial images. The suite of imagery contains not only the images, it also provides ground truth information in the form of labeled objects in the scene. The vocabulary for labels consists of about a dozen generic industrial features (Table 2). The five sorts of industries include chemical processing plants, heat processing plants, mechanical processing plants, heavy manufacturing facilities, and semiconductor plants.

LESSONS LEARNED

1. Having many users use the Benchmark Suite to test more algorithms and assess more scenarios is essential to guide future efforts in creating a strong benchmark suite of images.
2. The detailed requirements on the imagery are not completely known. The question “what is good enough” arose time and again with respect to many aspects of creating the current Benchmark Imagery Suite. For example, what counts as a realistic synthetic image? This issue is somewhat subjective and can depend upon a specific facility, algorithm or sensor.
3. Stand-alone automated segmentation and annotation algorithms are not plentiful. This, we believe, is because analysts tend to use toolkits of many types of algorithm, “brewing their own” procedures as needs dictate.
4. Having a capability to render images quickly and to change key aspects of a scene easily such as time of day, components or layout is important but is not sufficient for creating synthetic images quickly.
 - Commercial models of industrial facilities or their components may not be available.
 - Available models may be overly limited in their appearances.
 - A large part of generating synthetic images of specific types of industrial plants is spent learning what features should be included in the scene and the details of how they should appear when rendered. This determination requires significant time spent poring over images of various sites and consulting with SMEs.

A PATH FORWARD

Our recommended path forward includes three tasks:

- Enlarge the contents of the suite.
- Improve the semantics behind the suite.
- Disseminate the suite.

First, enlarge the existing suite of images and ground truth. The present Benchmark Imagery Suite is but a small step, with essentially only one type of imaging modality (visible-near-infrared), CONUS images only, one image per industrial site, split among five industrial classes.

Therefore, increase the number of images and the types of facilities represented and the fraction of images that are exhaustively annotated. Some specific improvements would be:

- Add more imaging modalities, especially lidar with normalized digital elevation models (height above ground).
- Increase the number of facility images in the present industrial classes.
- Add images of OCONUS facilities.
- Increase the number of areal annotations to include all of the real-world images in the Benchmark Imagery Suite, not just 51 of them.
- Increase the number of industry classes represented in the suite.
- Expand the set of synthetic imagery, first by creating additional layouts of electrical power plants on all of the current base terrains as was done for only nuclear power plants on an arid background. Then add more types of facility and types of background.

Second, expand the underlying semantic foundations of the Benchmark Imagery Suite. The meaning or interpretation of an image in words is limited by the vocabulary available to describe it. Moreover, the process of assigning humanly meaningful descriptors to an image of a facility requires an extensive network of like and dislike images and knowledge that is not directly in the image — knowledge of manufacturing processes at a given type of facility, for example. Refining the definitions of the industrial classes, increasing the size of the lexicon, defining new industrial features like boundaries, and creating detailed, pertinent ontologies are practical ways to expand the semantic foundations.

Finally, make the Benchmark Imagery Suite available to as wide an audience of users as is practical. If possible, receive and incorporate their comments and suggestions to improve the suite. In the opinion of the authors disseminating the suite is the most important next step in the path forward. Although not a trivial task, posting it online in an appropriate manner with autonomous access to authorized users would help disseminate it. Doing so will almost certainly result in greater insights into its utility, and boost research into semantic algorithm design, testing, and inter-comparisons. The precedent set by the computer vision community (e.g., Cal Tech 256 [4], OIRDS [10], etc.) is quite clear in this regard. Interactions with users in relevant working groups and professional societies such as the Geospatial Working Group [43], the American Society for Photogrammetry and Remote Sensing [44] or the Open Geospatial Consortium [45] would provide useful insight. It is anticipated that increasing the access to the suite will cause more algorithms to be tested and additional test metrics to be used, which will improve understanding of both the algorithms being tested and the benchmark images.

IX. References

- [1] See, for example, Niles Ritter and Mike Ruth, “GeoTIFF Format Specification, GeoTIFF Revision 1.0,” <http://www.remotesensing.org/geotiff/spec/geotiffhome.html>.
- [2] NNSA Funding Opportunity Announcement, Funding Opportunity Number DE-FOA-0000568, CFDA Number 81.113, December 29, 2011.
- [3] Mizoguchi, R, "Part 1: Introduction to ontological engineering, NEW GENERATION COMPUTING, Volume: 21 Issue: 4 Pages: 365-384 Published: 2003
- [4]. L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories,” *Computer Vision and Image Understanding*, 106(1), pp. 59–70, 2004.
- [5] Field of Dreams, Phil Alden Robinson (screenplay), Universal Pictures, April 1989.
- [6] Smith, B and Ceusters, W, "Ontological realism: A methodology for coordinated evolution of scientific ontologies," *Applied Ontology* 5(3-4), pp. 139-188.
- [7] Thomas C. Chisnell and Gordon E. Cole, “‘Industrial Components’ — a Photointerpretation Key on Industry,” *Photogrammetric Engineering* 24, 590 – 602, March 1958.
- [8] The html references that were used to form the industrial facility gazetteer are:
<http://www.epa.gov/ttn/atw/area/facilities.html>
<http://www.epa.gov/enviro/html/fii/ez.html>
<http://www.energyjustice.net/map/>
<http://www.osha.gov/pls/imis/establishment.inspection>
- [9] J. Shufelt, "Performance evaluation and analysis of monocular building extraction from aerial imagery," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(4), pp. 311–326, April 1999.
- [10] F. Tanner et al., “Overhead imagery research data set - an annotated data library & tools to aid in the development of computer vision algorithms,” in Proceedings of the 2009 IEEE Applied Imagery Pattern Recognition Workshop (AIPRW), Oct 14-16, 2009, pp. 1–8.
- [11] http://en.wikipedia.org/wiki/Lidar#History_and_etymology_of_lidar.2FLIDAR
- [12] “LAS SPECIFICATION VERSION 1.4 – R13,” The American Society for Photogrammetry & Remote Sensing, 5410 Grosvenor Lane, Suite 210, Bethesda, Maryland 20814-2160, 15 July 2013.
- [13] Peterson, Jeffery. (2010), “Predicting Pollutant Dispersion and Exposure in Mixed Outdoor-Indoor-Vehicle Environments” US Army Dugway Proving Ground. Chemical and Biological Defense Science, and Technology Conference, Orlando Florida. [Predicting Pollutant Dispersion and Exposure in Mixed Outdoor-Indoor-Vehicle Environments](#)

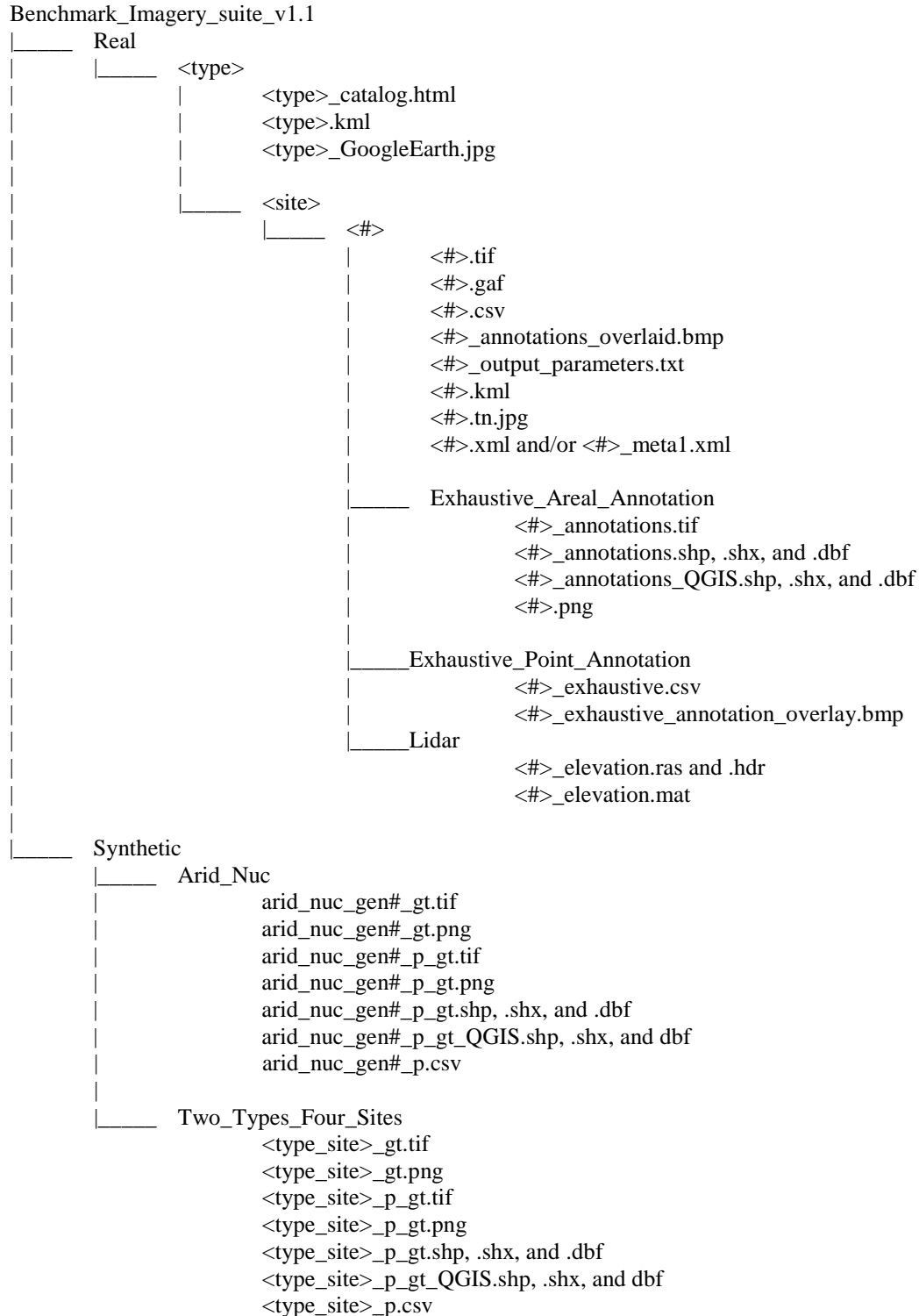
- [14] Gaurang, Dave. (2010), “Joint Expeditionary Collective Protection System Performance Model (JECSP SPM) Overview” Naval Surface Warfare Center, Dahlgren. Chemical and Biological Defense Science and Technology Conference, Orlando Florida. [Joint Expeditionary Collective Protection System Performance Model \(JECSP SPM\) Overview](#)
- [15] Chen, Tung-Chieh, and Yao-Wen Chang. (2006), “Modern Floorplanning Based on B*-Tree and Fast Simulated Annealing,”. *IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems*, Vol. 25, No. 4.
- [16] TurboSquid, 935 Gravier Street, Suite 1600, New Orleans, LA, 70112, <http://www.turbosquid.com/>.
- [17] Trimble 3D Warehouse. (Formerly Google 3D Warehouse). <http://sketchup.google.com/3dwarehouse/>
- [18] 3DSMax software application. <http://www.autodesk.com/products/autodesk-3ds-max/overview>
- [19] OSG Exporter Plugin from Delta3D. <http://www.delta3d.org/forum/viewtopic.php?showtopic=17151>
- [20] <http://www.google.com/earth/index.html>.
- [21] Global Mapper software application. <http://www.bluemarblegeo.com/products/global-mapper.php>.
- [22] Trimble Sketchup software application. <http://www.sketchup.com/>
- [23] GNU Image Manipulation Program, <http://www.gimp.org/>
- [24] PNG format. https://en.wikipedia.org/wiki/Portable_Network_Graphics.
- [25] COLLADA format. <https://collada.org/>
- [26] OSGConv Tool from Open Scene Graph. <http://trac.openscenegraph.org/projects/osg/wiki/Support/UserGuides/osgconv>
- [27] “The USGS National Map Viewer and Download Platform,” <http://nationalmap.gov/viewer.html>. Last date accessed May 29, 2013.
- [28] Goforth, J. W., L. R. Gaines, W. T. White, and P. A. Pope, “Selecting Real Imagery for Generating Composite Images,” LLNL-TR-631254, March 27, 2013.
- [29] L. R. Gaines, "Automatic Facility Layout Generation for Nuclear Power Plants," Poster presented at the 2013 LANL Student Symposium, July 26th – 27th, 2013 held in Los Alamos, NM, LA-UR-13-25407.

- [30] Robin S. Liggett, “Automated facilities layout: past, present and future,” *Automation in Construction* 9, 2000, 197–215.
- [31] Authors: Saurabh Adya, Hayward H. Chan, Igor Markov, <http://vlsicad.eecs.umich.edu/BK/parquet> .
- [32] Zhou, Jie. “Modern Floorplanning Based on B*-Tree and Fast Simulated Annealing,” VLSI Physical Design: From Graph Partitioning to Timing Closure, Chapter 3 in Lectures on Chip Planning, University of Michigan, Fall 2011.
- [33] Khan M. Iftexharuddin, “Overhead Image Segmentation Final Report,” September 20, 2013, Department of Electrical and Computer Engineering, Old Dominion University, submitted in partial fulfillment of Subcontract #B604457 with Lawrence Livermore National Security, LLC.
- [34] National Cancer Institute, The Cancer Genome Atlas Data Portal, <https://tcga-data.nci.nih.gov/tcga/tcgaDataType.jsp>.
- [35] Liping Di, Peng Yue, Ziheng Sun, “Ontology-supported Complex Feature Discovery in a Web Service Environment,” IEEE International Geosensing and Remote Sensing Symposium (IGARSS 2012), held 22–27 July 2012, Munich, Germany, Paper WE3.10.2.
- [36] R. S. Roberts, P. A. Pope, M. Jiang, T. Trucano, C. Aragon, K. Ni, T. Wei, L. Chilton, A. Bakel, “GSV Annotated Bibliography,” LLNL-TR-487191, June 14, 2011.
- [37] Proposal #LL11-FY11-191-PD06 Project Lifecycle Plan, “Benchmark Imagery for Assessing Geospatial Semantic Content Extraction Algorithms (Updated),” Prepared for the U.S. Department of Energy (DOE) National Nuclear Security Administration, Office of Nonproliferation and Verification R&D (NA-22), January 2012.
- [38] L. Prasad and S. Swaminarayan, “Hierarchical Image Segmentation by Polygon Grouping”, *Proc. of IEEE CVPR Workshop on Perceptual Grouping in Computer Vision*, Anchorage, AK, June 2008.
- [39] Wikipedia, “Binary Classification,” http://en.wikipedia.org/wiki/Binary_classification, last accessed September 29, 2013.
- [40] Wikipedia, “Confusion Matrix,” http://en.wikipedia.org/wiki/Confusion_matrix, last accessed September 29, 2013.
- [41] Wikipedia, “Terminology and derivations from a confusion matrix,” http://en.wikipedia.org/wiki/Sensitivity_and_specificity#See_also , last accessed Sept. 29, 2013.
- [42] P. Pope, “Benchmark Imagery Project: Photointerpretation Key,” April 16, 2013, LAUR-13-22729.
- [43] The Geospatial Intelligence Standards Working Group, <http://www.gwg.nga.mil/index.php>.
- [44] ASPRS The Imaging and Geospatial Information Society, <http://www.asprs.org/>.
- [45] The Open Geospatial Consortium (OGC), <http://www.opengeospatial.org/>.

X. Appendix A — Organization of the Benchmark Imagery Suite v1.1a

Organization of the Data

The Benchmark Imagery Suite v1.1a files are organized in the following directory hierarchy:



Real

There are five directories, one for each of five different industry types (<type>): Chemical Processing (CP), Heat Processing (HP), Heavy Manufacturing (HM), Mechanical Processing (MP), and Semiconductor Industry (SI). Under each of these directories, there are the following files:

<type>_catalog.html - Contains catalog information in a tabular layout, stored in HTML format (e.g., Chemical_Processing_catalog.html).

<type>.kml- Contains the center point locations of all sites of a specific industrial type, stored in KML format (e.g., Chemical_Processing.kml).

<type>_GoogleEarth.jpg - a screen capture of the site locations as rendered by GoogleEarth(c) (Figure 1).

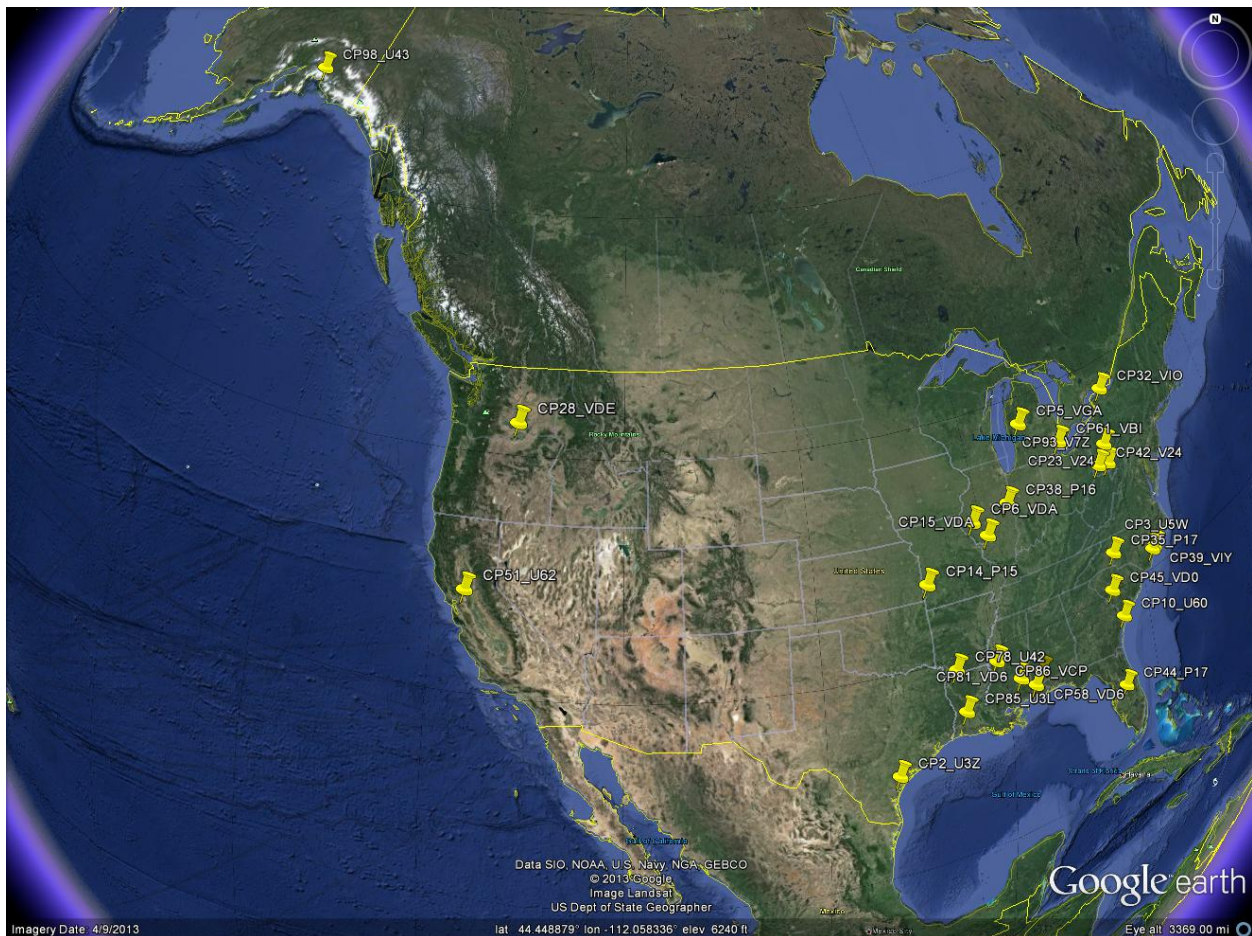


Figure 1. Screen capture of the site locations for the Chemical Processing (CP) industry type as rendered using GoogleEarth(c).

Under each of the five industry type directories, there are subdirectories whose names are the alphanumeric site names (<site>; e.g., “CP2_U3Z”).

Under each of these alphanumeric site name subdirectories, there is a single subdirectory whose name is the image number (<#>; e.g., “44608466”).

Under each image number directory, there are the following types of files:

<#>.tif - An USGS orthoimage in GeoTIFF format (with georeferencing).

<#>.gaf - Point annotations in GAF format (i.e., an IDL structure in IDL SAVE file format).

<#>.csv - Point annotations in CSV format.

<#>_annotations_overlaid.bmp - The point annotations overlaid onto the orthoimage (Figure 2).



Figure 2. The point annotations (“44608466.csv”) overlaid on the orthoimage (“44608466.tif”) for Chemical Processing site CP2_U3Z, image number 44608466, as contained in the file “44608466_annotations_overlaid.bmp.”

<#>_output_parameters.txt - A metadata file containing GSD and geographic extents information.

<#>.kml - The geographic extents of the orthoimage, expressed as a polygon, in KML format.

<#>.tn.jpg - Thumbnail version of the orthoimage, referenced by the catalog HTML file (<type>_catalog.html).

<#>.xml and/or <#>_meta1.xml - Metadata for the orthoimage as provided by the USGS.

Under some of the image number directories, there are up to three other directories; "Exhaustive_Point_Annotation", "Exhaustive_Areal_Annotation", and "Lidar."

Under the "Exhaustive_Point_Annotation" directory, there are the following types of files:

<#>_exhaustive_point.gaf - Point annotations in GAF format (i.e., an IDL structure in IDL SAVE file format).

<#>_exhaustive_point.csv - Point annotations in CSV format.

Under the "Exhaustive_Areal_Annotation" directory, there are the following types of files:

<#>_annotations.tif - The exhaustive areal annotations in the form of a raster mask, in TIFF format (without georeferencing).

<#>_annotations.shp, .shx, and .dbf - The exhaustive areal annotations raster mask, converted to vector format, suitable for overlaying on the image stored in the TIFF or PNG file by using, for example, ENVI.

<#>_annotations_QGIS.shp, .shx, and .dbf - The exhaustive areal annotations raster mask converted to vector format, suitable for overlaying on the image stored in the TIFF or PNG file by using, for example, QGIS.

<#>.png - The USGS orthoimage in PNG format (without georeferencing).

Under the "Lidar" directory, there are the following types of files:

<#>_elevation.ras and .hdr - The point cloud contained in a LAS file, converted to a raster of elevations above mean sea level stored in ENVI standard raster format (i.e., row concatenation; lexicographical ordering), and its associated ENVI header file stored in ASCII format.

<#>_elevation.mat - Same as the <#>_elevation.ras file, but in Matlab format.

Synthetic

Under the Synthetic directory, there are two subdirectories, Arid_Nuc and Two_Types_Four_Sites.

Under the Arid_Nuc directory, there are the following types of files:

arid_nuc_gen#_gt.tif - The synthetic image in GeoTIFF format (with georeferencing).

arid_nuc_gen#_gt.png - The synthetic image in PNG format (without georeferencing).

arid_nuc_gen#_p_gt.tif - The posterization image associated with the arid_nuc_gen#.tif image. This is the synthetic image equivalent of the exhaustive areal annotations mask created for the real imagery.

arid_nuc_gen#_p_gt.png - The posterization image in PNG format (without georeferencing).

arid_nuc_gen#_p_gt.shp, .shx, and .dbf - The posterization image converted to vector format, suitable for overlaying on the image stored in the TIF or PNG file by using, for example, ENVI.

arid_nuc_gen#_p_gt_QGIS.shp, .shx, and dbf - The posterization image converted to vector format, suitable for overlaying on the image stored in the TIF or PNG file by using, for example, QGIS.

arid_nuc_gen#_p.csv - The color triplet-to-object “key” for the posterization image stored in CSV format.

Note that the generation number (gen#) runs from gen1 to gen30.

Under the Two_Types_Four_Sites directory, there are the following types of files:

<type_site>_gt.tif - The synthetic image in GeoTIFF format (with georeferencing).

<type_site>_gt.png - The synthetic image in PNG format (without georeferencing).

<type_site>_p_gt.tif - The posterization image associated with the <type_site>_gen#.tif image. This is the synthetic image equivalent of the exhaustive areal annotations mask created for the real imagery.

<type_site>_p_gt.png - The posterization image in PNG format (without georeferencing).

<type_site>_p_gt.shp, .shx, and .dbf - The posterization image converted to vector format, suitable for overlaying on the image stored in the TIF or PNG file by using, for example, ENVI.

<type_site>_p_gt_QGIS.shp, .shx, and dbf - The posterization image converted to vector format, suitable for overlaying on the image stored in the TIF or PNG file by using, for example, QGIS.

<type_site>_p.csv - The color triplet-to-object “key” for the posterization image stored in CSV format.

Note that <type_site> takes on all eight combinations of the two types, “nuc” (i.e., nuclear power plant) and “coal” (i.e., coal-fired power plant), and the four sites, “arid,” “denver,” “forest,” and “ocean.”

XI. Appendix B — Photointerpretation Key

Paul A. Pope
15 April 2013

GENERAL GUIDANCE

The [controlled vocabulary](#) is biased toward helping an automated algorithm discern between objects. For example, a “Cooling Tower” is a type of “Cooling Unit;” however, “Cooling Towers” are (generally) massive in size (and therefore cast a long, large shadow, and the structure itself exhibits large perspective distortions; characteristics which an automated algorithm might be able to leverage), whereas a “Cooling Unit” is sized somewhere between an “A/C Unit” or “HVAC unit” and a “Cooling Tower.” Another example is “Pond” versus “Tank.” For our purposes, a “Tank” is a fully enclosed structure, whereas a “Pond” is a container of any shape (e.g., natural, round, rectangular, square, etc.), installed either above or below ground, which uses either soil or concrete or metal to effect the containment, and which has an open view to the sky, such that the liquid within often appears dark in tone. Additionally, a “Tank” might contain solid, liquid, or gas, whereas a “Pond” will only contain liquid (if present). Note that in this context a “Fish Tank,” like those found at a fishery, would instead be labeled as “Pond” rather than “Tank.” This line of reasoning is taken because the generally dark toned, exposed liquid is probably a feature that an automated algorithm could be trained to discern, and therefore this differentiation in terms should help in algorithm development.

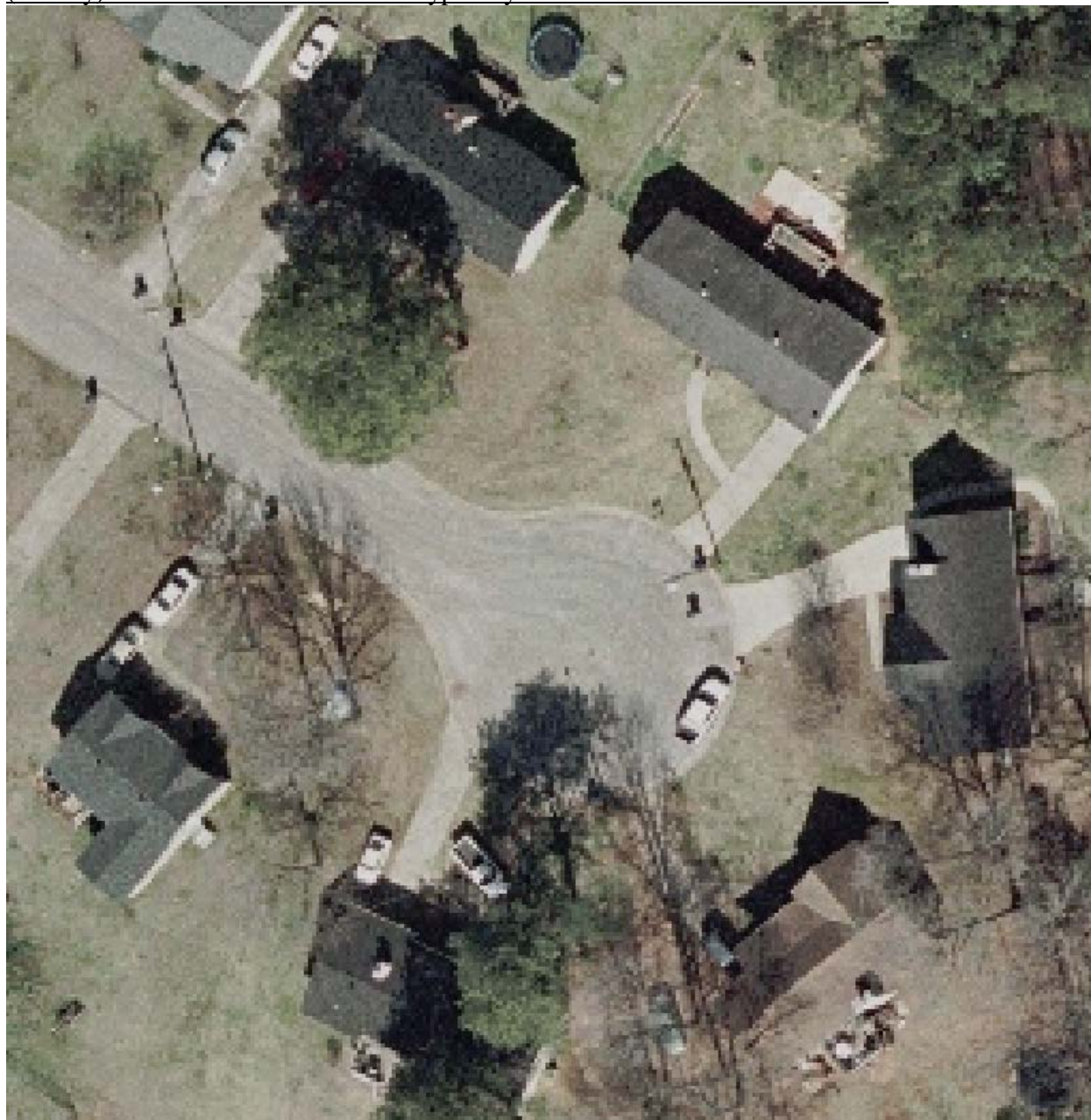
If the mouth of a stack is occluded due to smoke or steam, then do not label that particular object, even though it might be particularly salient. Similarly for “Raillines;” please label the track and its associated bed and not the “Train” atop it, because even though the presence of a “Train” can be used to infer the presence of a “Rail line” underneath it, the “Train” occludes the “Rail line.” Individual objects and not groups of objects (e.g., “Tank Farm”) should be labeled. Current object detection algorithms already find it challenging to consistently find even single objects, much less groups of them. Adhering to these guidelines will lend consistency to the labeling and therefore should aid in developing an automated algorithm.

If a clear semantic distinction cannot be made and/or the object cannot be seen clearly (i.e., it is occluded by other objects, or cloud/smoke/steam, etc.), then please do not label an object. This will help to ensure labeling consistency across our team. Bottom Line: “When in doubt, leave it out.”

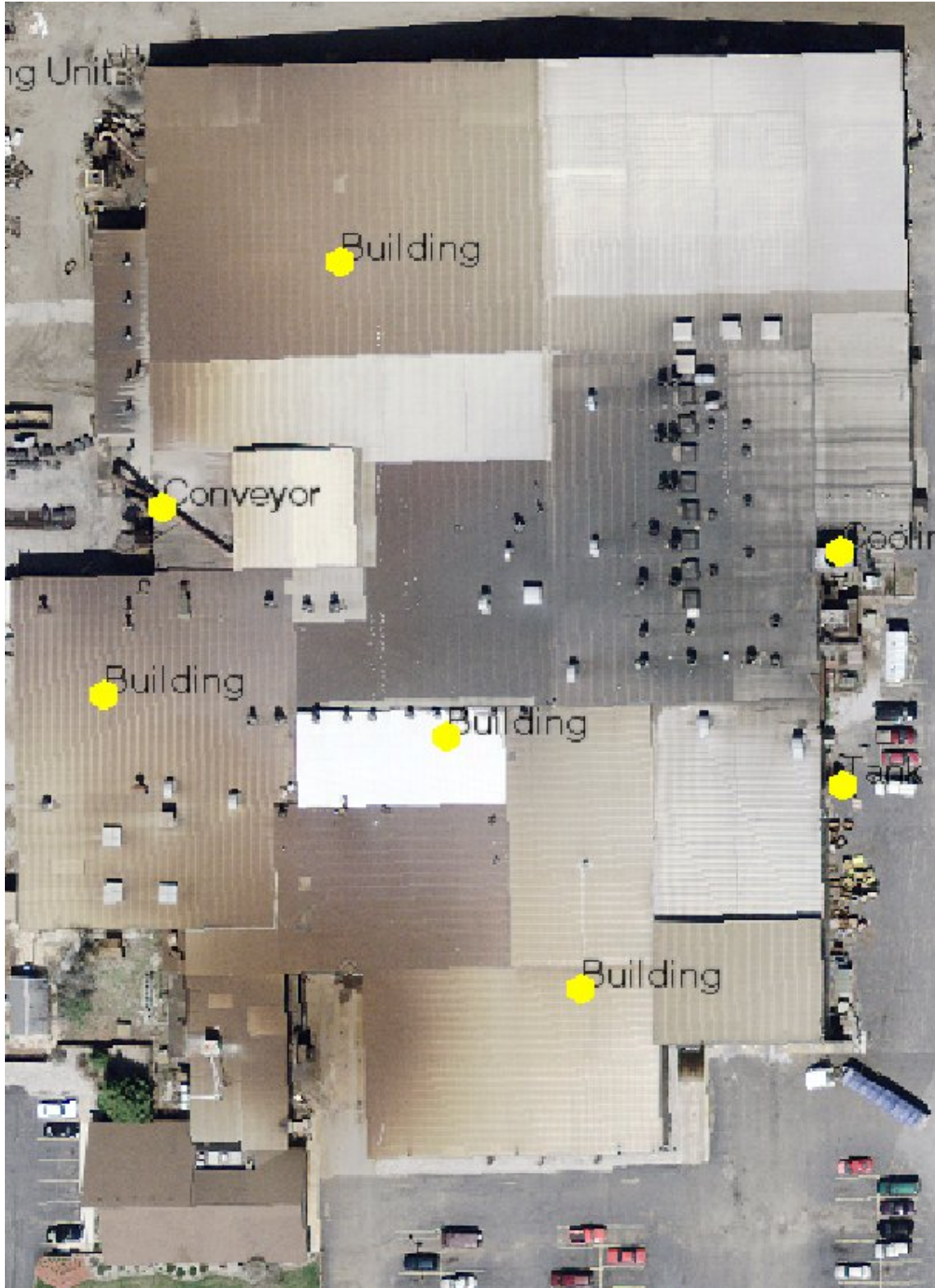
“Building” - A structure consisting of, at a minimum, a foundation, a roof, and supports. This “catch-all” label is to be used for buildings which are not “House” nor “Industrial Building,” and for buildings whose use cannot be readily determined. A detached “Garage” or “Church” or recreational facility or fast-food restaurant or strip-mall building or other medium-sized commercial building are examples of a “Building.”



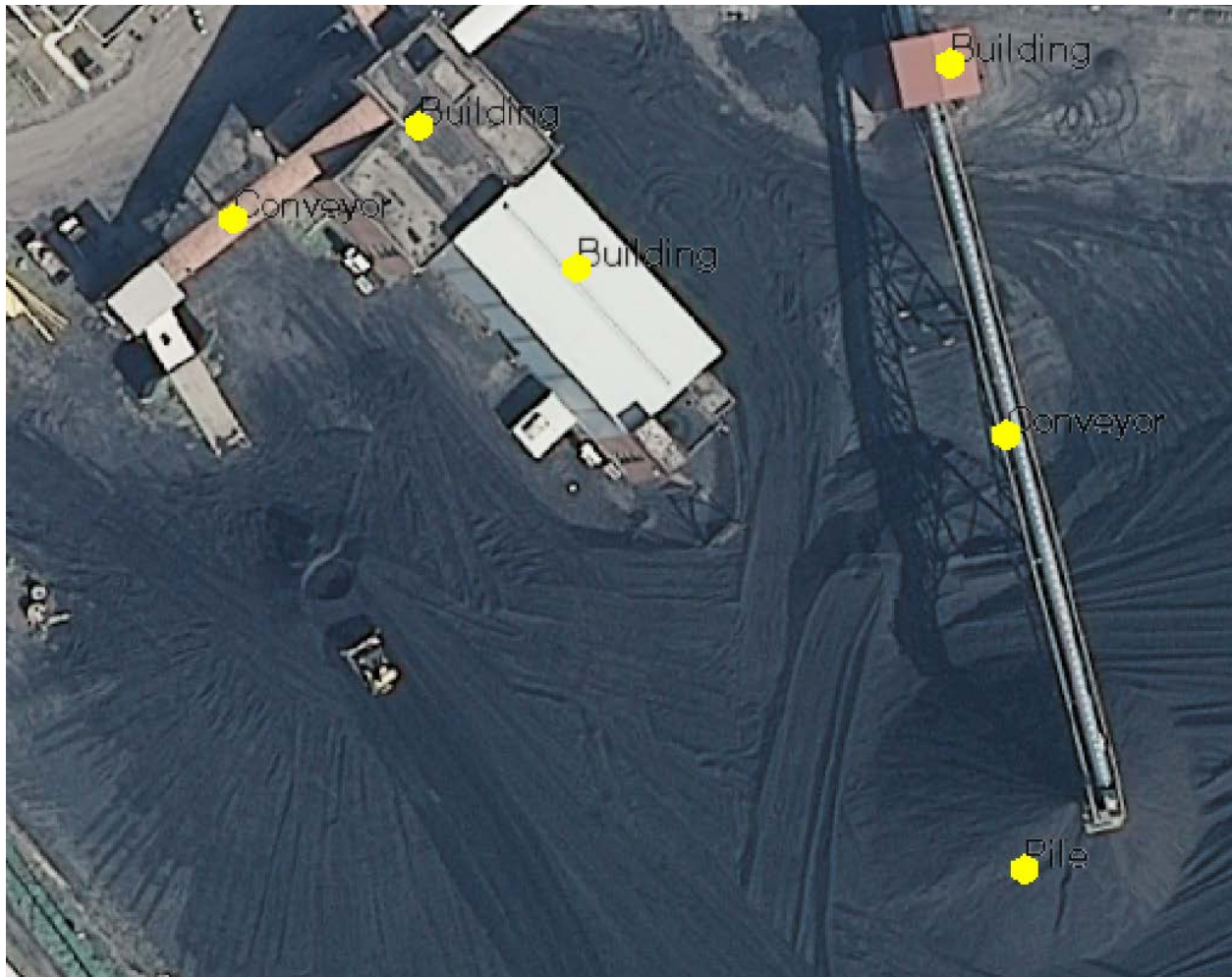
”House” - A structure consisting of, at a minimum, a foundation, a roof, and supports, which is (usually) small in areal extent and is typically located within a residential area.



“Industrial Building” - A structure consisting of, at a minimum, a foundation, a roof, and supports, which is (usually) large in areal extent and located within the bounds (e.g. fence-line) of an industrial facility.



“Conveyor” - A structure (e.g., belt, duct, pipe, etc.) used to move material (solid, liquid, or gas; e.g. a conveyor belt, gas/liquid pipeline, or steam pipe), electricity (e.g., conduit), or people (e.g., skybridge).



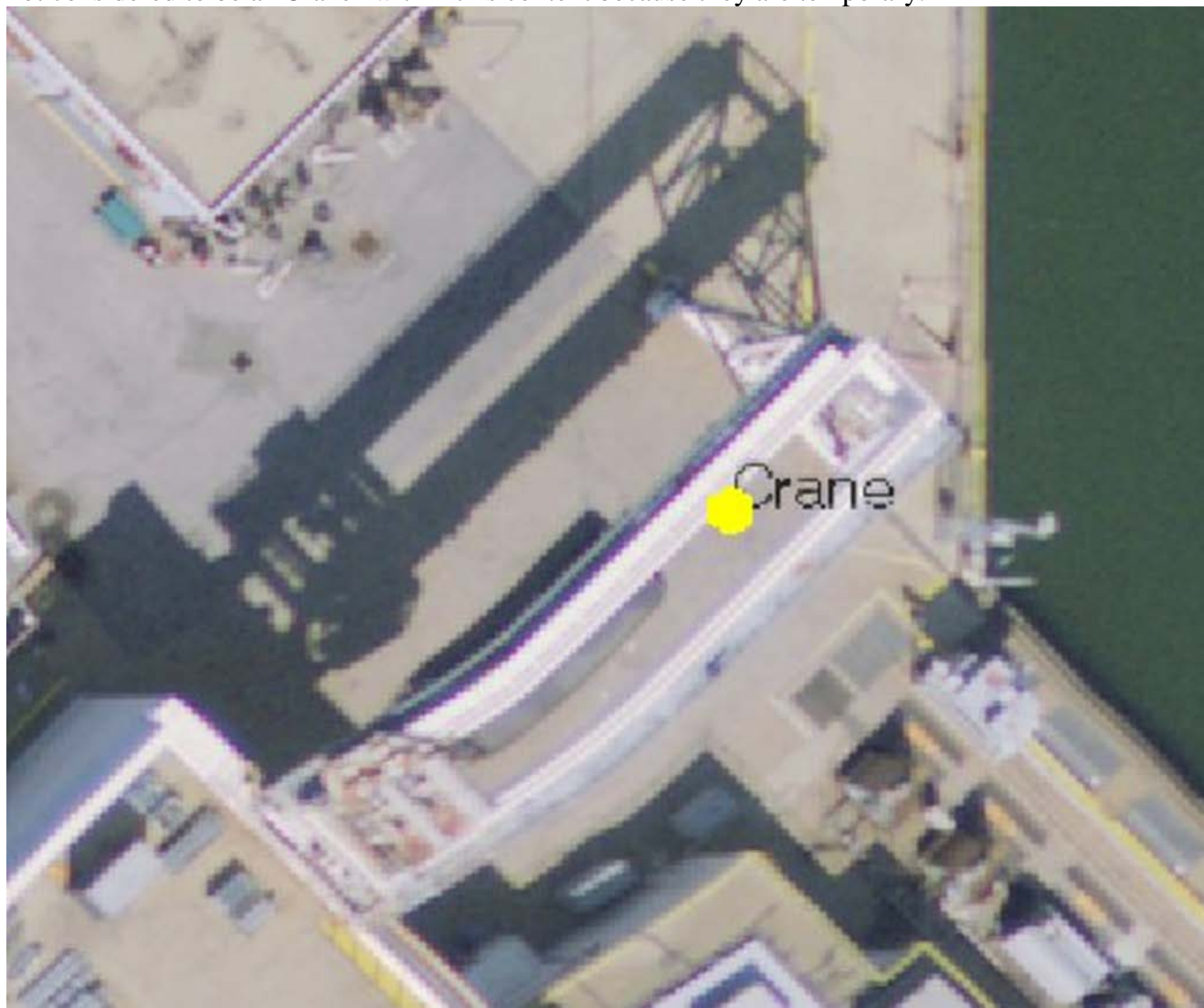
“Cooling Tower”- A type of “Cooling Unit” which is typically a very tall, hyperboloid-shaped structure. “Cooling Towers” are most often associated with nuclear power plants, although they can also be found at other facilities.



“Cooling Unit”- A piece of equipment used to dissipate or remove heat from other equipment or a building by dumping it into the surrounding environment through the use of an intermediate fluid (e.g., water, refrigerant). This includes “Chillers” and “Cooling Towers.” “Cooling Units” are smaller in stature than “Cooling Towers” and their capabilities are somewhat self-contained (akin to an “A/C Unit” or “HVAC Unit” but are larger in size).



“Crane” - Permanently mounted equipment affixed to a facility which is used to lift, and possibly move, heavy objects. Specifically, “Overhead Crane” like those used in the Heavy Manufacturing industry. Mobile equipment and vehicular-mounted cranes such as “Truck-mounted Crane,” “Backhoe,” etc. are not considered to be “Cranes” within this context. Although they are often affixed on top of or adjacent to a facility, a “Construction Crane” is also not considered to be a “Crane” within this context because they are temporary.



“Electrical Substation” - A collection of structures used for the transmission, transformation, distribution or switching of electric power.



“Kiln” - Equipment (e.g. furnace or oven) for drying, baking, or burning material. A “Cement Kiln” and a “Rotary Kiln” are types of “Kilns” within this context.

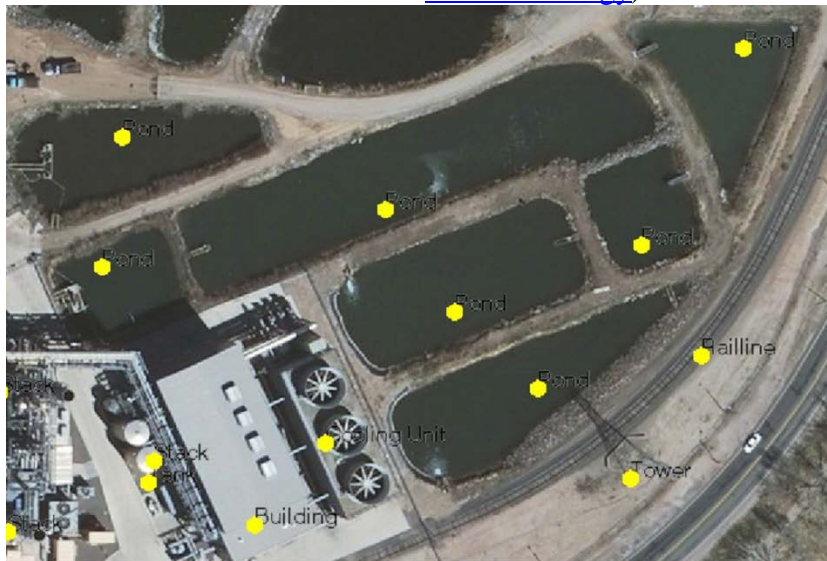


A horizontal, rotary kiln can be seen at the center of this image of the cement plant south of Cape Girardeau, Missouri as referenced in “Land Use/Land Cover and Environmental Photointerpretation Keys,” USGS Bulletin 1600, pp. 57-58.

“Pile” - A purposefully-placed heap of solids with some angle of repose. Note that “Pile” can refer to fine materials like sand, gravel, or coal that has some angle of repose, or it can refer to tires, lengths of pipe, junk (as in “Junk Pile”), or scrap metal. Note that items which are stacked (e.g., lengths of pipe or I-beams) are a type of “Pile,” albeit they are generally more orderly in their arrangement.



“Pond” - A container of any shape (e.g., natural, round, rectangular, square, etc.), installed either above or below ground, which uses either soil or concrete or metal to effect the containment, and which has an open view to the sky, such that the liquid within often appears dark in tone (since the liquid may contain significant solids). Exceptions to this generally dark-toned appearance may be water-aeration ponds, treated water storage reservoirs, and natural ponds with high algal content. Note that “Holding Pond,” “Settling Pond,” and “Natural Pond” are types of “Pond” within this context, as well as “Skimming Tank,” and “Fish Tank.” A “Natural Pond” is a dammed (either naturally or artificially) body of water (smaller than a lake) such that the flow of water is either constrained or completely restricted. “Ponding” (i.e., where the flow of liquid is momentarily restricted) is not considered a “Pond” within this context. A “Pond” is an [endurant](#) (persistent), whereas “Ponding” is a [perdurant](#) (fleeting) (Note: click on these words for their definitions within the context of [formal ontology](#)).



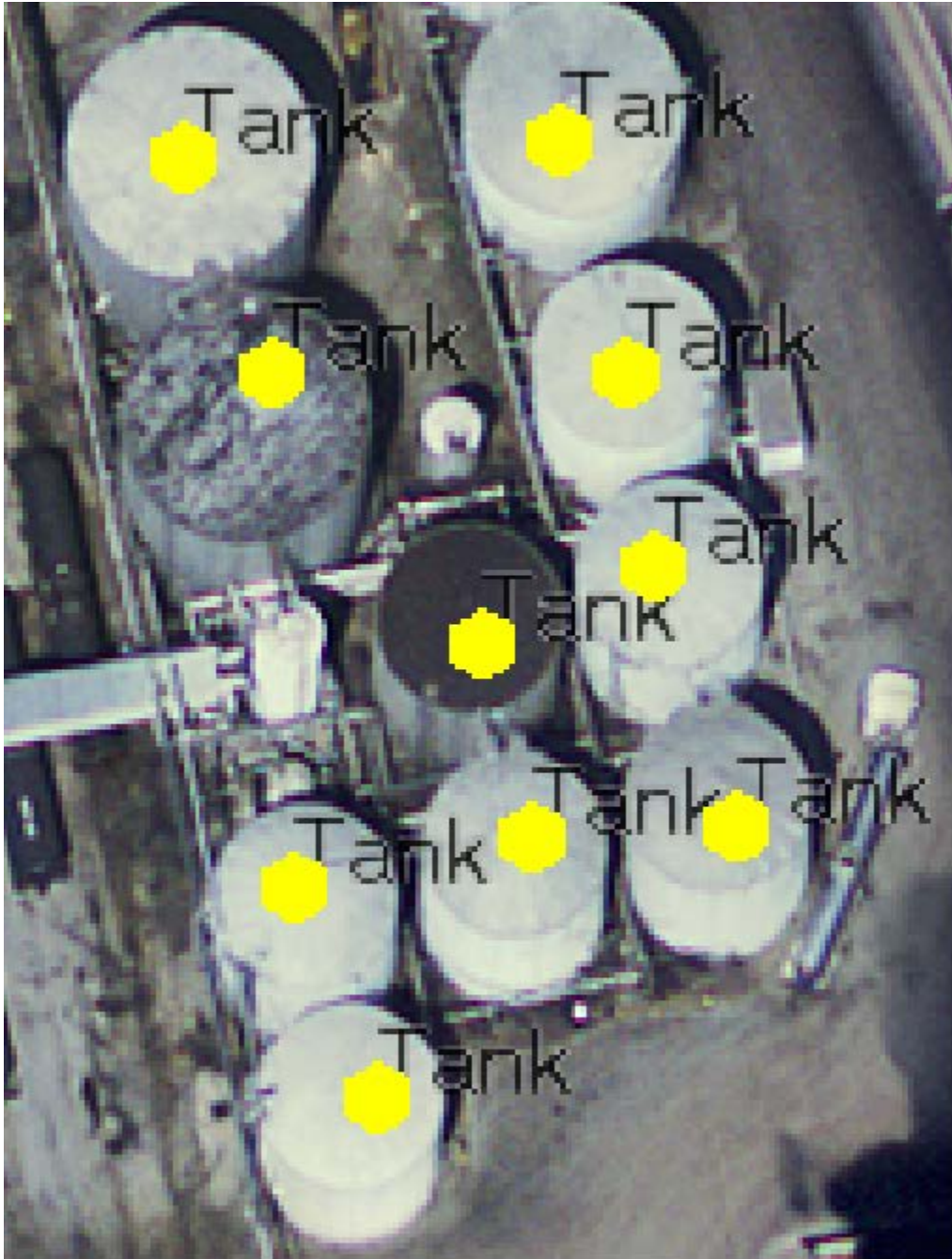
”Rail line” - A linear structure consisting of a foundation (e.g. compacted material) upon which rests horizontally oriented bars of metal, to facilitate the movement of locomotives, railcars, etc. Note that even though a “Rail line” can be inferred by the presence of a “Train,” only the track and its associated bed should be labeled. If a “Train” obscures a “Rail line,” then the “Train” should not be labeled as “Rail line.”



”Stack” - A vertical structure used to convey exhaust; usually relatively tall in height compared to surrounding structures. Note that an orderly “Pile” of material (e.g., lengths of pipe or tires) is not considered a “Stack” (a homonym) within this context.



“Tank” - A non-mobile (i.e., fixed), enclosed structure (i.e., with a lid and walls), of wide variety in shape (e.g. cylindrical, spheroidal, etc.), and oriented either horizontally or vertically, for holding solid, liquid, or gas. Note that a “Tank Car” (e.g., as part of a “Train”) is not considered a “Tank” within this context.



“Tower” - A vertical piece of equipment (e.g. watch tower, fractionating tower, cracking unit, etc.), usually relatively tall in height compared to surrounding structures. This term includes electrical power transmission towers, distillation towers (columns) in chemical plants, and security towers. Note however that a “Cooling Tower” is not a type of “Tower” within this context. A “Cooling Tower” is a type of “Cooling Unit.”



“Water Tower” - A tank, elevated via placement on top of a tower. Note that its use for the storage of water is assumed.

