

A. Building Contour Detection and Height Estimation Problem

Limits: 2 sec., 256 MiB

Buildings are the most prominent man-made structures on Earth. An accurate footprint contour for building detection and building height prediction is critical in many applications such as urban planning, wireless infrastructure, population estimation etc. Currently, the most accurate building footprint contour and building height information is achieved from the LIDAR point cloud or multi-view images using dense matching. However, such data is hard to achieve. The building footprint contour extraction and height estimation using monocular satellite image has been a hot topic in the remote sensing field in recent years. For this contest, we are providing the monocular satellite images and label data with building contour and height information, to look for good building contour extraction and height estimation algorithm in the remote sensing field.



Training data

The training data set contains a set of 2200 monocular satellite images of Earth, each containing at least one building. For each image in the training data set, we provide a JSON file in a LabelMe format containing contours and heights for all the buildings on the image.

The training data archive can be downloaded using the link below. It contains two folders: **images** and **ground_truth_files**. The former contains images in **png** format and the latter contains data files in **json** format. An image and a corresponding data file share the same name. Each image has a size of exactly 512×512 pixels.

Test data

The test data contains a set of 4444 monocular satellite images of Earth each containing at least one building.

The test data archive can be downloaded using the link below. It contains images in **png** format. Each image has a size of exactly 512×512 pixels.

The testing data is internally split into two groups: provisional and final. During the contest, you will only see your score for the provisional set. After the end of the contest, the results for the final data set will be revealed.

You don't know which image corresponds to provisional or final set, so in order to make a submission you need to evaluate the whole test data set, which contains both provisional and final sets.

Submission

Your submission is a *zip* archive with the estimated data for the test data set. Each file in the archive should have the same name as one of the images from the test data set, a *json* extension, and should contain estimated data for the buildings in a corresponding image in a LabelMe format. For example, estimated data for image `aaaeexkyyo.png` should be named `aaaeexkyyo.json`.

Please note that the archive should contain only files described above. You should not create any folders or include any image data in the archive.

LabelMe format

LabelMe is an open-source cross-platform software used for visualizing and creation of polygonal label data. For more details on the data format and installing the software installation, please visit their GitHub page.

In order for your submission to be correctly evaluated, the `shapes` field of a root JSON object should contain the list of all buildings. The `points` field of each building object should contain the list of polygon vertices that represent the contour of the building. The `group_id` field should contain a height of the building in meters. The following is an example of a valid file with two buildings: the first one has a shape of a triangle and a height of nine meters; the second one has four vertices and a height of seven meters:

```
{
  "shapes": [
    {
      "points": [
        [ 316, 486 ],
        [ 307, 510 ],
        [ 312, 512 ]
      ],
      "group_id": 9
    },
    {
      "points": [
        [ 416, 457 ],
        [ 435, 446 ],
        [ 421, 423 ],
        [ 402, 434 ]
      ],
      "group_id": 7
    }
  ]
}
```

Please note that the fields described above are necessary for the scorer of this problem to correctly score your estimations. In order for your data to be compatible with LabelMe software, it might need some additional fields. Please refer to the LabelMe GitHub page or the data from the training data archive for further clarification.

Scoring

The score for the submission is calculated as a combination of three scores: precision score, recall score and height score:

$$Score = \max(0, [(PrecisionScore + RecallScore - 4 \cdot HeightScore) \cdot 5 \cdot 10^4]).$$

Your goal is to maximize this score.

Precision score

The precision score is defined as a portion of an area of the buildings in your output files that are correctly placed to a total area of buildings that you found:

$$PrecisionScore = \frac{CorrectlyPlacedArea}{TotalAreaInOutput} \cdot 100.$$

A building A from your output file is considered to be correctly placed if the total area of intersection of the polygon of building A with all the buildings from the ground truth for the corresponding image is greater than a half of the area of the polygon of the building A .

Please note that if there are no buildings in your output files, i.e. $TotalAreaInOutput = 0$ your precision score will be 0.

Recall score

The recall score is defined as a portion of an area of the buildings in the ground truth that are correctly identified to a total area of buildings in the ground truth files:

$$RecallScore = \frac{CorrectlyIdentifiedArea}{TotalAreaInGroundTruth} \cdot 100.$$

A building B from a ground truth file is considered to be correctly identified if the total area of intersection of the polygon of the building B with all the buildings from the corresponding file of your output is greater than a half of the area of the polygon of the building B .

Height score

The height score is defined as a square root of a mean square error of heights of all matched pairs of buildings:

$$HeightScore = \sqrt{\frac{\sum_{a,b:matched} (height(a) - height(b))^2}{|matched|}}.$$

Matched pairs of buildings consist of three groups:

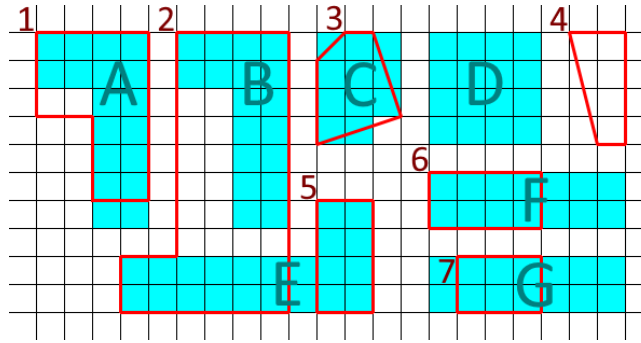
- Matched pairs. A building from your output A is considered to be matched with a building B from the ground truth if the area of intersection of buildings A and B is at least a half of the area of building A or at least a half of the area of building B :

$$area(intersection(A, B)) \geq \frac{\min(area(A), area(B))}{2}.$$

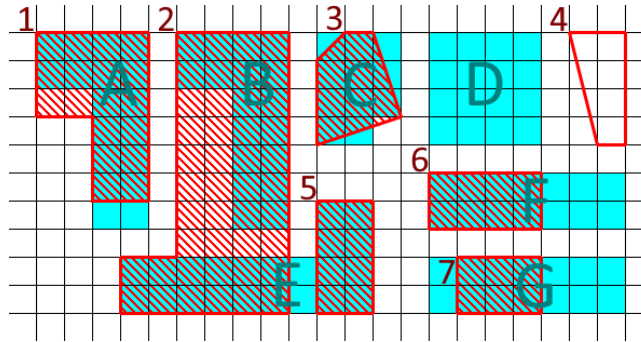
Please note that a building might be matched with more than one other buildings.

- Unmatched buildings from output. Each building from your output that is not matched with any of the buildings from the ground truth files is considered to be matched with a building of height 0.
- Unmatched buildings from ground truth. Each building from the ground truth file that is not matched with any of the buildings from your output is considered to be matched with a building of height 0.

Scoring example

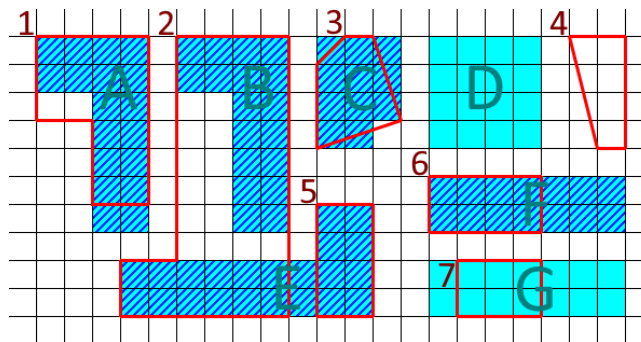


Let's consider an example where we have 7 buildings in the ground truth file which are displayed on the image in blue and named by letters A, B, ..., G and 7 buildings in the output file which are displayed on the image in red and named by numbers 1, 2, ..., 7. Buildings A, B, C and D have the height of 10 meters, buildings E, F and G have the height of 100 meters; buildings 1, 2, 3, 4 have the height of 11 meters, building 5, 6 and 7 have the height of 100 meters.



Building 1, 2, 3, 5, 6 and 7 are considered to be correctly placed while building 4 is not correctly placed, so the precision score in this case will be:

$$PrecisionScore = \frac{18 + 44 + 8.5 + 8 + 8 + 6}{18 + 44 + 8.5 + 6 + 8 + 8 + 6} \cdot 100 = \frac{92.5}{98.5} \cdot 100 \approx 93.9086294.$$



Buildings A, B, C, E and F are considered to be correctly identified while buildings D and G are not correctly identified, so the recall score in this case will be:

$$RecallScore = \frac{18 + 18 + 11 + 22 + 14}{18 + 18 + 11 + 16 + 22 + 14 + 14} \cdot 100 = \frac{83}{113} \cdot 100 \approx 73.4513274.$$

The following pairs of buildings are considered to be matched: A and 1 (heights 10 and 11), B and 2 (heights 10 and 11), E and 2 (heights 100 and 11), C and 3 (heights 10 and 11), E and 5 (heights 100 and 100), F and 6 (heights 100 and 100), G and 7 (heights 100 and 100); while buildings D (height 11) and 4 (height 10) are not matched with any of the other buildings. The height score in this case will be:

$$\begin{aligned}
 HeightScore &= \sqrt{\frac{(10-11)^2 \cdot 3 + (100-11)^2 + (100-100)^2 \cdot 3 + (0-11)^2 + (10-0)^2}{9}} = \\
 &= \sqrt{\frac{1 \cdot 3 + 7921 + 0.00 \cdot 3 + 121 + 100}{9}} \approx 30.0832179.
 \end{aligned}$$

The total score is:

$$Score = \lfloor (93.9086294 + 73.4513274 - 4 \cdot 30.0832179) \cdot 5 \cdot 10^4 \rfloor = 2351354.$$

Invalid files

If you submit an invalid archive file, you will receive a **Compilation Error** verdict.

A file in the archive that corresponds to one of the images from the test data set is considered invalid if:

- the file is not a valid JSON file,
- the file is not in a LabelMe format,
- at least one of the polygons has self-intersections,
- the number of buildings exceeds 1000,
- the number of vertices in a polygon in any of the buildings is larger than 300,
- the total number of vertices in a file is larger than 5000,
- the height of any building is negative or exceeds 1000,
- a coordinate of a building's polygon point is negative or exceeds 512,
- there are two buildings A and B such that the area of intersection between A and B is larger than 10% of the area of the smaller of the buildings A , B :

$$area(intersection(A, B)) > 0.1 \cdot \min(area(A), area(B)).$$

- the scoring script is not able to correctly read and process the file for any other reason.

Each invalid file is considered by the scoring script to be empty, i.e., containing no buildings. Please refer to the scoring script for more clarifications. You can find the link to the scoring script below.

Scoring script

A script used for scoring is available for download using the link below. Feel free to use it for local testing of your outputs on the training set. Also, you can refer to the code of the scorer for any clarifications of the scoring function.

In order to run the scoring script, you should have Python3 installed on your machine. First, unarchive the scoring script to a directory and run the following command to install required dependencies:

```
pip3 install -r requirements.txt
```

After that, you can run the scoring script with the following command:

```
python3 scorer.py -o {path to a folder with your output files} \
                  -g {path to a folder with ground truth files}
```

The script will print a total score to stdout.

Additional notes

- The images for training, provisional and final data sets are chosen randomly from the pool of images without any bias.
- The source of the label data is the same for training and testing data sets.
- The label data is produced by a third party company manually and may contain a small portion of invalid labels.
- The contour of the building is defined as the footprint of the building on the ground, however in case of small buildings the contour may sometimes contain a roof of the building as well.
- Some of the ground truth data files may have a small offset to the building contours compared to their position on the images, which is not necessarily constant for all buildings on the image.
- A small portion of the ground truth data files may contain a rather large offset of the building contours.
- A small portion of the ground truth data files may contain overlapping or incorrect labels.
- A size limit for your submission is 15 MiB.
- A size limit for unarchived data is 256 MiB.
- You can submit your code once every 6 hours, and you will get feedback with your score for the provisional data set.

Links

- Training data: [mlc_training_data.zip](#)
- Test data: [mlc_test_images.zip](#)
- Scoring script: [scorer.zip](#)