Supplementary Material of CVPR 2016 publication: Cataloging Public Objects Using Aerial and Street-Level Images – Urban Trees

Jan D. Wegner[a,1] Steve Branson[b,1] David Hall[b] Konrad Schindler[a] Pietro Perona[b]
ETH Zürich[a] California Institute of Technology[b]

A. Supplementary Material: Google maps imagery

In this section we provide the form of the projection functions $\mathcal{P}_v(\ell, c)$ that convert from geographic locations to pixel locations in aerial view and street view images. We give the form of the inverse function $\mathcal{P}_v^{-1}(\ell', c)$ that converts from pixel locations to geographic coordinates.

Aerial images: Aerial view imagery in Google maps is represented using a Web Mercator projection, a type of cylindrical map projection that unwraps the spherical surface of the earth into a giant rectangular image. A pixel location $\ell' = (x, y)$ is computed from a geographic location $\ell = (\text{lat}, \text{lng})$ in radians, as $(x, y) = \mathcal{P}_{av}(\text{lat}, \text{lng})$:

$$\begin{aligned} x &= 256(2^{\text{zoom}})\left(\text{lng} + \pi\right)/2\pi \\ y &= 256(2^{\text{zoom}})\left(1/2 - \ln\left[\tan\left(\pi/4 + \text{lat}/2\right)\right]/2\pi\right) \end{aligned} \tag{10}$$

where zoom defines the resolution of the image.

Using simple algebraic manipulation of Eq. 10, the inverse function $(\text{lat}, \text{lng}) = \mathcal{P}_{av}^{-1}(x, y)$ can be computed as:

$$\begin{aligned} \text{lng} &= \frac{\pi x}{128(2^{\text{zoom}})} - \pi \\ \text{lat} &= 2\tan^{-1}\left(\exp\left(\pi - \frac{y\pi}{128(2^{\text{zoom}})}\right)\right) - \frac{\pi}{4} \end{aligned} \tag{11}$$

Map images: Map images contain drawings of streets, buildings, parks, *etc*. They are pixelwise aligned with aerial view images and subject to the same projection functions.

Street view images: Each Google street view image captures a full 360° panorama and is an equidistant cylindrical projection of the environment as captured by a camera mounted on top of the Google street view car (see Figure 2(a)). The car is equipped with other instruments to record its camera position c, which includes the camera's geographic coordinates $\text{lat}(c)$, $\text{lng}(c)$, and the car's heading $\text{yaw}(c)$ (measured as the clockwise angle from north). On urban roads, Google street view images are typically spaced around 15 m apart.

Using simple algebraic and trigonometric manipulation of Eq. 2, the inverse function $(\text{lat}, \text{lng}) = \mathcal{P}_{sv}^{-1}(x, y)$ can be computed as:

$$\begin{aligned} \text{lat} &= \text{lat}(c) + \arcsin(e_y, R) \\ \text{lng} &= \text{lng}(c) + \arcsin(e_x/\cos(\text{lat}(c)), R) \end{aligned} \tag{12}$$

where we have first obtained z, e_x, e_y by reversing Eq 2:

$$\begin{aligned} z &= -h/\tan\left(-y\frac{\pi}{H} + \pi/2\right) \\ e_x &= \sin\left(x\frac{2\pi}{W} - \pi + \text{yaw}(c)\right) z \\ e_y &= \cos\left(x\frac{2\pi}{W} - \pi + \text{yaw}(c)\right) z \end{aligned} \tag{13}$$

[1] joint first authorship

B. Supplementary Material: Piecewise CRF learning

In Section 4.2, we described a piecewise learning method for learning each potential function in Eq. 3. We first learn parameters for each potential term separately, optimizing conditional probabilities:

$$\alpha^* = \arg\max \sum_{t \in \mathcal{D}_t} \log p(t|T) \tag{14}$$

$$\log p(t|T) = \Lambda(t, T; \alpha) - Z_1 \tag{15}$$

$$\beta^* = \arg\max \sum_{t \in \mathcal{D}_t} \log p(t|\mathrm{mv}(t)) \tag{16}$$

$$\log p(t|\mathrm{mv}(t)) = \Omega(t, \mathrm{mv}(t); \beta) - Z_2 \tag{17}$$

$$\delta^* = \arg\max \sum_{t \in \mathcal{D}_t} \log p(t|\mathrm{av}(t)) \tag{18}$$

$$\log p(t|\mathrm{av}(t)) = \Psi(t, \mathrm{av}(t); \gamma) - Z_3 \tag{19}$$

$$\gamma^* = \arg\max \sum_{t \in \mathcal{D}_t} \sum_{s \in \mathrm{sv}(t)} \log p(t|s) \tag{20}$$

$$\log p(t|s) = \Phi(t, s; \delta) - Z_4 \tag{21}$$

where normalization terms $Z_{1\ldots4}$ are computed for each training example individually to make probabilities sum to 1. Note that the last two terms match the learning problems used in R-CNN training (which optimizes a log-logistic loss), and the first two terms are simple logistic regression problems.

Next, we fix $\alpha, \beta, \delta, \gamma$ and use the validation set to learn scalars k_1, k_2, k_3, k_4 to weight each potential term separately. Here, we optimize detection loss (measured in terms of average precision) induced by our greedy inference algorithm. This allows us to learn a combination of the different sources of information while optimizing a discriminative loss. We iteratively select each scalar k_i using brute force search.

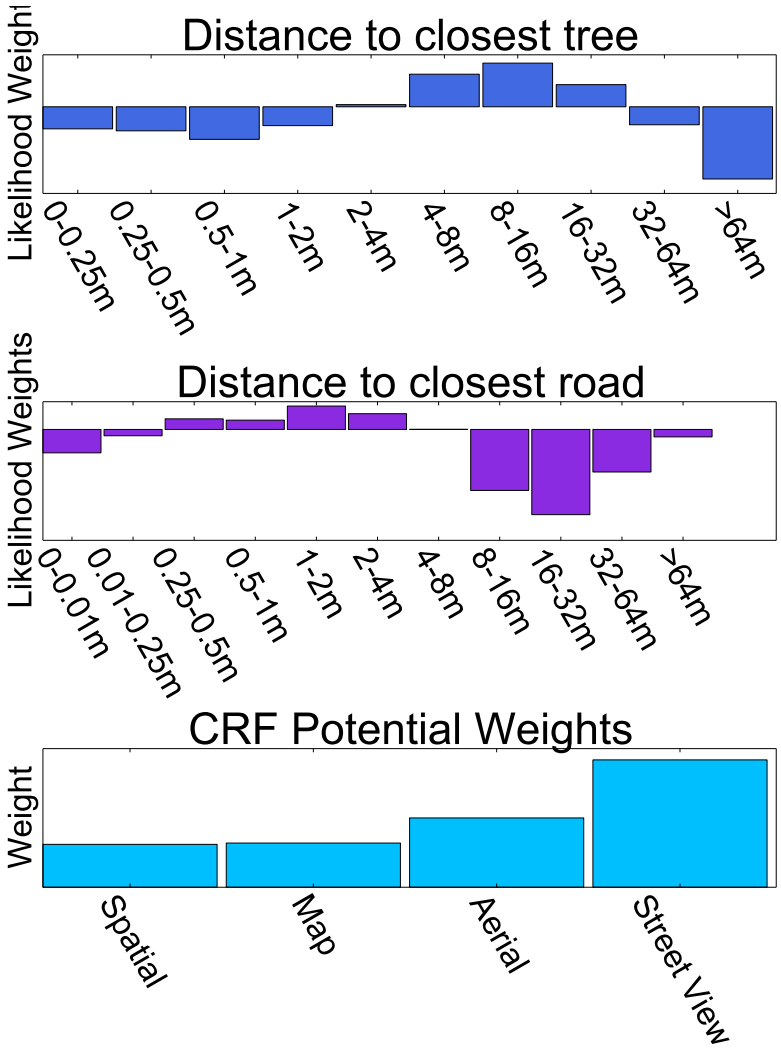


Figure 8. Visualization of learned spatial context parameters (top), map potential parameters (center), and (bottom) scalars k_1, k_2, k_3, k_4 for combining detection CRF potentials.

C. Supplementary Material: Visualization of learned model weights

In Figure 8 we visualize components of the learned model. The first histogram shows learned weights α for the spatial context potential. Intuitively, we see that the model penalizes most strongly trees that are closer than 2m or further than 32m to the nearest tree. The 2nd histogram shows learned map weights β–we see that the model penalizes most heavily trees that are too close to the road (less than .25m) or too far from the road (greater than 8m). The last histogram shows learned weights on each CRF potential term–these match earlier results that streetview and aerial images are most important.

Error Name	Error Description	Detection examples
Private tree	A detection corresponds to a real tree. The tree is on private land, whereas the ground truth inventory only includes trees on public land, resulting in a false positive.	F10, F11, J6, J7, O11
Missing tree	A tree on public land appears to be missing from the inventory (test set), which is older than the Google imagery (results in a false positive). Usually a recently planted tree.	C10, D5, F7, G12, J5, N10, N11, O9
Extra tree	An extra tree appears in the ground truth inventory, probably due to human error.	O5
Telephone pole	False positive because a telephone pole or lamp post resembles the trunk of a tree. Usually happens when foliage of a nearby tree also appears near the pole.	B6, B10, B11, F6, F9, F14, G14, M14, M16, P4
Duplicate detection	A single tree is detected as 2 trees.	B7, B9, F13, G13, M15, O12
Localization Error	A detected tree is close to ground truth, but not within the necessary distance threshold, resulting in a false positive and negative. Usually happens when the camera position and heading associated with a Google street view panorama are slightly noisy.	E12/E7, G11/G9, K7/K4, L9/L5, N7/N6, N8/N1
Weak detection	A tree is detected in a street view, but its combined confidence is just below the necessary threshold, resulting in a false negative.	C2, C3, C5, C8, D2, E8, E9, F1, K2, K3, L3, L4
Occluding object	A tree is occluded (*e.g.*, by a car or truck) in street view, resulting in a false positive or error localizing the base of the trunk.	E5, F3

D. Supplementary Material: Detection Qualitative Results

In Figures 9-24, we show detailed qualitative results of our tree detection system on 16 random geographical regions. Each figure shows one such example region and one example is shown per page. In the top row, the first column shows the input region, with blue circles representing the location of available street view cameras. The 2nd column shows results and error analysis of our full detection system, which combines aerial, street view, and map images and spatial context. Here, true positives are shown in green, false positives are shown in red, and false negatives are shown in magenta. The 3rd column shows single view detection results using just aerial images. The bottom two rows show two selected street view images–the images are numbered according to their corresponding blue circle in the 1st row, 1st column. The 2nd row shows single view detection results using just street view images. The bottom row visualizes the same results and error analysis visualized in the 1st row, 2nd column, with numbers in the center of each box matching across views.

E. Supplementary Material: Detection Error Analysis

We attempt to come up with human understandable explanations for the main types of detection errors, as measured on the test set of the publicly available inventory of public trees in Pasadena (see Section 6). We came up with 8 categories that can explain all 56 errors in the qualitative results shown Figures 9-24, and manually assign each error to one of the categories. An explanation for each assignment is included in the caption for each figure. In the table above, we list each error category and the detection examples assigned to them. Each detection example is denoted by a number and a letter, where the letter denotes the figure number, and the number denotes the bounding box number. For example *B3* corresponds to bounding box 3 (see numbered boxes in Figure 10) in example B (Figure 10).

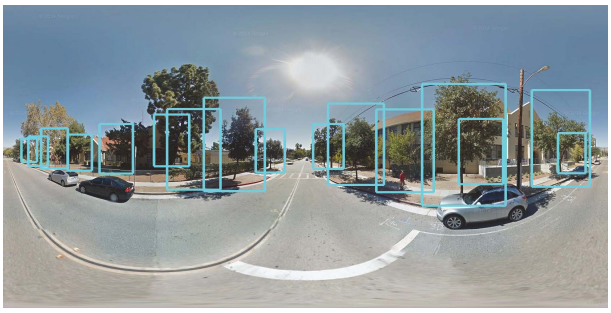
We note that at least $14/56 = 25\%$ of measured errors arose due to issues with the ground truth Pasadena inventory test set–these were correct detections that were penalized as false positives because the dataset does not include trees on private land or because the inventory is less recent than Google maps imagery. It is also likely that many of the 12 false negatives due to weak detections partially arose due to this issue–the algorithm attempted to learn to distinguish between public and private trees, and thus a strict detection threshold was required. Lastly, 12 errors occured due to detections that were close to ground truth trees, but localization was not quite accurate enough to meet the requisite distance threshold.

Thus it is likely that roughly $25 - 68\%$ of measured detection errors occurred due to a benchmark that was possibly too strict. We are currently in touch with the city of Los Angeles to obtain a record of the delineation between public and private land–this is an obvious addition that should improve results significantly. It also appears that many errors were caused by noise in the camera position and heading meta data associated with Google street view panoramas. An area of future research is to attempt to use detection matches between different views to better calibrate cameras.

true positive **false positive** **false negative** **single view detection**

A

Input Region **Combined Det. Err. Vis.** **Aerial Detections**



1. Street View Detections

2. Street View Detections

1. Combined Det. Err. Vis.

2. Combined Det. Err. Vis.

Figure 9. **Example A:** The detection system has correctly detected 7 trees, with no false positives or negatives. It has also correctly rejected two large trees (top right of the input region) that are just on private property.

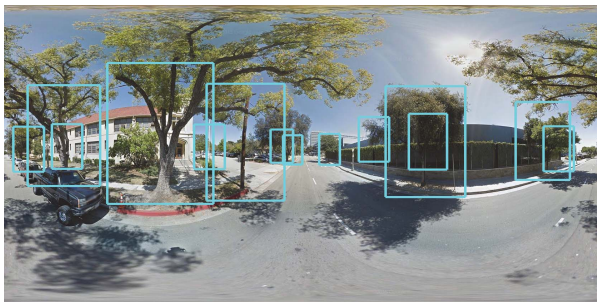
B

Input Region

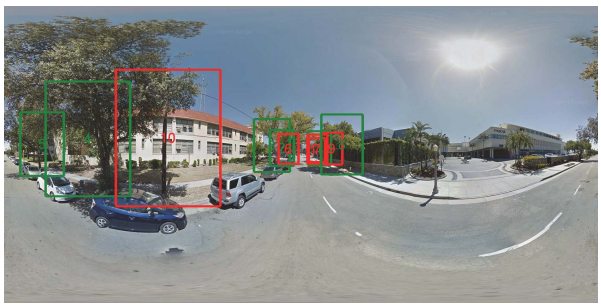
Combined Det. Err. Vis.

Aerial Detections

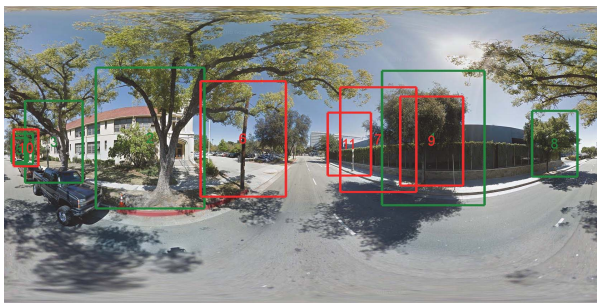
3. Street View Detections



5. Street View Detections



3. Combined Det. Err. Vis.

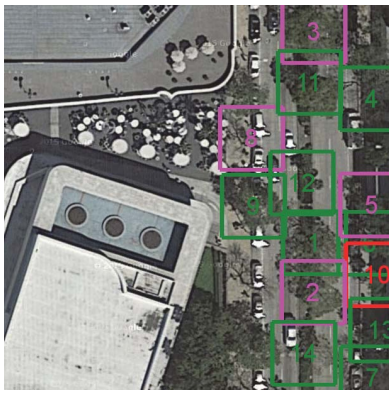


5. Combined Det. Err. Vis.

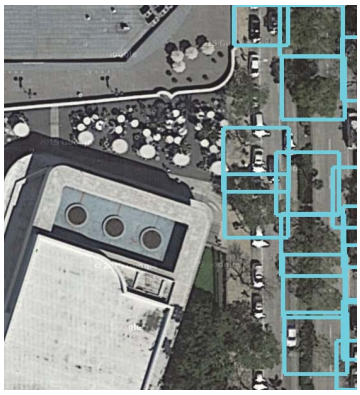
Figure 10. **Example B:** The detection system has correctly detected 6 trees, and correctly rejected a couple of trees on private property. However, it has 5 false positives, including 3 false positives caused by wooden telephone poles near foliage (boxes 6, 10, 11), and 2 trees that were split into duplicate detections.

C

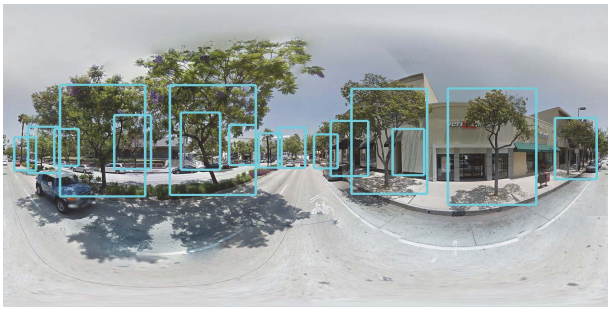
Input Region



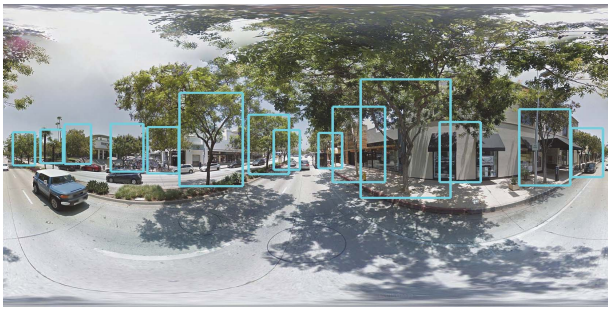
Combined Det. Err. Vis.



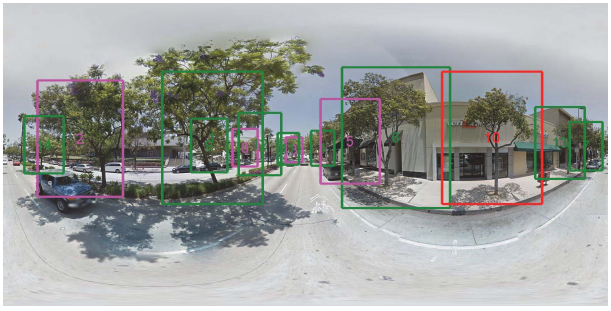
Aerial Detections



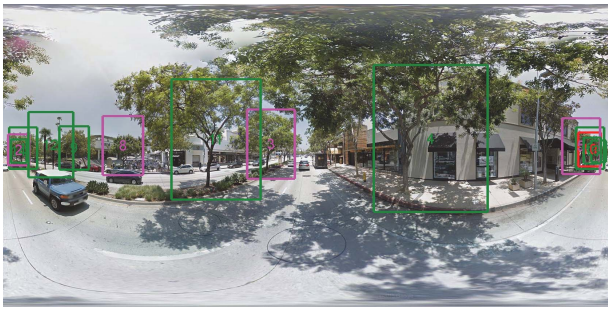
3. Street View Detections



4. Street View Detections



3. Combined Det. Err. Vis.



4. Combined Det. Err. Vis.

Figure 11. **Example C:** The detector has correctly detected 9 trees. It has correctly rejected several trees on private land. It has one measured false positive (box 10); however, closer inspection reveals that a tree is actually present and on public land (see red box in the 1st streetview image)–most likely the tree was planted after 2013 (when the inventory was catalogued). It has 4 false negatives (boxes 2,3,5,8), all of which were weak detections with scores that fell just below the detection threshold, as evidenced by single view detections in the streetview and aerial images.

D

Input Region | **Combined Det. Err. Vis.** | **Aerial Detections**

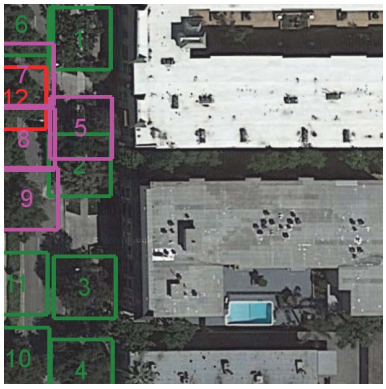
6. Street View Detections | **8. Street View Detections**

6. Combined Det. Err. Vis. | **8. Combined Det. Err. Vis.**

Figure 12. **Example D:** The detector has correctly detected 4 trees and correctly rejected several trees on private land. It has one measured false positive (box 5); however, closer inspection reveals that a tree is actually present and on public land (see red box in the 1st streetview image)–most likely the tree was planted after 2013 (when the inventory was catalogued). It also has 1 false negative (box 2), which was a weak detection with scores that fell just below the detection threshold. The score was probably weaker than normal because a car partially occludes the base of the trunk in the nearest street view.

E

Input Region



Combined Det. Err. Vis.

Aerial Detections

2. Street View Detections

4. Street View Detections

2. Combined Det. Err. Vis.

4. Combined Det. Err. Vis.

Figure 13. **Example E:** The detector has correctly detected 7 trees and correctly rejected upwards of 7 trees on private land. The detector correctly detected another tree; however, the localization was inaccurate, resulting in a false positive (box 12) and false negative (box 7). The inaccuracy probably occurred because the recorded camera position and heading of the Google street view camera was noisy, which is a common problem and subject of future research. This is probably the case because box 7 is in the correct place in the street view image but not the aerial image. This camera noise also probably contributed to two false negatives due to weak detections with scores that fell just below the detection threshold (boxes 8, 9). These trees were detected in the street view images, but misalignment between aerial and street views weakened combined scores. One last false negative (box 5) also had a weak detection score that was just below the required threshold–the cause was most likely a parked car that occluded the base of the trunk.

F

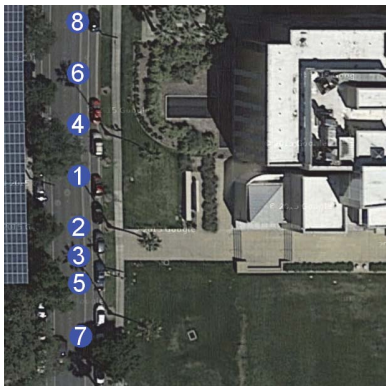
Input Region **Combined Det. Err. Vis.** **Aerial Detections**

4. Street View Detections **5. Street View Detections**

4. Combined Det. Err. Vis. **5. Combined Det. Err. Vis.**

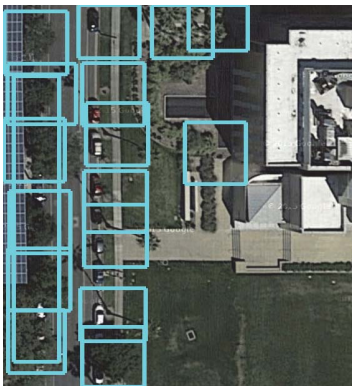
Figure 14. **Example F:** The detector has correctly detected 6 trees and correctly rejected upwards of 10 trees on private land. However, there were 2 measured false positives (boxes 10, 11) that were actual trees but not included in the test set inventory because they are on private land. A 3rd measured false positive appears to be a valid tree on public land, but was missing from the test set inventory for unknown reason. 3 other false positives occurred due to wooden telephone poles near foliage. One false negative occurred due to a weak detection with score that fell just below the detection threshold (boxes 1), as it was detected in the street view image. A second false negative probably occurred because a white van occluded the base of the trunk.

G



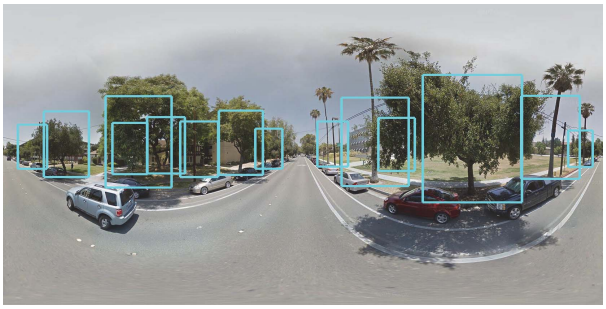
Input Region

Combined Det. Err. Vis.



Aerial Detections

6. Street View Detections



7. Street View Detections

6. Combined Det. Err. Vis.

7. Combined Det. Err. Vis.

Figure 15. **Example G:** The detector has correctly detected 9 trees. Another detected tree (box 12) appears to be a valid tree that was missing from the test set for unknown reason, resulting in a false positive. The detector correctly detected another tree; however, the localization was inaccurate, resulting in a false positive (box 11) and false negative (box 9). The inaccuracy probably occurred because the recorded camera position and heading of the Google street view camera was noisy, which is a common problem and subject of future research. This is probably the case because box 11 is in the correct place in the street view image but not the aerial image. This noise probably contributed to another false positive (box 13) due to a duplicate detection (a single tree detected twice), which often happens when street view detections are misaligned with aerial detections. One last false positive (box 14) occurred due to a wooden telephone pole.

H

Input Region **Combined Det. Err. Vis.** **Aerial Detections**

Figure 16. **Example H:** This geographical region occurs in an area where no street views are present. The detector correctly finds that there are no trees present.

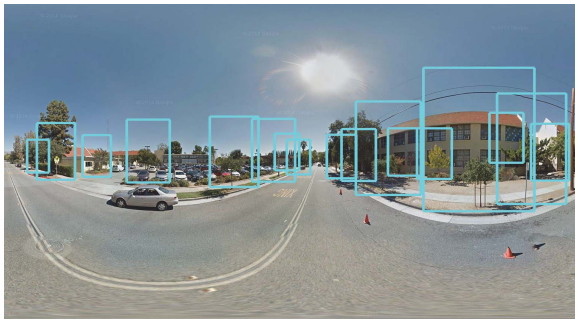
I

Input Region

Combined Det. Err. Vis.

Aerial Detections

1. Street View Detections



5. Street View Detections

1. Combined Det. Err. Vis.

5. Combined Det. Err. Vis.

Figure 17. **Example I:** The detector correctly detected 3 trees and correctly rejected 3 large trees on private land, with no measured errors.

J

Input Region **Combined Det. Err. Vis.** **Aerial Detections**

1. Street View Detections **4. Street View Detections**

1. Combined Det. Err. Vis. **4. Combined Det. Err. Vis.**

Figure 18. **Example J:** The detector correctly detected 4 trees and correctly rejected several trees on private land. There are 3 measured false positives; however, 2 are actual trees but on private land (boxes 6,7), and 1 is a valid tree that is on public land (box 5) but not included in the test set inventory. It was probably missing because it appears to be a recently planted tree and the inventory was collected in 2013.

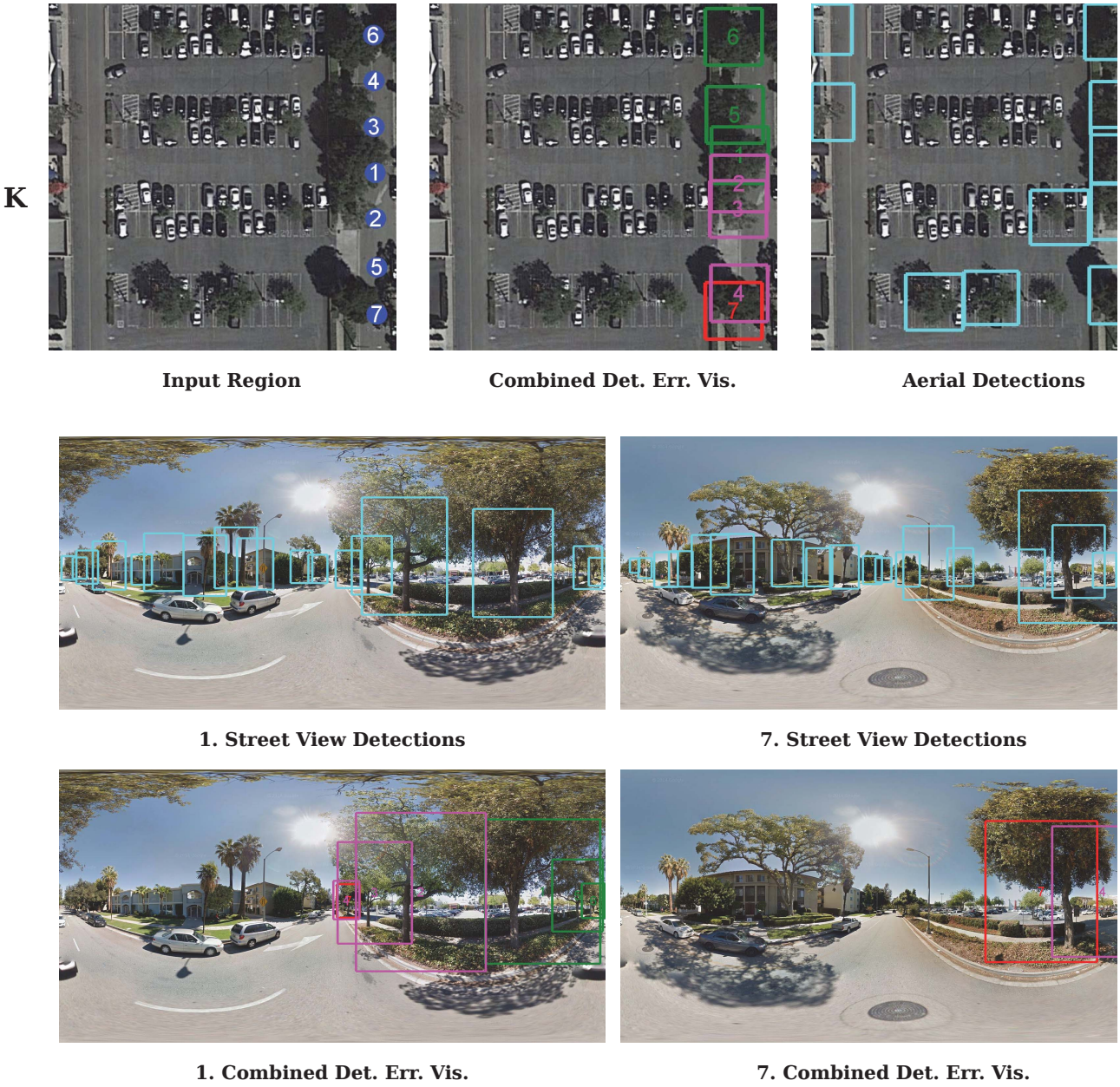
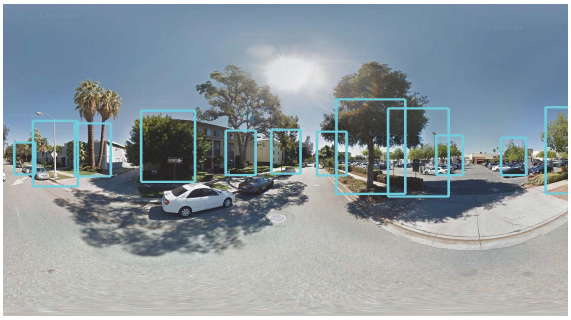


Figure 19. **Example K:** The detector correctly detected 3 trees and correctly rejected at least 12 trees on private land. The detector correctly detected another tree; however, the localization was inaccurate, resulting in a false positive (box 7) and false negative (box 4). The inaccuracy probably occurred because the recorded camera position and heading of the Google street view camera was noisy, which is a common problem and subject of future research. This noise probably contributed to 2 other negatives (boxes 2,3) due to weak detections with score below the detection threshold, since those trees were correctly detected in the street view images. The misalignment between street view and aerial detections often causes lower combined scores.

L

Input Region **Combined Det. Err. Vis.** **Aerial Detections**

1. Street View Detections



4. Street View Detections

1. Combined Det. Err. Vis.

4. Combined Det. Err. Vis.

Figure 20. **Example L:** The detector correctly detected 5 trees and correctly rejected several trees on private land. The detector correctly detected another tree; however, the localization was inaccurate, resulting in a false positive (box 9) and false negative (box 5). The inaccuracy probably occurred because the recorded camera position and heading of the Google street view camera was noisy, as evidenced by the fact that box 9 is in the correct location in the street view image but not the aerial image. This is a common problem and subject of future research. This noise probably contributed to 2 other negatives (boxes 3,4) due to weak detections with score below the detection threshold, since those trees were correctly detected in the street view images. The misalignment between street view and aerial detections causes lower combined scores.

M

Input Region **Combined Det. Err. Vis.** **Aerial Detections**

5. Street View Detections

7. Street View Detections

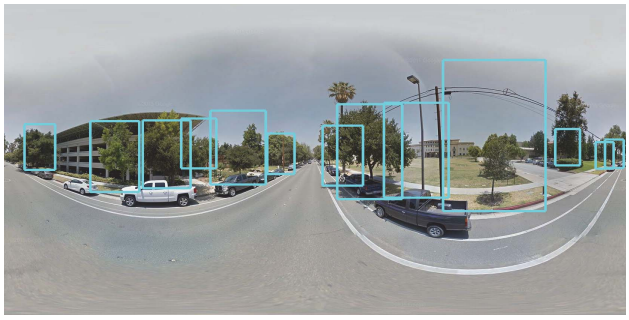
5. Combined Det. Err. Vis.

7. Combined Det. Err. Vis.

Figure 21. **Example M:** The detector correctly detected 13 trees and correctly rejected upwards of 6 trees on private land. 2 false positives occurred due to telephone poles or lamp posts (boxes 14,16) that resemble tree trunks when they are next to foliage of another tree. A 3rd false positive (box 15) occurred due to a duplicate detection where a single tree was detected twice.

N

Input Region **Combined Det. Err. Vis.** **Aerial Detections**



3. Street View Detections

6. Street View Detections

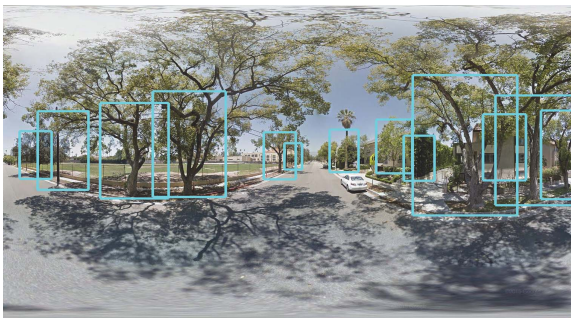
3. Combined Det. Err. Vis.

6. Combined Det. Err. Vis.

Figure 22. **Example N:** The detector correctly detected 5 trees and correctly rejected 5 trees on private land. The detector also correctly detected 2 more trees (boxes 10,11) that were penalized as false positives because they were missing from the test set inventory. They were probably missing because they appear to be recently planted and the inventory was collected in 2013. The detector correctly detected 2 other trees; however, the localization was inaccurate, resulting in false positives (boxes 7,8) with respective false negatives (boxes 6,1). The inaccuracy probably occurred because the recorded camera position and heading of the Google street view camera was noisy, as evidenced by the fact that boxes 7 and 8 are in the correct locations in the street view images but not the aerial image. This is a common problem and subject of future research.

O

Input Region **Combined Det. Err. Vis.** **Aerial Detections**



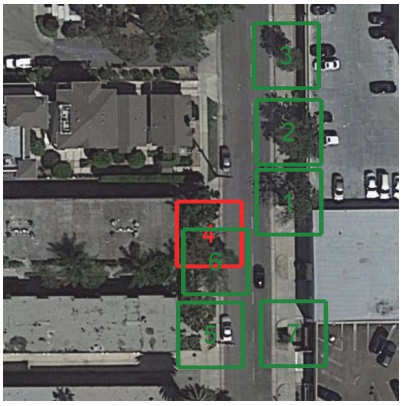
2. Street View Detections **3. Street View Detections**

2. Combined Det. Err. Vis. **3. Combined Det. Err. Vis.**

Figure 23. **Example O:** The detector correctly detected 8 trees. A false negative penalty (box 5) is absorbed; however, this appears to be an error in the ground truth test set, as it occurs in the middle of a driveway. An additional detection (box 9) was penalized as a false positive; however it appears to be a valid tree that was missing from the inventory. Another false positive penalty (box 11) was absorbed due to a detection of an tree on private land (the test set only includes trees on public land). One last false positive (box 12) occurred due to a duplicate detection of a single tree, which sometimes happens for very large trees with a lot of foliage and branching.

P

Input Region



Combined Det. Err. Vis.

Aerial Detections

1. Street View Detections

2. Street View Detections

1. Combined Det. Err. Vis.

2. Combined Det. Err. Vis.

Figure 24. **Example P:** The detector correctly detected 6 trees and correctly rejected many trees on private land. A single false positive occurred due to a lamp post that was in front of foliage of a nearby tree.