Denied-Area Modeling using the LEGO[1] System

[1]Large-scale Environment-modeling with Geometric Optimization

Nick Maxwell nmaxwell@ara.com

John Papadakis jpapadakis@ara.com

Craig Stutts cstutts@ara.com

August 2021

NATIONAL SECURITY

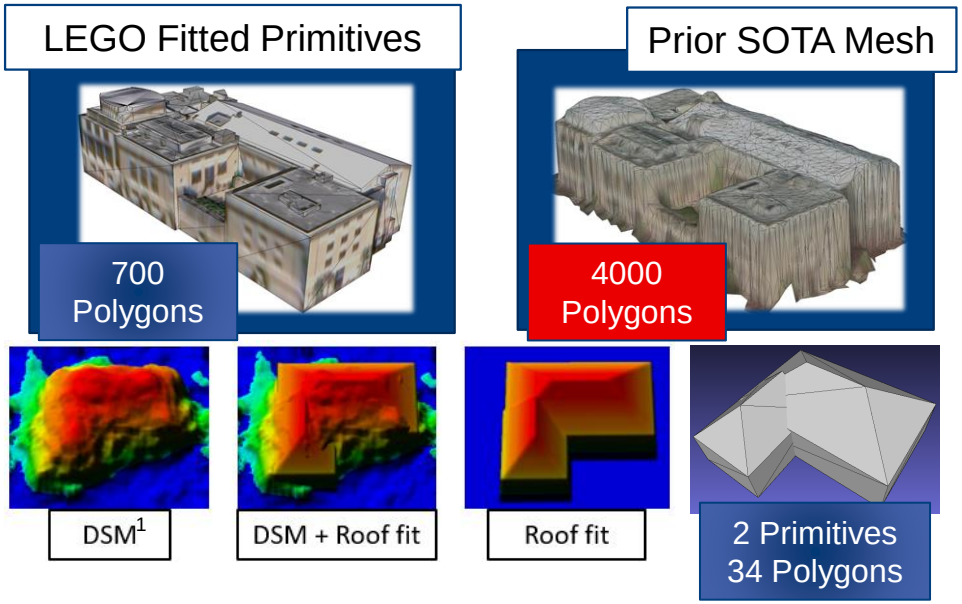
INFRASTRUCTURE

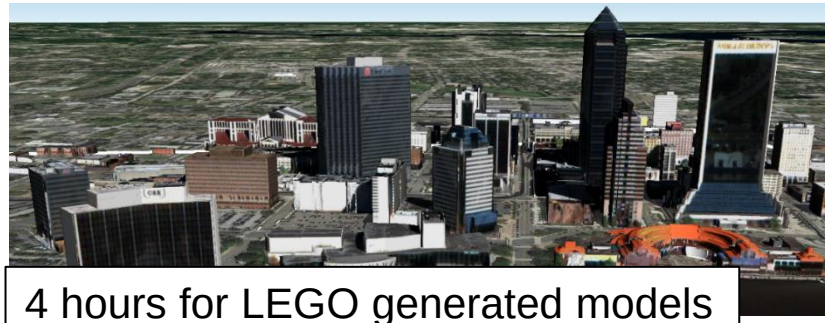
ENERGY & ENVIRONMENT

HEALTH SOLUTIONS

Origins: IARPA CORE3D

- Creation of Operationally Realistic 3-D Environments, Sep 2017 – Sep 2020
- Fully automated system to generate 3D CAD-like models of urban scenes from satellite imagery, DTED, and OSM road vectors
- LEGO started as ARA's solution to CORE3D
- Academic Team: ETH Zurich, Purdue, Duke, Univ. of TN





4 hours for LEGO generated models

5 staff days for manual models

ARA

LEGO Runs on a Variety of Urban Areas …

- San Diego, CA
- Los Angeles, CA
- Jacksonville, FL
- Omaha, Nebraska
- San Fernando, Argentina
- Wright-Patterson Air Force Base
- Kabul, Afghanistan
- Homs, Syria
- Phnom Penh, Cambodia
- Baghdad, Iraq
- More…

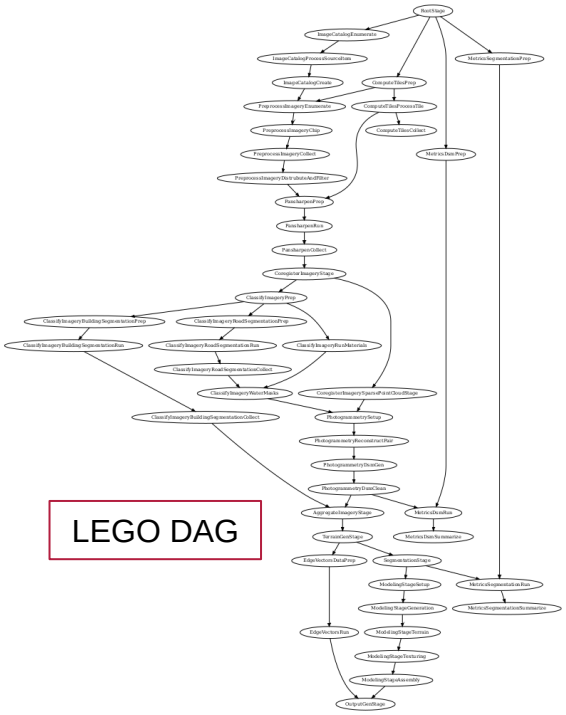
Several ongoing programs are testing LEGO on many new disparate locations worldwide

USC, Los Angeles, CA

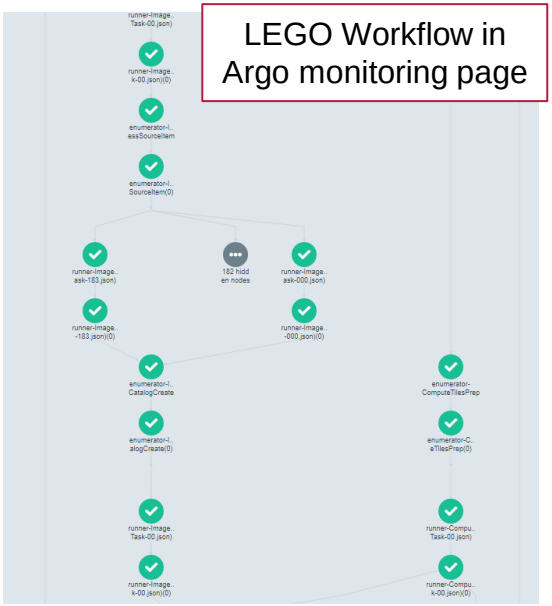
Homs, Syria

LEGO Architecture

- Mostly Python with C++ extensions
- Leverage many open-source third-party projects
- Runs on a laptop, workstation, cluster or AWS
- Scalable distributed processing on Kubernetes clusters
- Argo Workflows as workflow manager
  - Kubernetes native
    - Inherit resilience and robustness from kubernetes
    - autoscaling
  - https://argoproj.github.io/
  - Main drawback to Argo is the maintenance of cumbersome YAML workflow file
  - However, LEGO **auto-generates** this YAML



LEGO DAG



LEGO Workflow in Argo monitoring page

LEGO Runtimes

- 3-7 Hours / km^2 **on a laptop**
- Excellent scalability expected in AWS
  - TBD, this is under active development
  - Many processing steps area easily parallelized
  - LEGO can easily leverage Kubernetes autoscaling with Argo Workflows
- Runtime depends on:
  - (image processing) quantity of imagery available
  - (modeling, texturing) density and complexity of buildings in scene
- Note: LEGO currently performs buffered naive tiling:
  - Large buildings that straddle tile boundaries may be cut off
  - Needs further R&D

Name	AOI	Image Count	Area (km^2)	Rate hr/km^2	Total (hr)	Reconstruction	Modeling	Segmentation
Jax200K	Jacksonville, FL	34	0.3	6.7	2.2	0.9	0.5	0.2
Jax400K	Jacksonville, FL	34	0.4	5.7	2.4	1.0	0.5	0.2
Jax1M	Jacksonville, FL	34	1.0	6.7	6.4	1.9	2.7	0.4
Khomeini2M	khomeini space center, Iran	28	2.0	3.0	6.0	2.0	0.5	1.0

Runtimes on a laptop (8 cores, 128GB RAM, 16GB GPU)

LEGO at a High Level

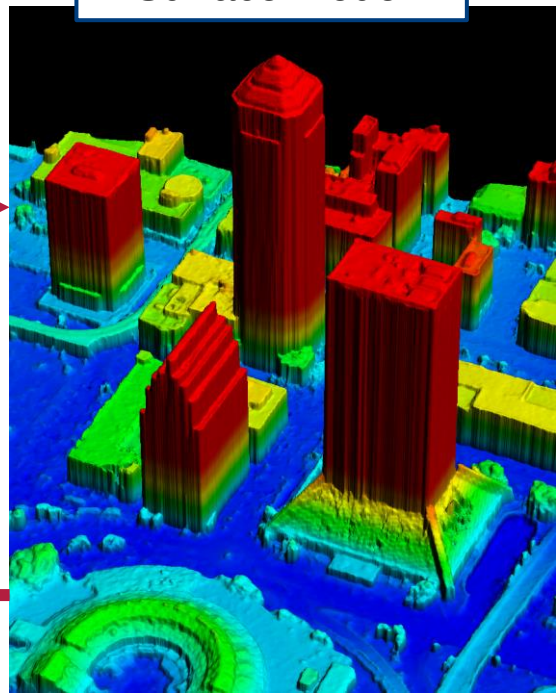
- Inputs:
  - Region of interest (e.g., KML polygons)
  - Commercial or government high-resolution Level 1B satellite imagery
  - DTED (usually global 1-arc-sec SRTM DEM)
  - OSM road vectors (optional)
- Outputs:
  - 3D CAD models, textured 3D meshes, and 3D wireframe models.
    - Models of buildings and elevated roads
  - Orthographic products: 0.3-meter DSM and DTM along with PAN and Pan-sharpened RGB, materials, functional classifications aggregate images
- Fully automatic
- No dependence on aerial data or other previous collections

LEGO at a High Level

Stack of high-res satellite imagery

- Radiometric Correction
- Pan-sharpening
- Alignment
- Segmentation
- Photogrammetry
- DSM Generation

Surface model



Textured 3D CAD models

- Material Classification via fusion
- Terrain Segmentation, Fitting
- True Orthorectification
- 3D Modeling, Primitive Fitting
- Projective Texturing

LEGO Pipeline

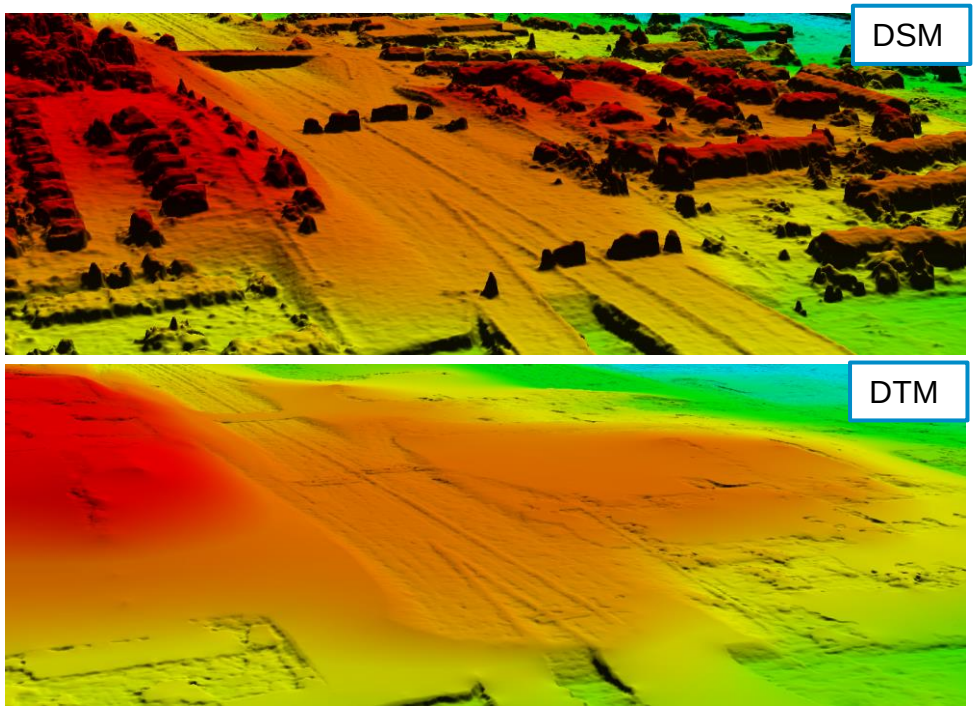
Multiple satellite images

- **Tiling** — Setup tiling to run LEGO on AOI
- **Pre-process Imagery** — Converts imagery to top-of-atmosphere reflectance, pan-sharpens imagery
- **Co-register Imagery** — Aligns imagery, correcting for errors in satellite pose
- **Image Classification** — Pixel-level neural network classification of materials and buildings
- **DSM Generation** — Generates stereo point clouds, fuses point clouds and refines data to create DSM[1]
- **Image Aggregation** — Orthorectifies then aggregates per-view imagery and classifications
- **Terrain Generation** — Create high resolution bare earth DTM[2] using DSM and classifications
- **Segment Features** — Fuse all data into final building and bridge estimates, segment buildings
- **Model Generation** — Fits geon models using pipeline data products
- **Output Creation** — Copies data to output folder

[1]DSM – Digital Surface Model
[2]DTM – Digital Terrain Model

Terrain Segmentation and Fitting

- Given a DSM, estimate bare-earth terrain, or DTM
- Not a well-defined problem:
  - What is the terrain underneath a building?
  - Terrain is not directly observed
- Plays important role in:
  - Estimate of building height
  - Anchoring point for models
  - Prior for other segmentation models
- Approach:
  - MRF with semantic segmentation prior
  - Iterate with surface fitting



Example Material Classification Result

Output Categories

- No Data
- Vegetation
- Soil
- Water
- Shadow
- Concrete
- Asphalt
- White Polymer
- Black Polymer
- Glass
- Solar Panel
- Metal
- Red Ceramic
- Red Paint
- Blue Paint
- Green Paint
- Gray Paint
- Brown Tile
- Clouds

Single-view result: subject to noise, view geometry, and atmospheric effects

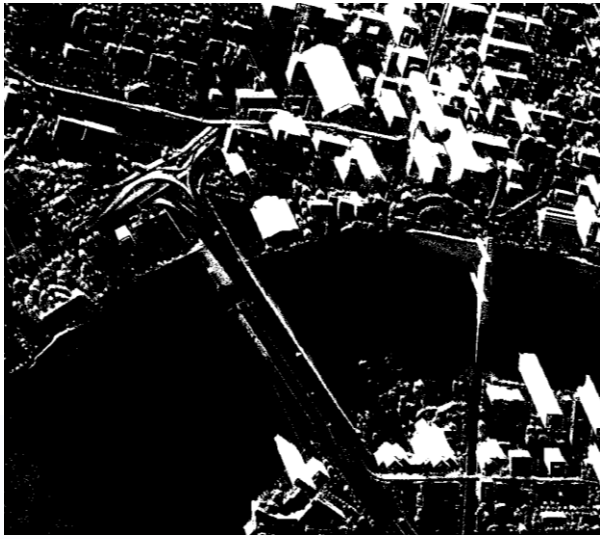
Multi-View Image Aggregation

- Classification performance improves when multiple results are aggregated
- Transient scene content is removed (e.g., cars, construction, clouds, shadows, etc.)
- The most commonly occurring classification is selected for each pixel (mode operation)
- Occluded regions are filled
- Shadows are identified and excluded

Single View

MSI

Shadow Mask



Aggregated View

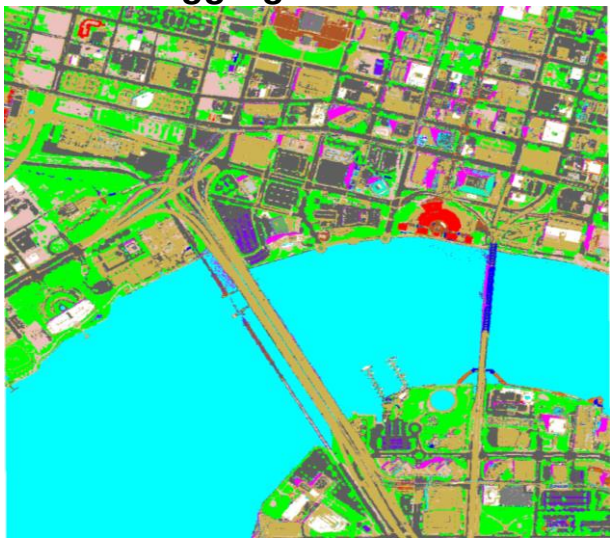


Image Alignment and Orthorectification

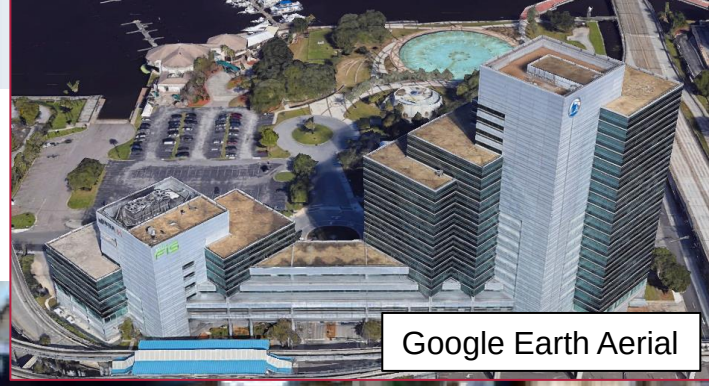
- Alignment:
  - WorldView imagery is subject to alignment error that must be corrected image-to-image
  - Perform bundle adjustment on the PAN and MSI to compute an XY shift for each image's RPC model
- Orthorectification
  - Central component in LEGO processing
  - Common reference frame in which to combine multiple images and data sources.
  - Many segmentation, classification, and modeling steps performed in 2.5D
  - Must take high-resolution DSM into account, true-orthorectification is key
  - Tall objects leave occluded regions

Original View

Occlusion Mask

Orthorectified View

Example: Original Image

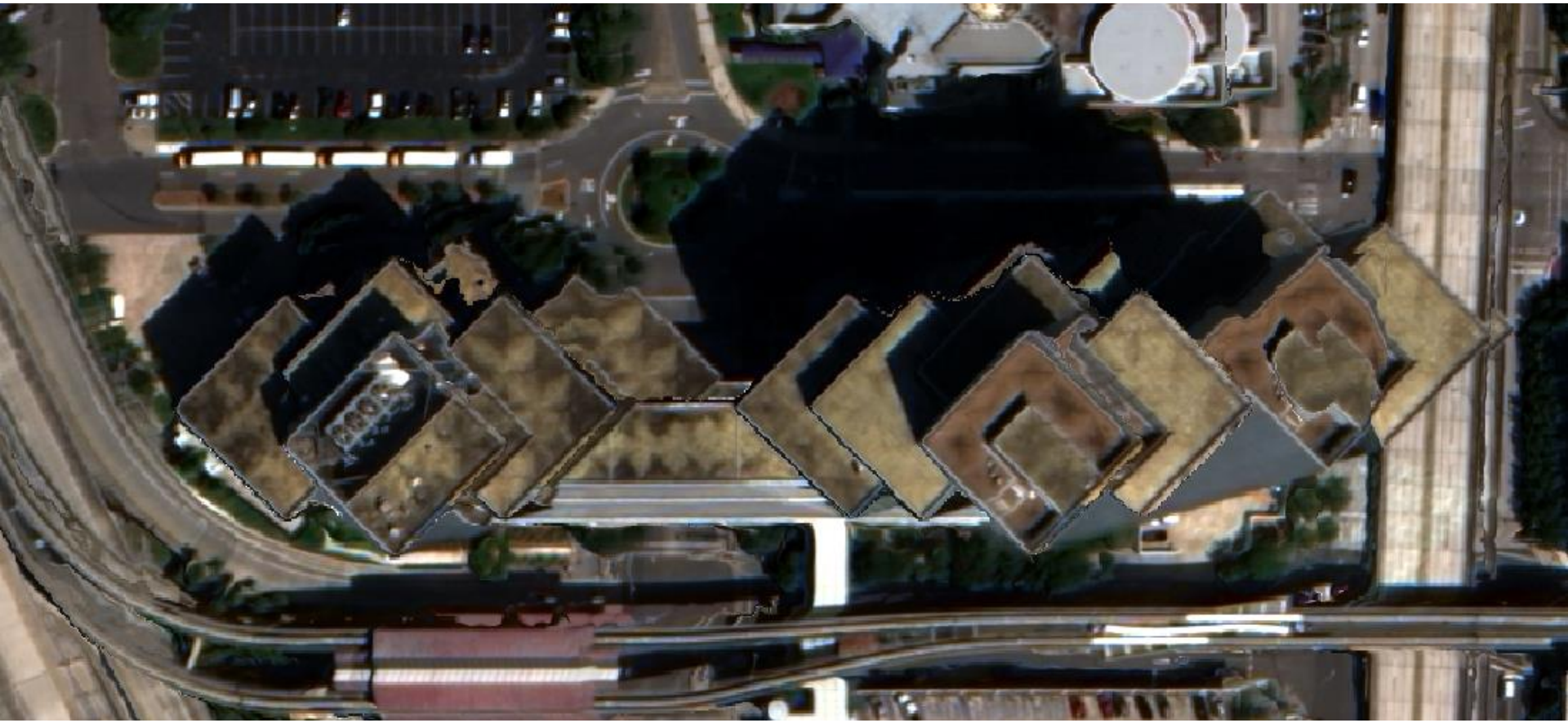


- Façade views change image-to-image based on parallax

Example: DTED-Orthorectification

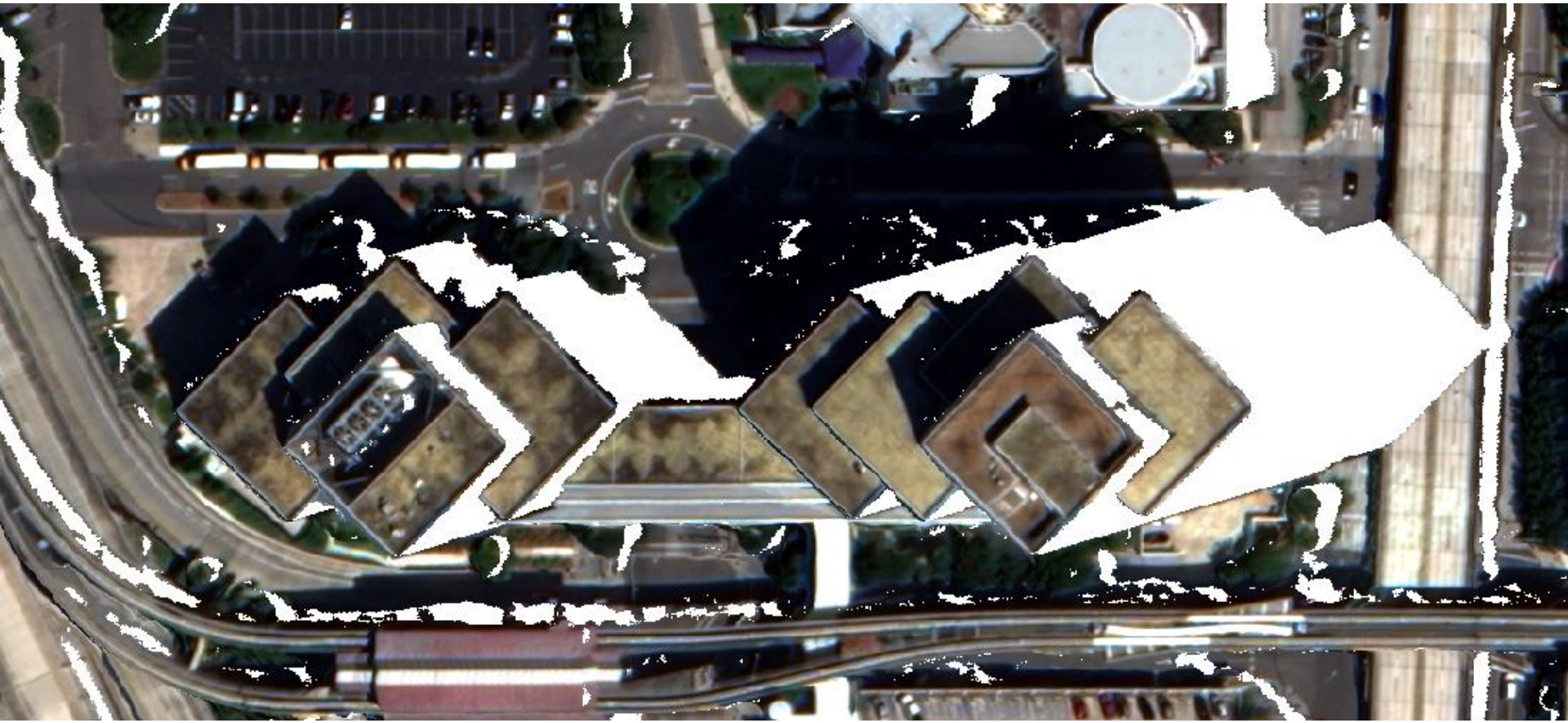
- Distortions due to low-frequency noise in DTED
- Even with better DTED, building facades still apparent.
- DTED does not represent buildings.

Example: DSM-Orthorectification (gdalwarp)



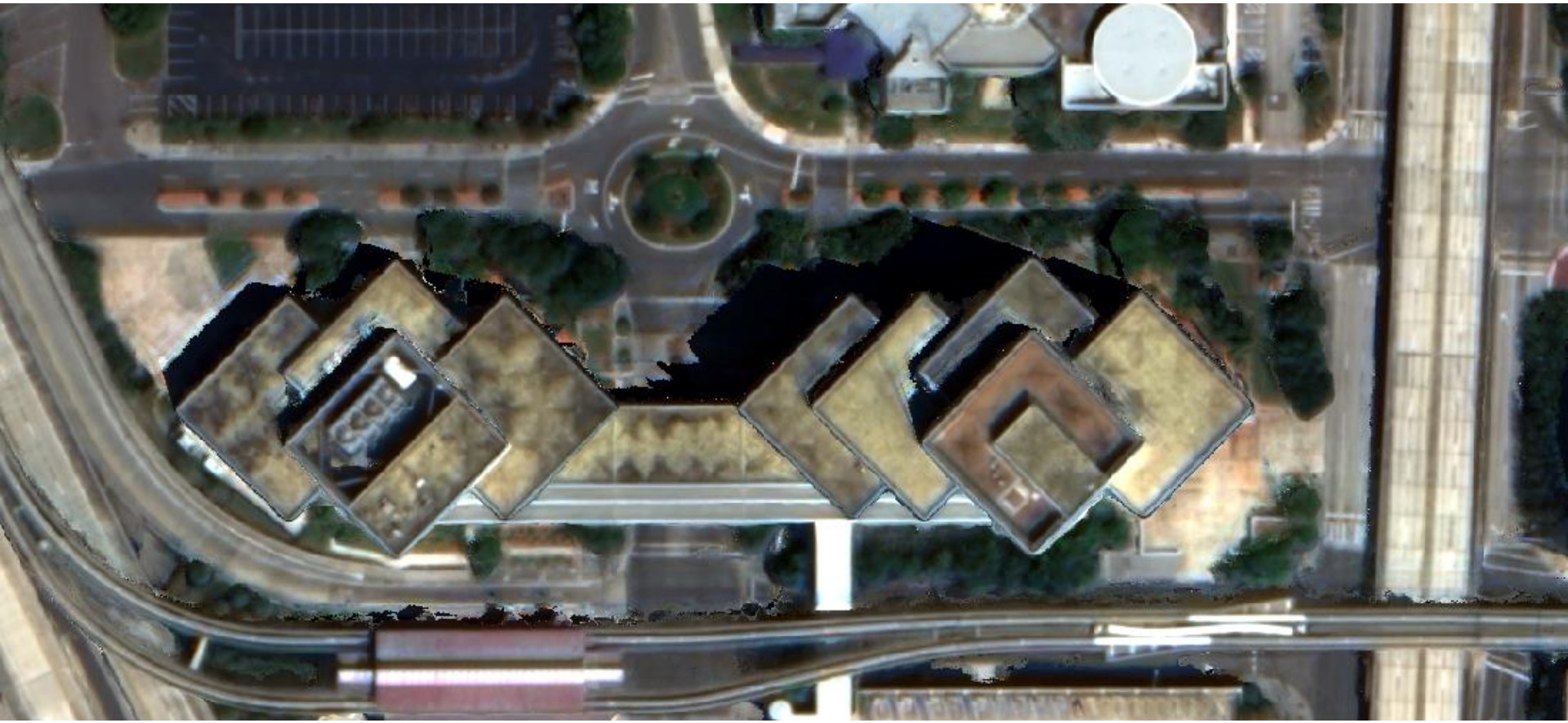
- Out of the box gdalwarp
- Facades are now removed
- Occlusions not handled
- Known as a ghosting effect

Example: LEGO (True) Orthorectification



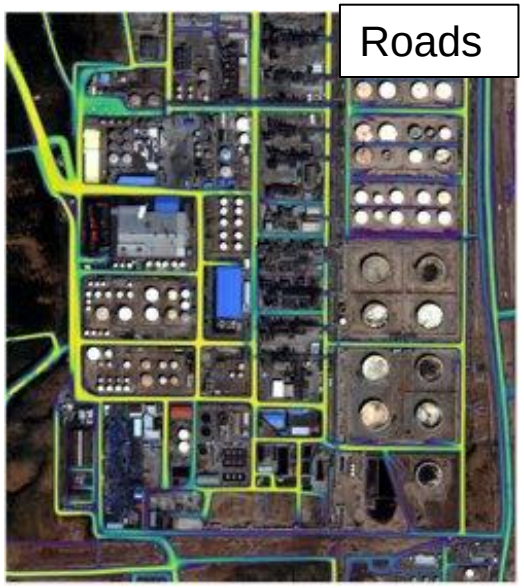
- Occlusion handled
- Unobserved areas are masked out
- Transient features present – cars, clouds, shadows

Example: LEGO Aggregated Ortho



- Aggregate across several images
- Infills shadows and occluded areas during aggregation.
- Certain areas are always in shadow, remain dark (sun-synchronous orbit)
- No transient features – no cars, or clouds

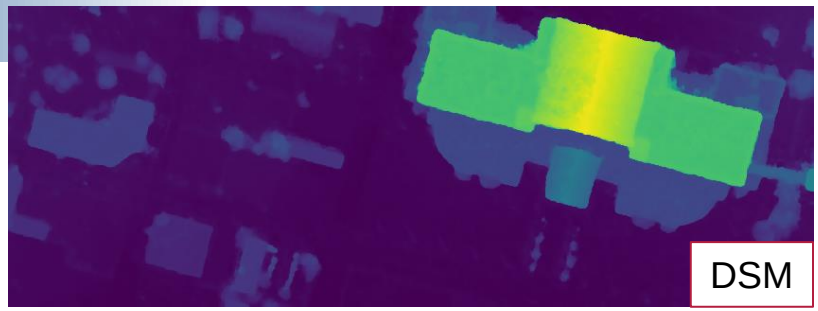
Semantic Segmentation

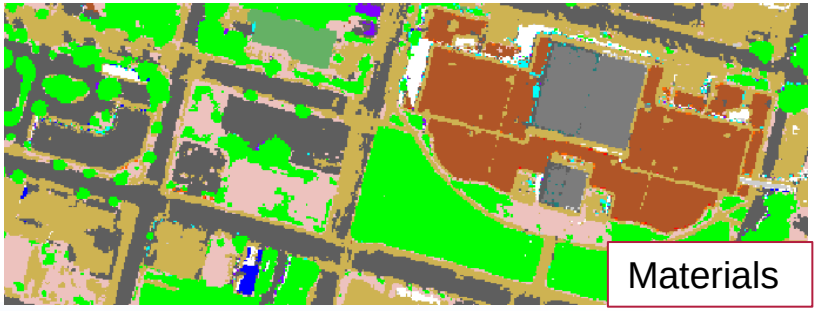


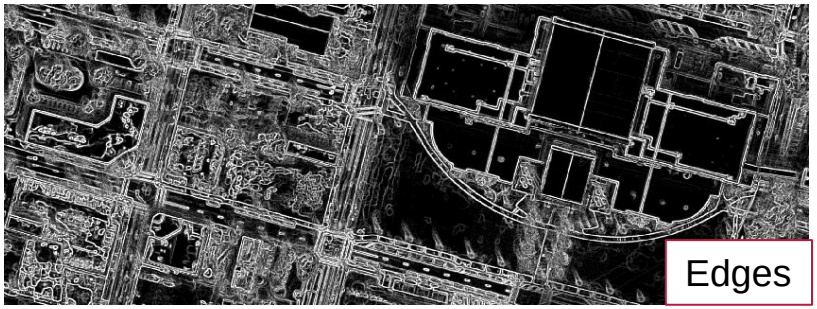
- ResUNet deep networks
- Trained on a variety of open-source datasets and Maxar imagery collected from global locations

Orthographic Products

All are critical for modeling

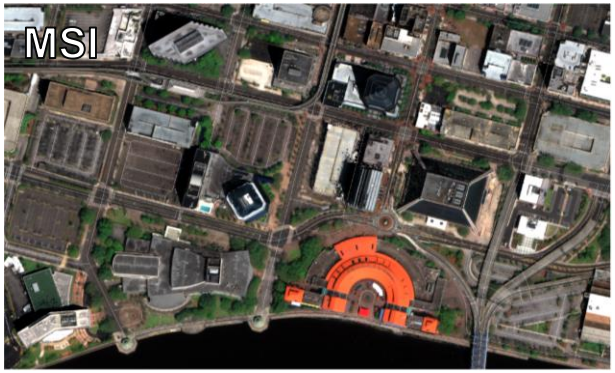


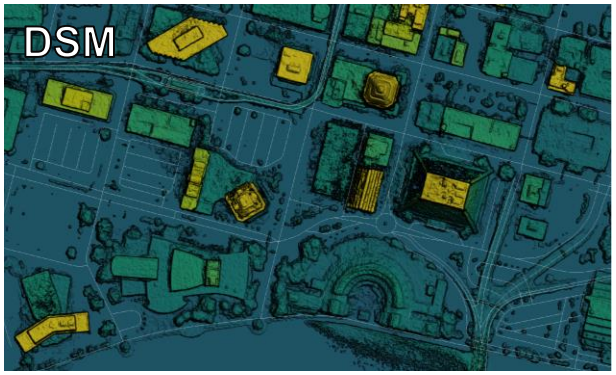


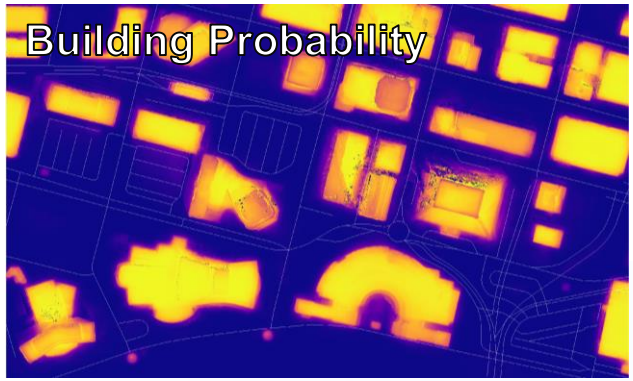


Post-Processing & Fusion

- Can correct common misclassifications by fusing land cover with derived CORE3D products
  - Estimation of the Surface leads to heights above ground
  - Semantic CNN segmentation leads to building masks
  - Bridge and raised roadway classification
  - Vegetation above 2 meters is classified as Trees
- Fusion approach moving forward:
  - Markov random field over naive Bayes probability map
  - Perform over-segmentation to generate super-pixels
  - Assign single data value to each super-pixel
  - Generate graph where super-pixels are nodes and edges connect adjacent super-pixels
  - Construct flow model and find regularized solution

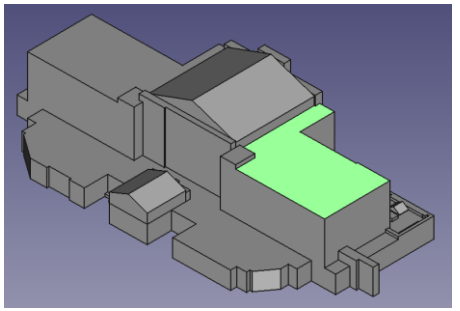






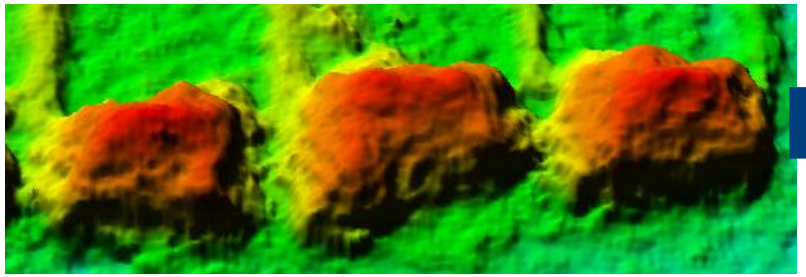
Modeling

- For each building cluster:
  - Several internal modeling methods are applied and compete
  - Best model is chosen based on accuracy to DSM and complexity.
- Bridges and Elevated roads modeled separately
- Modeling is performed in 2.5D
  - DSM is 2.5D
  - Satellite imagery is typically not oblique enough to capture true 3D features (overhangs, arches, etc.)
- Projective texturing selects best satellite view of each face.
- Output formats:
  - **CAD: STEP files (ISO 10303-21)**
  - Wavefront OBJ
  - COLLADA, KMZ
  - OpenFlight FLT
  - Initial support for OGC ODB

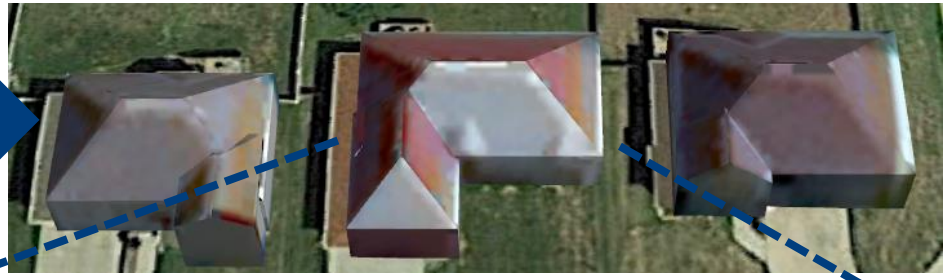


STEP file imported into FreeCAD

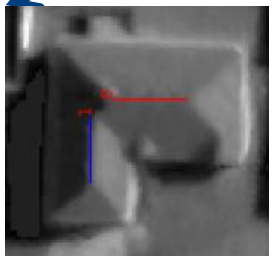
Model Fitting Leverages Multiple LEGO Data Products



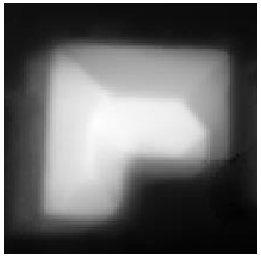
?



- Plane-fitting of roof components ***not possible*** from satellite-based DSM alone
- Roof component detections from imagery drive DSM fitting process



Pan Roof Component Shading Detection

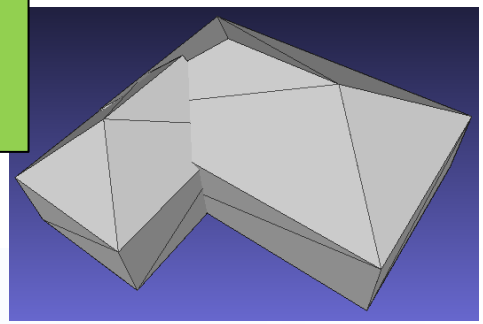


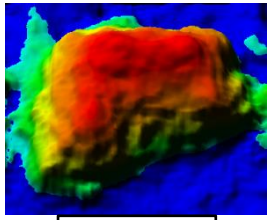
Building Detection

Material Classification

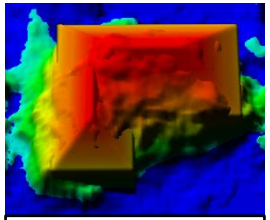
Stereo point counts

2 geometric primitives
34 triangles

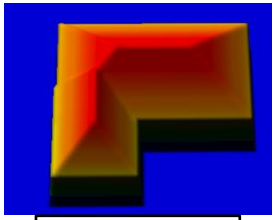




DSM

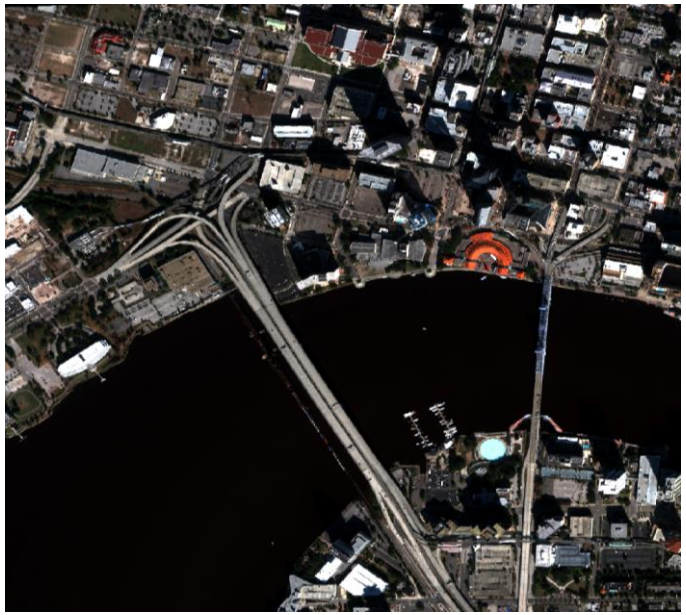


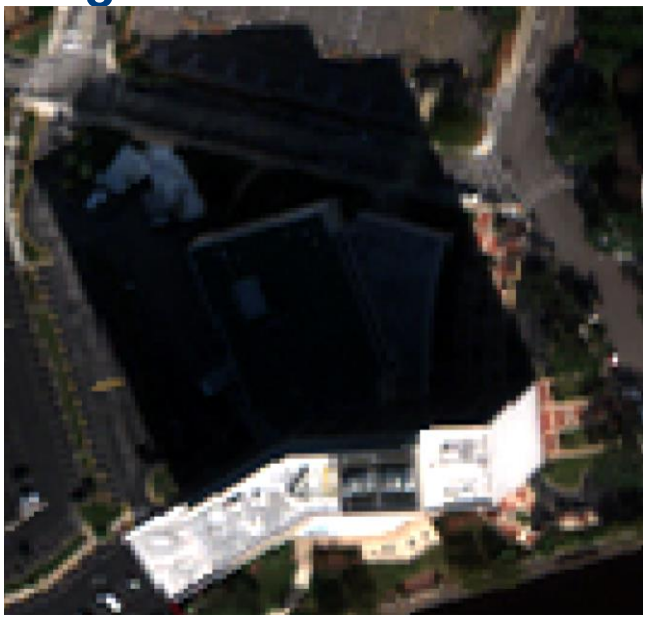
DSM + Roof fit

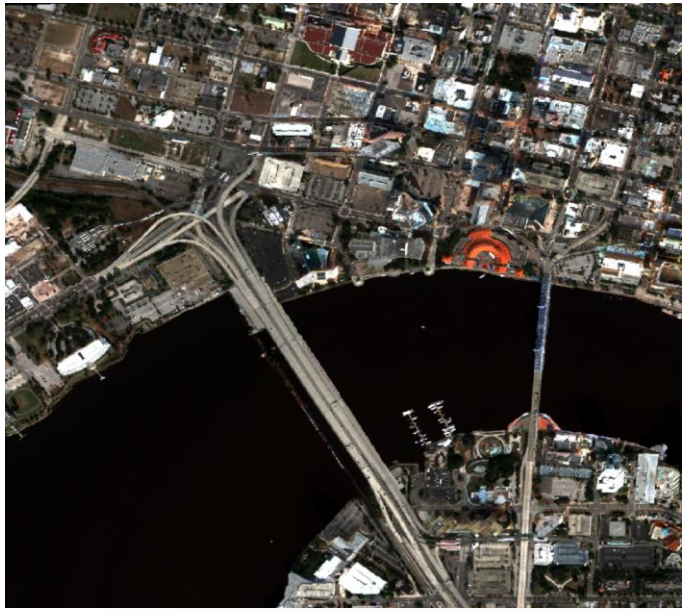


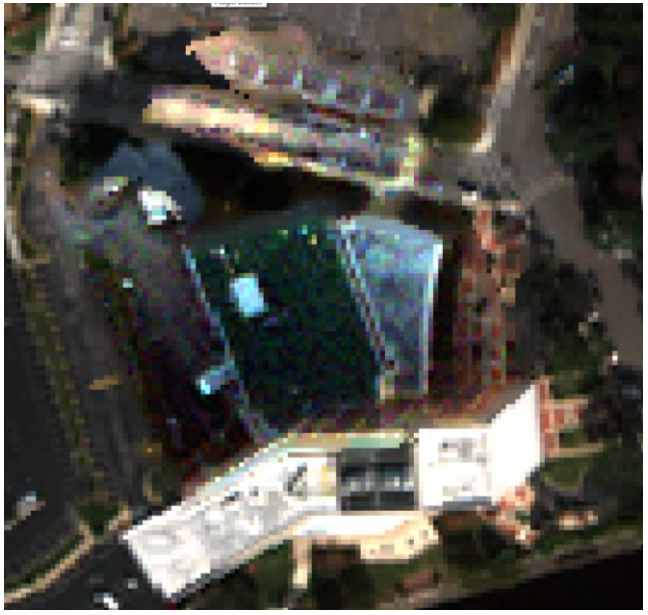
Roof fit

Shadow Correction Result via gamma control-based mapping

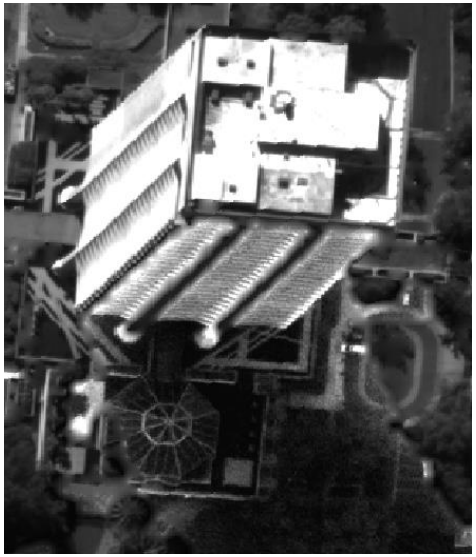








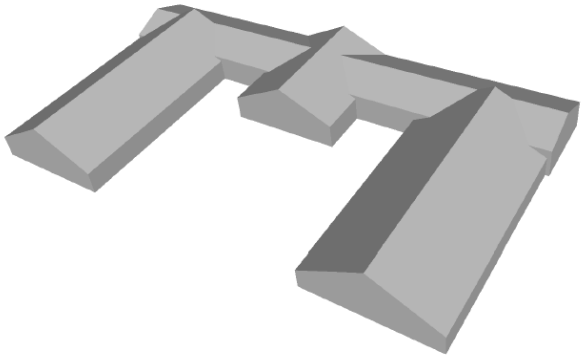
Recovered Facade

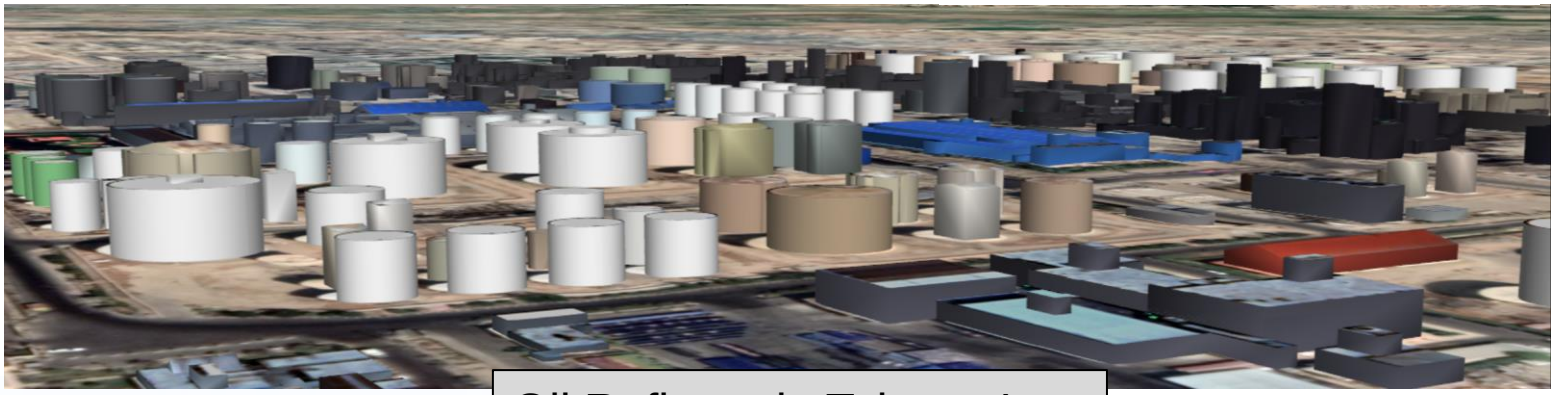


Examples

Yemen

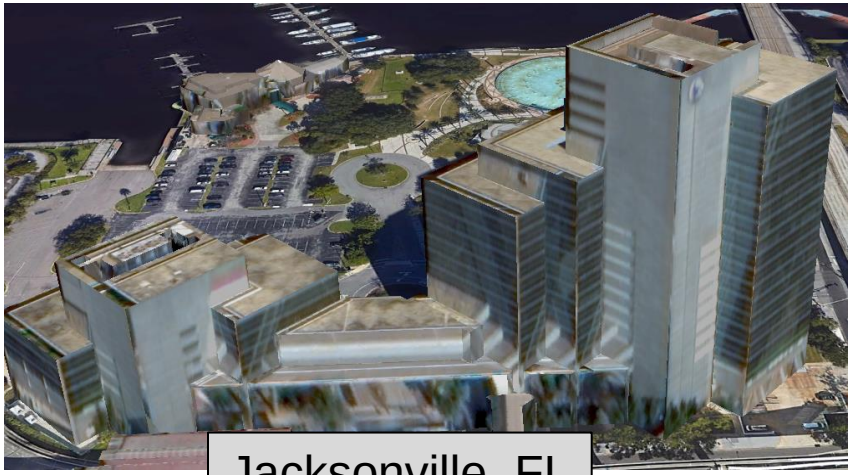
Google Earth Aerial





Oil Refinery in Tehran, Iran

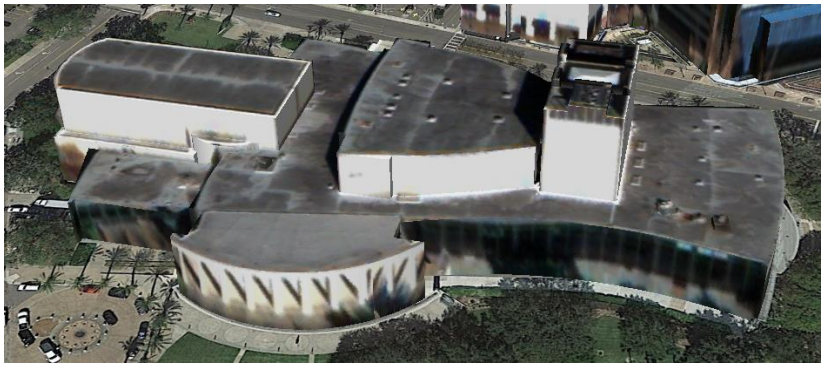
Examples



Jacksonville, FL

Phnom Penh, Cambodia

Comparison to Google Earth Aerial Models



ARA-LEGO

Google Earth Aerial

END