Ortho-Image Analysis for Producing Lane-Level Highway Maps

Young-Woo Seo
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
ywseo@ri.cmu.edu

Chris Urmson
Google
1600 Amphitheatre Parkway
Mountain View
CA 94043
curmson@google.com

David Wettergreen
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
dsw@ri.cmu.edu

ABSTRACT

Highway driving can be more safe and reliable when maps contain lane-level detailed cartographic information. Such maps are a resource for driving-assistance systems, enabling them to provide human drivers with precise lane-by-lane advice.

This paper proposes new aerial image analysis algorithms that, from highway ortho-images, produce lane-level detailed maps. We analyze screenshots of road vectors to obtain the relevant spatial and photometric patterns of road image-regions. We then refine the obtained patterns to generate hypotheses about the true road-lanes. A road-lane hypothesis, since it explains only a part of the true road-lane, is then linked to other hypotheses to completely delineate boundaries of the true road-lanes. Finally, some of the refined image cues about the underlying road network are used to guide a hypothesis-linking process.

We tested the accuracy and robustness of our algorithms with high-resolution, inter-city highway ortho-images. Experimental results show promise in producing lane-level detailed highway maps from ortho-image analysis – 89% of the true road-lane boundary pixels were successfully detected and 337 out of 417 true road-lanes were correctly recovered.

1. INTRODUCTION

Maps are important for human navigation. Given a route, a route-guidance system consults maps to provide human drivers with turn-by-turn directions to their destinations. Such guidance helps us safely drive through familiar and even entirely foreign terrains. Route-guidance systems work well because those systems rely on exceptional human perception capabilities. In particular, when a route's direction is given, a human driver steers his vehicle along a particular road-lane while taking note of geometric shapes of the roads, the posted rules, and road-lane boundaries.

However, such route guidance can be very confusing when, due to a lack of information, a guidance system does not detail the actual road geometries. For example, a person driving in the far left lane of a four-lane highway will not be able to exit a ramp on the right immediately after being advised to do so. If the cartographic database serving the guidance system has lane-level detailed information, the route-guidance will surely be more reliable.

This paper proposes new aerial image analysis algorithms that, from highway ortho-images, produce a map of road-lanes that appear on a given highway ortho-image. A road-lane (or lane), in this paper, refers to the part of a road built for controlling and guiding a single line of vehicles. The output of this procedure is cartographic information about road-lanes in a set of pixel coordinates of road-lanes' centerlines and lateral road-widths. Such lane-level detailed highway maps with traffic rules and accurate coordinates can be prepared in advance to facilitate the guiding of highway driving.

To extract such lane-level detailed information from a given aerial image, pixels along road-lane boundaries must be visually and computationally accessible. To meet this requirement, we choose ortho-images with 15-centimeter ground resolution in which lane boundaries can be observed by the naked eye and can potentially be processed computationally. Because the normal longitudinal pavement markings on highways are 4-12 inches wide (10.16-30.48 centimeters) [24], there is at least one pixel for laterally delineating a part of lane-markings. Highways appearing in our target images are inter-city (or arterial) highways built for facilitating transportation between cities [22].

Since our target images are depicted in high-resolution, such image objects as lane-markings and road image-regions contain significant variations in their appearances, such that an object appears differently based on the condition of an image acquisition process and road surface materials. For example, even in a given arterial highway image, road surfaces may be covered with different materials, such as asphalt or concrete. Such variation in road surfaces cause an inconsistency in the color and texture of lane-markings and road-regions. Another example of appearance variation is occlusions caused by man-made structures such as buildings, over-hanging traffic signs, as well as overpasses and their shadows. These structures make parts of roads partially or completely unobservable. The geometry of arterial highways also makes it difficult to delineate a lane's boundary. Ramps with circular paths have high curvatures that require a boundary-following process that tracks non-linear paths. Road-lane junctions near an overpass require extra care due to the complex traffic directions. Road-boundary tracking must also be carefully done at a bifurcation point, where one splits into two, because one of the multiple track-

ing lanes might disappear.

To effectively tackle these challenges, we develop a hierarchical approach to three tasks: gathering road boundary image cues, generating road-lane hypotheses, and linking the hypotheses. To this end, we first scrutinize input images, to harvest two types of image cues about the underlying roads: road image-regions and the geometry of underlying road-lanes. Knowledge of road image-regions are useful in specifying where to look for road-lane boundaries. To obtain the information of road image-regions, we formulate this image segmentation problem as a binary classification. Another important image cue we collect is the geometry of the underlying roads. To obtain this information, we extract lines and analyze the screenshot image of the road-vector to estimate the legitimate driving direction and to identify relevant road structures, such as overpasses. These collected image cues about road surface and geometry will provide strong evidences of the true road-lanes. In particular, these cues facilitate a road-lane hypothesis generation and guide a linking of these hypotheses to build an accurate map of road-lanes. For the problem of linking road-lane hypotheses, we formulate it as a min-cover problem [25]. We look for a set of hypotheses about the unknown true road-lanes to maximally cover the estimated road image-regions with a minimum sum of costs.

In what follows, we detail the methods of harvesting low-level features, the methods of converting these low-level features into meaningful mid-level features, and the methods of generating road-lane hypotheses and of linking them, so as to generate a map of road-lanes in a given image.

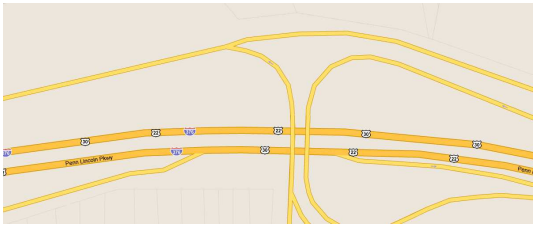
2. RELATED WORK

Guiding humans to their destinations, a map is an essential component for reliable and safe navigation. In the GIS community aerial image analysis has played a crucial role in maintaining existing cartographic databases [1, 2, 3, 6]. Despite being potentially out of date, the geometric relations among spatial objects (e.g., intersection between road segments) appearing on aerial images are invariant over a long period of time, even after natural disasters [20]. To delineate the underlying road network's geometry, structural top-views of interesting areas on aerial images provide better vantage points than those of perspective sensor measurements (e.g., vision sensors and range finders). This provides an alternative way of maintaining existing cartographic databases without virtually going out to the regions of interest. The clear difference between these work and ours is the ground resolution. Most of these work analyzed low-resolution aerial images in which the ground resolution was greater than one meter [1, 2, 3, 7, 8, 11].

We detect interesting road structures, such as intersections and overpasses, to identify potentially complex road geometry. For example, knowing the boundary (or a location) of an overpass is useful in correctly understanding a hierarchical spatial order among road-lanes passing thru the overpass. To recover such 3-dimensional road structures, researchers have directly accessed a road vector or utilized 3D data such as air-borne point clouds [13, 16, 17].

3. GENERATING LANE-LEVEL HIGHWAY MAPS FROM ORTHO-IMAGE ANALYSIS

(a) An input highway ortho-image with 15 centimeters per pixel ground resolution.



(b) A screenshot of the road-vector of the input image.

Figure 1: Two input images for our lane-level map building are shown.

3.1 Harvesting Road-Boundary Image Cues via Bootstrapping

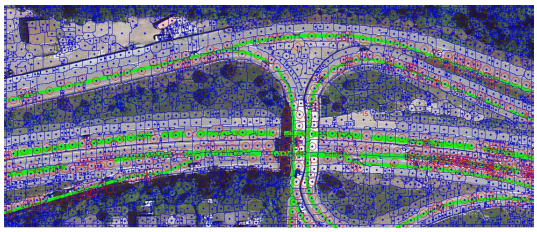
This section explains how to analyze input images to extract low-level image features and how to refine such features to produce task-specific image features that can be directly used to execute other sub-tasks. Our algorithms take two images as the input: highway ortho-image and the input image's road-vector screenshot. Figure 1(a) shows an example of a highway ortho-image and Figure 1(b) shows an example of a road-vector screenshot image.

A road-vector screenshot is a screencapture image that depicts, with distinctive colors, the underlying road-network of a highway scene. When a road-vector image is overlaid with an ortho-image, road-regions in the ortho-image are labeled with real-world cartographic information. One might think that the road-vector screenshot image would trivialize the task of identifying boundaries of road-lanes. Such is not the case. First, the sketches of road-vectors are just parts of images, meaning they do not possess any information about road-vectors, which are directly accessible in a computational form. Second, the road-vector sketches are not entirely overlapped with images of road-regions, resulting in cases where some road-regions remain uncovered. From a pattern recognition perspective, this is a very confusing signal in that some image regions of true road-lanes are marked as "non-road" and vice versa. Thus, when using a screenshot of a road-vector, one should take extra care.

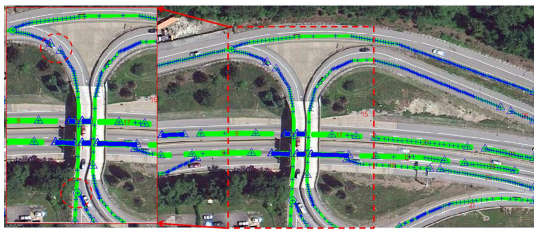
We first parse two input images, to extract low-level features, such as lines and superpixels. As a preprocessing step, we apply a histogram equalization to normalize intensity, a bilateral filter [21] to preserve natural edges, and then compute image gradients. For line extraction, we first run Canny edge detector on image gradients, link those pixels based on

their gradient orientations, and fit a line using eigendecomposition [9, 19]. To obtain a superpixel image, we apply the watershed segmentation algorithm to the gradients of the image to obtain a coarse segmentation. We then reiterate this process until the dimensions of individual superpixels are large enough [10]. We terminate this iteration when the number of superpixels is less than or equal to the predefined proportion (e.g., 15%) of the initial number of segments produced by the watershed algorithm. Figure 2(a) shows an example of a superpixel image.

To extract the useful geometric information of the underlying roads from a road-vector screenshot, we first extract image regions of road-vector sketches, i.e., yellow or yellowish paintings in Figure 1(b), and produce a binary image. This image contains only those fragments of road-vectors without any map-symbols. We then further analyze each of the road-vector fragments, to obtain their geometric properties, such as extremity and bifurcation points. Figure 2(b) shows a result of such analysis. Each (green) polygon represents road-vector fragments where "+" indicates a ridge point, "+" with a triangle is an extremity point, and "+" with a circle is a bifurcation point.



(a) A superpixel image. The elongated green polygons (or blobs) are fragments of a road-vector screenshot.



(b) An analysis of road-vector fragments is performed to obtain their geometric properties.

Figure 2: An intensive image analysis results in two low-level image features. These figures are best viewed in color.

Since these low-level features contain only basic information about road-lanes appearing on the input image, we need to refine these features, making them more relevant and useful in executing our task of producing road-lane maps. These new features include a segmentation of a road image-region, an estimation of some legitimate driving directions of roads appearing on the input image, a lane-marking detection, and locations of interesting road-structures, such as intersections and overpasses.

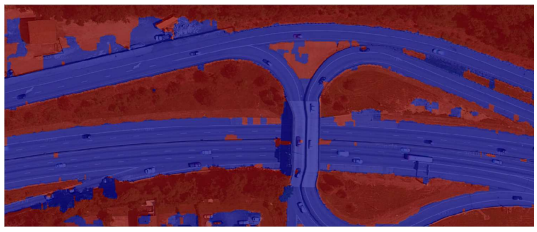
Road-Region Segmentation Having knowledge of road image regions would help one know where to look for road-lane boundaries. Acquiring knowledge of road image-regions is carried out through an image segmentation task that divides an input image into two image sub-regions: road- and non-road-region. We tackle this image segmentation problem as a binary classification problem that takes superpixels as input and assigns each superpixel with one of two class labels: road or non-road. We utilize one of the inputs, a road-vector screenshot image, to prepare, without human intervention, training data. In particular, we treat a superpixel as a positive example if its area is significantly overlapped (e.g., more than 90%) with road-vector paintings; otherwise we treat it as a negative example. Notice that the sketches (or drawings) of road-vectors are not entirely overlapped with image road-regions, resulting in some of road-region superpixels being treated as non-road regions.

To execute the superpixel classification, we first represent each of the superpixels as a feature vector. A feature vector consists of color and texture information. We use a histogram to represent color values in a superpixel and a texton [12] to represent texture values. To handle the initial noisy superpixels' class assignments based on road-vector screenshots, we learn a classifier, a Gaussian Mixture Model (GMM), to probabilistically assign individual superpixels with class labels. To smooth out the potentially inconsistent outputs of the GMM, we run pairwise Markov Random Fields (MRF) and infer the most probable segmentation of the input image using loopy belief propagation. Figure 3(a) shows a result of image road-region segmentation. Results of the road-region segmentation define image regions of interest where all of the remaining tasks for building lane-level highway map have been executed.

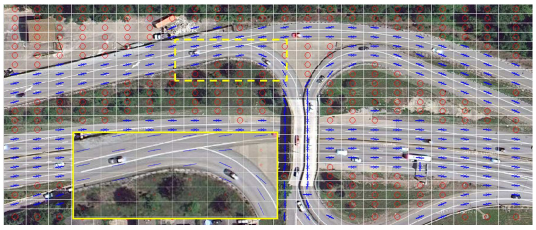
Driving-Direction Estimation The goal of our task is to extract boundaries of individual road-lanes in the given image. This requires tracking boundary pixels of road-lanes that appear in the given image. Thus knowing the driving direction at any given image location is useful for tracking road-lanes boundaries.

To approximate the driving direction from a given image, we use line extraction results that each of the extracted lines partially explain as the contour of roads in a given image. It is undesirable to approximate the driving direction at a pixel level because of all the noise that must be handled. Instead we partition the input image into a number of grid cells. For each grid cell, we identify extracted lines which pass by it and use them to approximate the driving direction of the grid cell. Suppose there are m number of lines identified as passing the ith grid cell. We compute the direction of a grid cell, i, by using the vector sum method, $\hat{\theta}_i = \arctan(y, x)$, where $x = \sum_j^m \cos(\theta_j)$ and $y = \sum_j^m \sin(\theta_j)$, where θ_j is the orientation of the jth line. The orientation of a grid cell is mostly homogeneous to its neighboring cells, particularly in road image regions. To enforce such a constraint, we run an MRF to infer the most probable driving direction of the input image as a whole. Our method of approximating driving direction is motivated by the method proposed in [4] where the authors extract lines from laser-scan data and run an MRF to infer the driving direction homogeneous to a given parking lot image. They use a combination of Canny edge detection and Hough transform to extract lines from a laser scan image. We also tried this combination but found the extracted lines too short to use. Figure 3(b) shows a result of driving-direction estimation.

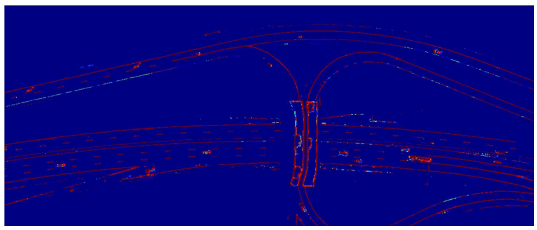
Lane-Marking Detection Lane-markings are one of the most important photometric cues for extracting road-lane



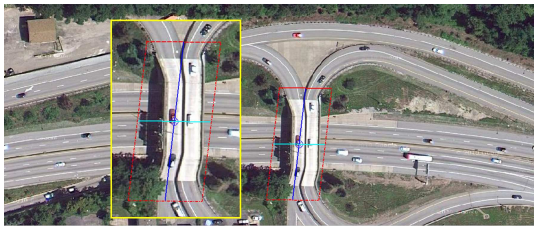
(a) Results of road image region segmentation. The blue regions represent identified road image-regions and the red regions represent non-road image-regions. Although some non-road image-regions are labeled as road, for the most part segmentation results correctly depicted road image-regions.



(b) Results of driving direction estimation. The blue lines indicate the estimated driving direction of a grid cell and the non-drivable regions are depicted by red circles.



(c) Lane-marking detection results. Because the outputs of our lane-marking classification are probabilistic, the results are shown in a heat-image where the color closest to red represents the highest probability.



(d) Results of overpass detection. A red parallelogram represents the boundary of the detected overpass, and two (blue and cyan) lines inside the polygon depict two principal axes.

Figure 3: Four task-specific image features. Viewed best in color.

boundaries. In fact, our goal would be easier met if we had a perfect lane-marking classification on a given ortho-image.

Lane-markings are a type of road-marking that depicts boundaries of road-lanes. On an ortho-image, we can, readily with the naked eye, distinguish lane-markings because they have whitish (or yellowish) colors, relatively higher intensity than their neighboring background pixels. However, these salient features are not always available for image processing because the actual values of lane-marking pixels vary based on image acquisition conditions.

To effectively address the challenge of appearance variation of lane-marking pixels', we formulate the lane-marking detection task as a binary classification problem of discriminating non lane-marking pixels from true lane-marking pixels. To this end, we first downloaded 20 highway ortho-images separated from the images for generating lane-level highway map and sampled 47,640 pixels as a lane-marking classification data. These consisted of 15,204 lane-marking (positive) pixels and 32,436 non lane-marking (negative) pixels. We converted some of these sampled pixels into features, to learn a binary classification model of lane-marking pixels' photometric variations. For the feature representation, we look into the contrast of intensity values between a lane-marking pixel and its neighboring pixels. In fact, to generate a feature vector of the pixel, we use the local binary pattern (LBP) [15] and four different statistics, such as entropy, smoothness, uniformity, and variance of a target pixel and its neighboring pixels. Note that this is the only place we use manually labeled data for training a part of our system.

To find the best one for our lane-marking detection task, we first set aside a portion (about 30%, 14,292) of the labeled pixel data as testing data and used the rest of them to train a classifier. We tried six different classification setups and found that the AdaBoost outperformed all others – AdaBoost with a feature representation without color information produced 0.98 precision and 0.97 recall rates on average. Figure 3(c) shows a result of lane-marking detection.

Interesting Road-Structure Detection To accurately delineate a road-lane's boundaries, it is necessary to recognize road-structures such as overpasses and intersections that may indicate complex road geometries. For example, a presence of an overpass suggests the fact that multiple roads pass each other orthogonally in the same image region.

The input of the overpass detection algorithm is the road-vector screenshot. As described earlier, the road-vector screenshot image is analyzed and converted into a set of road-vector fragments. Each of the road-vector fragments contains the geometric characteristic of parts of the underlying roads. For each of the road-vector fragments, we extend each of the extremity points in the direction of the fragment and identify any intersection with other fragments if their intersection angle is greater than a given threshold (e.g., $\pi/3$). A potential overpass is localized by investigating intersections of these extended lines.

To finalize the overpass detection process, it is necessary to identify the boundary of a localized potential overpass. To this end, we search for any of the closest extracted lines that intersect with any of the two lines from the overpass localization and are greater than the same threshold used earlier. Figure 3(d) shows the final result of overpass detection. The bounding box of a detected overpass lets other sub-tasks of extracting lane-level highway information know of the existence of an overpass and that the road geometry around this bounding box has more than one direction. Our method of detecting an overpass is much simpler than those relying on 3-dimensional data of road vector databases [16, 17]. In particular, our method require no 3-dimensional geometry-related computation.

We describe how we obtain four mid-level image features. Aside from the lane-marking detection in which we used some human-labeled data to train a lane-marking classifier, we obtain, without human intervention, three other important cues – road-region segmentation, driving direction estimation, and overpass detection.

In the next section we detail how these four mid-level features are used to generate road-segment hypotheses and how to link them to build lane-level detailed highway maps.

4. ROAD-LANE HYPOTHESES

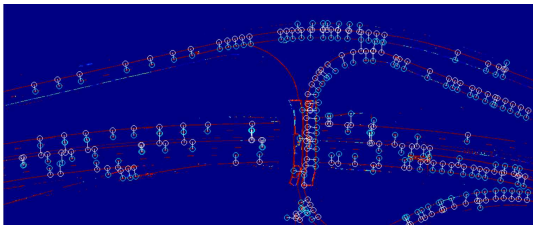
The previous steps of extracting image cues about the true road-lanes provide us a better understanding of road-lanes appearing in the input image. In particular, we know which image sub-regions are most probably road-regions, which pixels within the road-regions are likely parts of lane-markings, how the roads are laid out, and where overpass structures occur. Based on this understanding, we are generating road-lane hypotheses and linking them, in order to delineate road-lane boundaries.

4.1 Generating Hypotheses about the True Road-Lanes

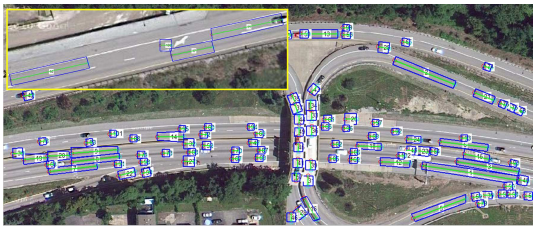
A road-lane is modeled by a piecewise linear curve that consists of multiple control points along the centerlines of the road-lane and their properties, such as lateral width and orientation. Thus, generating a hypothesis about a true road-lane would be equivalent to identifying these (control) points' locations. However, given that the boundary location of road-lanes are unknown, it is difficult to localize the centerlines of road-lanes. Instead we investigate pixels of lane-marking detection results (or lane-marking pixels). No lane-marking pixels along the true centerlines of road-lanes are available, but one can interpolate the centerline locations from a set of regularly-spaced, lane-marking pixels.

A true lane-marking pixel has many neighboring lane-marking pixels regularly-spaced longitudinally and laterally (or orthogonal to the longitudinal direction). Because two true lane-markings located at each other's side can be used to accurately measure the width of a road-lane at that location, we are looking for lane-marking pixels that have strong supportive (or neighboring) patterns in longitudinal and lateral directions of the roads.

While searching for lane-marking pixel candidates, we can also utilize our prior knowledge of the legitimate road-width of a normal highway. Given that we know the minimum width of a highway lane (i.e., 12 feet) in the U.S.[23] and the ground resolution of our test image, we can remove any pairs of lane-marking pixels that have lateral support (i.e., neighboring lane-marking pixels found at orthogonally to the estimated driving direction) shorter than 24 pixels (24 $\times$ 15cm = 3.75meters) or longer than any maximum values. However, care must be taken before incorporating such prior knowledge because road-widths vary – on arterial highway images, some of the road-lanes may have wider or narrower lateral distances. We empirically found that 22 to 35 pixels worked best for the variation in road-widths.



(a) Resulting road-width cues. A dumbbell-like symbol is a road-width cue where the two circles at the ends of a line indicate lane-marking pixel locations with strong longitudinal and lateral neighboring lane-markings.



(b) The road-lane hypothesis generation process produces 99 hypotheses about the 10 true road-lanes.

Figure 4: Results of road-lane hypothesis generation process. Viewed best in color.

From the road-region segmentation results and lane-marking detection results, we already have a good sense of which image sub-regions are likely to be parts of roads and which part of estimated road-regions are probably lane-markings. To make the search of these road-width cues efficient, we begin with superpixels that belong to the segmented road-regions. For each superpixel, we investigate whether each of the lane-marking pixels has a sufficient number of neighboring lane-marking pixels in longitudinal and lateral directions on the roads. Any lane-markings with more than the predefined threshold remain in the candidate list for generating road-lane hypotheses. Figure 4(a) shows the results of a road-width cue search. A road-width hypothesis is represented by a numeral width value and two lane-marking pixels.

After we find a set of road-width cues, the next step is to generate a set of road-lane hypotheses. This process is executed in a similar manner to that of the road-width cue search. For each road-width cue (or road-width hypothesis), we draw two lines, longitudinally, from the center of the two lane-marking locations and group together any road-width cues within extending line segments. This forms a road-lane hypothesis. The longitudinal direction corresponds to

the driving direction estimated earlier from extracted lines. This search results in grouping the neighboring road-width cues around the input road-width hypothesis. Figure 4(b) shows a set of resulting road-lane hypotheses.

4.2 Linking Road-Lane Hypotheses for Delineating Road Boundary

By searching for road-width cues and linking the identified cues, we generate a set of road-lane hypotheses. To extract boundary lines of true road-lanes, we need to link road-lane hypotheses. We formulate the problem of linking hypotheses as a min-cover problem in which we search for a set of road-lane hypotheses to maximally cover the estimated road regions with a minimum sum of costs. In particular, we are looking for a new set of road-lane hypotheses, $X = \{L_1, ..., L_k\}$, which link the generated road-lane hypotheses based on the previously obtained local evidences of the unknown true road-lanes with a minimum sum of linking costs. While linking road-lane hypotheses, the new set of road-lane hypotheses should maximally cover the estimated road image-regions.

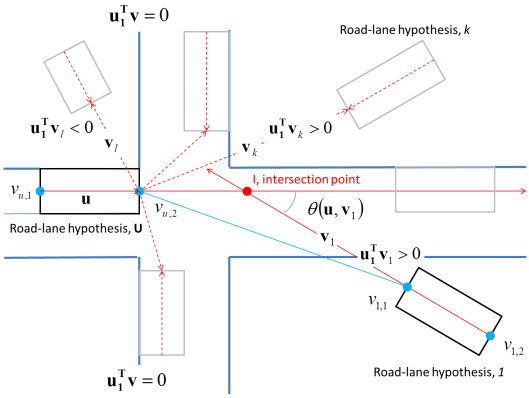
$$X^* = \arg\min_X Cost(X)$$
$$Cost(X) = \sum_{L_i \in X} C(L_i)$$

where $C(L_i)$ is the cost of linking any two road-lane hypotheses, H_r and H_s. Our formulation is motivated by two previous studies [5], [27]. For our case, we generate a set of hypotheses about unknown true road-lanes to cover approximated true road image regions. The previous studies generated hypotheses to delineate object contours [5] and to cover road regions in a LIDAR intensity image [27].

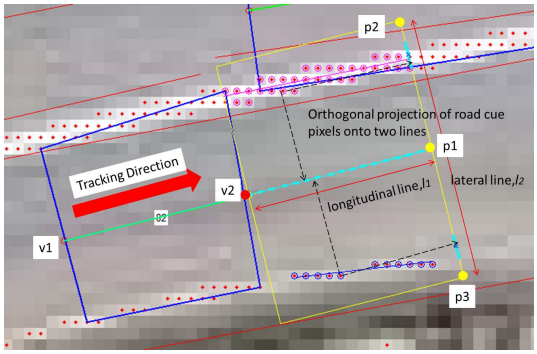
To find approximate solutions to these cost functions, we devise two linking functions. The first linking function considers a potential connection between any two hypotheses purely following geometric constraints. And the second function investigates any photometric constraints of a potential link.

While implementing the first linking function, we refer to the geometry of actual highways where the geometric shape of the road is highly correlated with its speed limits. In other words, it is easy to observe a low curvature road-shape on highways due to its higher speed limits. Another piece of useful knowledge for linking hypotheses based on geometric constraints is to observe driving direction between two road-lane hypotheses. It is highly unlikely for any two hypotheses to be linked to each other when a path of homogeneous driving direction is absent. Figure 5(a) illustrates an example of geometry-based hypotheses linking, where a road-lane hypothesis, $\mathbf{u}$, is searching for a good candidate hypothesis with which to link. Due to the fact that our target roads are highways, any hypotheses located behind an input hypothesis should be discarded. We compute one-to-many dot-products between an input hypothesis and all remaining hypotheses. We do this to filter out any hypotheses located behind the input hypothesis. In the example shown in Figure 5(a), the hypothesis, $\mathbf{v}_l$, is removed from the candidate list, $\mathbf{v}_1$ and $\mathbf{v}_k$, remain in the candidate list for further consideration. For each of the hypotheses in the candidate list, we compute the value of geometric linking potential based on three geometric properties: curvature, intersection angle, and Euclidean distance.

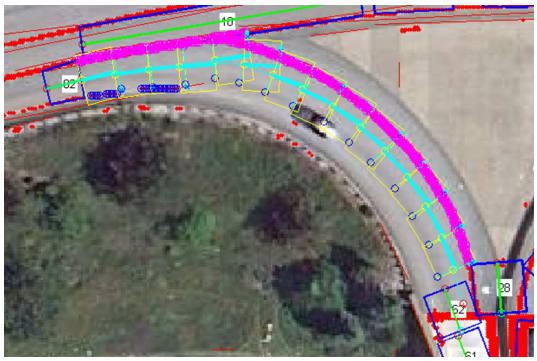
The second linking function investigates whether sufficient image photometric cues are present (i.e., lane-marking pix-



(a) The collected image cues and prior information about arterial highways imposes geometric constraints on finding a potential linking of any two road-lane hypotheses.



(b) The obtained photometric image cues provide strong evidence of potential links among the generated road-lane hypotheses. A tracking of photometric cues among any potential, geometrically plausible, links is conducted before actual linkings are established. Red dots represent lane-marking pixels and a blue rectangle represents a road-lane hypothesis.



(c) The result of photometric cue tracking illustrated in Figure 5(b). The road-lane hypothesis labeled 82 was successfully, through a high-curvature path, linked to another road-lane hypothesis labeled 62.

Figure 5: These figures illustrate how two linking functions find the best potential links among road-lane hypotheses. Viewed best in color.

els) on a potential path linking any two hypotheses. We assume that an optimal link always exists between two consecutive control points that maximizes photometric constraints around the link. The second linking function thus searches for the locally optimal link between two vertices along a potential path between two road-lane hypotheses. A connection between all these locally optimal links would result in an optimal approximation of a true road-lane. The incremental examination of consecutive links will be terminated when the next potential move intersects with either another road-lane hypothesis or one of the image bounds. While tracking the locally optimal path, the direction of tracking is initially set to the direction of the hypothesis, but after the initial step, the direction is adjusted by looking at the estimated driving direction.

Figure 5(b) illustrates such tracking of road-lane boundary cues. In this example, the tracking is about to begin at the vertex ($v2$) of a road-lane hypothesis and search for a locally optimal link for the next point. Currently, it examines one of the possible links to a point, $p1$, within the yellow rectangle where lane-marking pixels on the left side of the tracking direction are marked with magenta circles and lane-marking pixels on the right side are marked with blue circles. We use two line segments to collect road boundary cues: a longitudinal line, $l_1 = p1 - v2$ and a lateral line, $l_2 = p2 - p3$. We first project all lane-marking (magenta and blue) pixels onto these two lines. Let us denote $p(b)$ as a projected point of a lane-marking pixel p. The projected point $p(b)$ on line l_1, for example, can be expressed as $p(b) = v2 + b(p1 - v2)$, where $b = \frac{(P-v2)^T(p1-v2)}{(p1-v2)^T(p1-v2)}$. $p(b)$ is projected on the line segment l_1 if it satisfies $b \in [0, 1]$. Using these projected points, the second linking function evaluates the quality of a potential link to the next control point (e.g., a line segment between $v2$ and $p1$). In general, the optimal link has a wide spread projection on the longitudinal line and a narrow spread projection on the lateral line.

Figure 5(c) shows the result of a photometric road-lane boundary cue tracking. This linking function based on tracking is similar to work that traces road image cues to extract road-networks from low-resolution aerial images. In particular, Zhou et al. used for their road cue tracking an extended Kalman filter [26] and Movaghati and his colleagues utilized an unscented Kalman filter [14]. The primary difference is the ground resolution of testing images. Most of the variations in object appearances, imperative to analyzing high-resolution ortho-images, fail to appear in low-resolution aerial images.

In summary, the linking function based on local geometric constraints searches for the potential links that maximally satisfy geometric cues. The link function based on photometric constraints searches for a potential link that maximally complies with the spatial patterns of the detected lane-marking pixels. The optimal link between two road-lane hypotheses would be one that locally minimizes these two constraint functions. Unlike previous work of the min-cover algorithm applications [5],[27], where their solutions were explicitly searching for a sequence of hypotheses, we look for a set of hypothesis pairs such that their potential, geometrically plausible, links are sequentially traced by photometric image cues to cover road image-regions.

5. EXPERIMENTS

This section details experiments conducted to investigate the robustness of our approach to extracting a lane-level highway map and the accuracy of the resulting maps. In what follows, we first explain the experimental setup and evaluation methods, then show experimental results, and finally discuss the findings.

5.1 Experimental Settings

From Google's map service[1], we collected 50 ortho-images that are sampled from the route between the Squirrel Hill Tunnel to the Pittsburgh International Airport. We also saved road-vector screenshots of the ortho-images and manually drew boundary lines of individual road-lanes in each of the collected images for the ground truth.

Although the number of our testing images might seem insufficient, the images contain a sufficient level of difficulty, which, had we increased the number, we would have been challenged to overcome. For example, on 18 out of 50 images 23 ramps with high curvatures appear. When two lanes merge, one of the tracked lanes must, to produce a correct road geometry, disappear. From 27 images, we observed 39 lane-mergings. Variations in road-surface materials were observed from 33 out of 50 images.[2]

5.2 Experimental Results

In this section we discuss the findings from testing our algorithms. To the best of our knowledge, no prior work or image data was available on extracting road-lane boundaries that we could have used for comparison. Hence, we had to come up with reasonable ways of evaluating our results. We evaluate resulting road-lane boundary delineation in two-ways: accuracy of matching between output and ground truth pixels and counting the number of correctly recovered road-lanes in the final outputs. Matching pixel to pixel aims at investigating the performance of our approach at the micro-level; counting the number of road-lanes aims at revealing the accuracy of the resulting geometries.

To evaluate our results at a pixel-to-pixel level, we utilized the method from evaluating performance of object boundary detection [12]. Similar to [12], we regard the extraction of road-lane boundaries as a classification problem of identifying boundary pixels and of applying the precision-recall metrics using manually labeled road-lane boundaries as ground truth. Precision is manifested in the fraction of outputs that are true positives; recall is the fraction of correct outputs over true positives. Each of the output pixels is evaluated by whether it detects true positive pixels. Once we obtain such correspondence between output pixels and ground truth pixels, computing the precision and recall is straightforward.

While resolving this correspondence problem, we must carefully consider a localization error that accounts for the (Euclidean) distance between an output pixel and a ground truth pixel. Indeed, localization errors are present even in the ground truth images. For resolving the correspondence between output pixels and ground truth pixels, we utilized the Berkeley Segmentation Engine's [3] performance evaluation scripts. These scripts solve, using Goldberg's CSA package, the correspondence problem as a minimum cost bipartite assignment problem. We also used, as a baseline method, BSE's probabilistic boundary detection outputs. BSE was developed to detect generic object boundaries, not

[1] `http://maps.google.com`

[2] The complexity analysis of and the complete experimental results of these 50 test images are available from [18].

[3] The BSE and related information are available at `http://www.cs.berkeley.edu/~fowlkes/BSE/`

road-lane boundaries. In addition, since training BSE with our image data is impossible, it may fall short of being a fair comparison. But since anyone can think of such probabilistic boundary outputs as a starting point of delineating road-lane boundary lines, we compared it with our output. Table 1 presents an averaged performance difference between the two outputs over fifty test images.

	F-measure	Precision	Recall
Ours	0.82	0.77	0.89
BSE's	0.44	0.38	0.54

Table 1: An averaged precision-recall measure of micro-level performance between the two outputs.

In achieving our goal, the performance evaluation by a pixel-to-pixel matching for road-lane boundary extraction outputs might be insufficient. The pixel-to-pixel measure counted a match when an output boundary pixel was located to a true boundary pixel within a predefined distance threshold (e.g., 10 pixels). Therefore a collection of boundary pixels would not necessarily correspond to a road-lane boundary. To be useful, these detected boundary pixels must be interpreted as parts of a road-lane. In other words, the desirable output for our purpose, is one that treats a road-lane as a polygon, bounded by a closed path and image boundaries, where we can estimate lateral road widths, curvature, and other interesting geometric properties along the centerline of a road-lane polygon. To measure such macro-level performance, we first visually inspected our outputs and the input image to resolve the correspondence between the resulting road-lanes and true road-lanes appearing on the input image. We then counted the number of correct and incorrect output road-lanes and missed true road-lanes. If the area of overlap between a road-lane output and a true road-lane was roughly greater than 80%, then we counted it a correct match. This counting resulted in a two-contingency table for the performance of each test image. Table 2 shows a macro-level performance that is obtained by merging individual contingency tables over fifty test images. An averaged performance was then computed by using this table, precision $= 0.792 = \frac{337}{337+88}$, and recall $= 0.771 = \frac{337}{337+100}$, meaning that 79% of the resulting road-lanes were correct and 77% of true road-lanes appearing on the test images were correctly recovered.

		Ground Truth	
		Road-lane	Not road-lane
Output	Road-lane	337	88
	Not road-lane	100	×

Table 2: A contingency table is used to measure the macro-level performance of our highway map generation methods.

Examples of resulting maps are shown in Figure 6. Figure 6(a) shows some of the most accurate results with all of the true road-lanes appearing on test images recovered correctly. While processing these images, our approach successfully tracked high-curvature ramps, correctly connected road-lane boundaries around overpasses, effectively handled variations in road-surface materials, and partial image distortions.

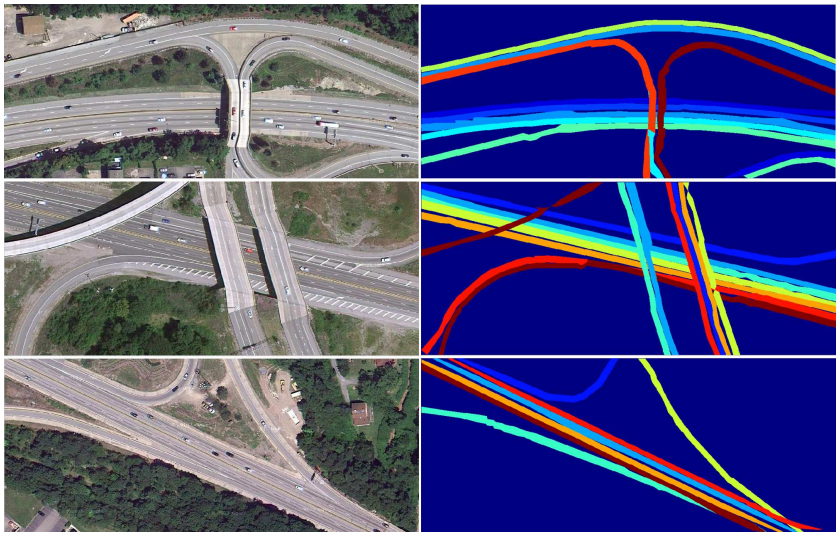
Figure 6(b) shows some reasonable results where most of the true road-lanes were recovered correctly. However, not all the true road-lanes have been recovered and some of the geometry of the resulting road-lanes is incorrect. Our approach was unable to correctly produce road-lane maps from the testing images in Figure 6(b) because these images contained more challenging image characteristics. For example, the overpass in the first row was successfully detected. But, the underestimated boundary of the detected overpass resulted in inaccurate linkages of road-lanes at the edge of the overpass. For the examples in the second row, there was a false positive around the ramp. This happened because our method identified the road-shoulder image-regions as a road-lane. In the testing image in the third row, the shadow of the overpass covers most of road-lanes located to the left of the overpass. Even with a successful detection of the overpass, due to a relatively high curvature, our approach failed to correctly identify the direction of road-lanes.

Figure 6(c) shows near-failure cases where some of the true road-lanes are not recovered and where some of the true road-lanes are incorrect. The test image shown in the first row posed the most significant challenge in our test image collection. The road-lanes appearing on the left of the image are significantly distorted and a cascade of overpasses makes it even harder to analyze. Although our approach recovered some parts of the true road-lanes, most of them were inaccurate and the linkages among them were incorrectly determined. In the second example, our approach failed to link road-lane hypotheses due to the presence of the bridge's suspension span and was unable to complete the linkage of road-lanes near the overpass at the bridge-entering region. Testing images shown in the third showed complicated road geometries. Image distortions appearing on overpasses made it even harder to track road-boundary image cues. Our lane-marking detector failed to detect road-boundary cues from the road surface of the overpass in the last example and was unable to correctly delineate road-lane boundaries, resulting in incorrect linkages of road-lane hypotheses around the overpass.

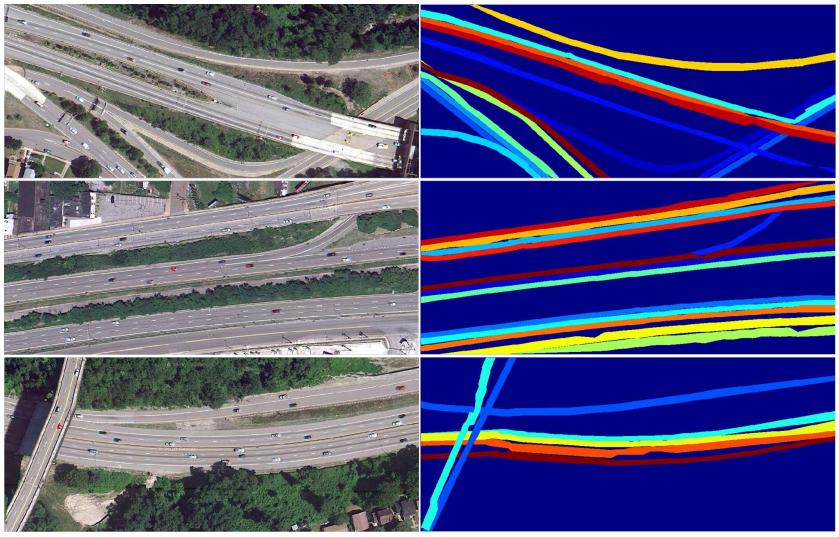
6. CONCLUSION

This paper presented a new approach to extracting lane-level detailed highway maps from a given ortho-image. We chose high-resolution, inter-city highway ortho-images as target images because pixels along road-lane boundaries must be visually and computationally accessible. To effectively address photometric variations in interesting object appearances, we developed a hierarchical approach to three tasks: to collecting road boundary image cues via bootstrapping, to generating hypotheses about the unknown true road-lanes, and to linking hypotheses with respect to the photometric and geometric constraints imposed by the collected image cues and prior information. To minimize human intervention for analyzing given ortho-images, we analyzed screenshots of road vectors to obtain the relevant spatial and photometric patterns of road image-regions. We then refined the obtained patterns to generate hypotheses about the true road-lanes. A road-lane hypothesis, since it explains only a part of the true road-lane, was then linked to other hypotheses to completely delineate boundaries of the true road-lanes. Finally, some of the refined image cues about the underlying road network were used to guide a hypothesis-linking process.

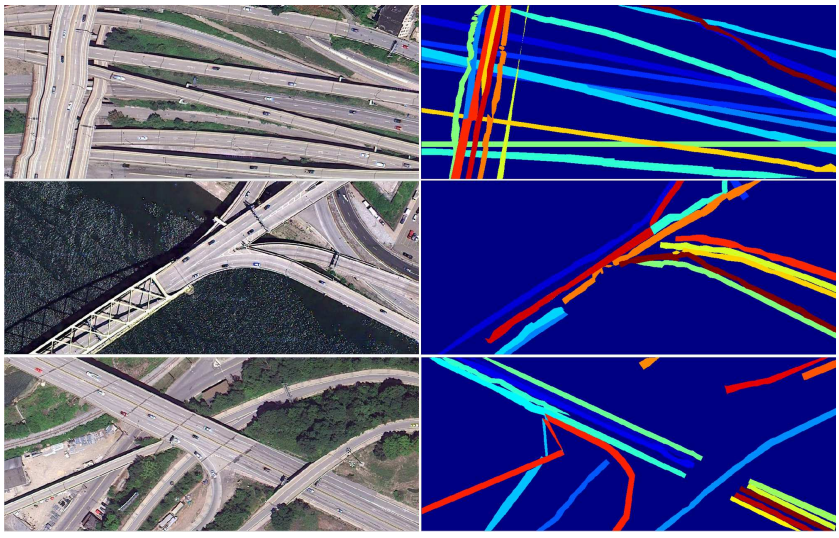
We tested our algorithms with 50 challenging arterial highway images. The results were evaluated according to two aspects: pixel-to-pixel matching and counting correct and incorrect outputs. Our approach demonstrated promising results in that, overall, 79% of the resulting road-lanes were



(a) Highly accurate results of highway map generation.



(b) Near-perfect results of highway map generation.



(c) Near-failure case of highway map generation.

Figure 6: Some of resulting maps. There are two sub-figures in each row. The figure on the left is a test image and the figure on the right is our output, where each road-lane output is depicted in a different color and the background is depicted in blue. These figures are best viewed in color.

correct and 77% of true road-lanes appearing on the test images were correctly recovered.

Although we believe our test images pose sufficient challenges for the task of producing lane-level detailed highway maps, for future work, we would like to test our algorithms with more challenging aerial images.

7. REFERENCES

[1] E. Baltsavias and C. Zhang. Automated updating of road databases from aerial imagery. *International Journal of Applied Earth Observation and Geoinformation*, 6:199–213, 2005.

[2] C.-C. Chen, C. A. Knoblock, and C. Shahabi. Automatically conflating road vector data with orthoimagery. *GeoInformation*, 10:495–530, 2006.

[3] Y.-Y. Chiang and C. A. Knoblock. Automatic extraction of road intersection position, connectivity and orientations from raster maps. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2008.

[4] D. Dolgov and S. Thrun. Detection of principal directions in unknown environments for autonomous navigation. In *Proceedings of the Robotics: Science and Systems*, 2008.

[5] P. Felzenszwalb and D. McAllester. A min-cover approach for finding salient curves. In *Proceedings of IEEE Workshop on Perceptual Organization in Computer Vision*, 2006.

[6] M. flavie Auclair Fortier, D. Ziou, C. Armenakis, and S. Wang. Automated updating of road information from aerial images. In *Proceedings of American Society Photogrammetry and Remote Sensing*, pages 16–23, 2000.

[7] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *Proceedings of European Conference on Computer Vision*, 2008.

[8] J. Hu, A. Razdan, J. C. Femiani, M. Cui, and P. Wonka. Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4144–4157, 2007.

[9] P. Kahn, L. Kitchen, and E. Riseman. A fast line finder for vision-guided robot navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11):1098–1102, 1990.

[10] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *Proceedings of European Conference on Computer Vision*, pages 322–335, 2010.

[11] F. Leberl, H. Bischof, H. Grabner, and S. Kluckner. Recognizing cars in aerial imagery to improve orthophotos. In *Proceedings of International Symposiums on Advances in Geographic Information Systems*, 2007.

[12] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1), 2004.

[13] P. Meixner, F. Leberl, and M. Bredif. Planar roof surface segmentation using 3d vision. In *Proceedings of ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 9–15, 2011.

[14] S. Movaghati and A. Moghaddamjoo. Using unscented kalman filter for road tracing from satellite images. In *Proceedings of Asia International Conference on Modelling and Simulation*, pages 379–384, 2008.

[15] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[16] C. Qian, B. Gale, and J. Bach. Earth documentation: Overpass detection using mobile lidar. In *Proceedings of IEEE International Conference on Image Processing*, pages 3901–3904, 2010.

[17] J. Schpok. Geometric overpass extraction from vector road data and dsms. In *Proceedings of ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 3–8, 2011.

[18] Y.-W. Seo. *Augmenting Cartographic Resources and Assessing Roadway State for Vehicle Navigation*. PhD thesis, The Robotics Institute, Carnegie Mellon University, April 2012.

[19] Y.-W. Seo and C. Urmson. A hierarchical image analysis for extracting parking lot structures from aerial image. Technical Report CMU-RI-TR-09-03, Robotics Institute, Carnegie Mellon Univ., 2009.

[20] B. Soleimani, M.-H. Z. Ashtiani, B. H. Soleimani, and H. Moradi. A disaster invariant feature for localization. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1096–1101, 2010.

[21] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of International Conference on Computer Vision*, pages 839–846, 1998.

[22] F. H. A. U.S. Department of Transportation. *FHWA Functional Classification Guidelines*, 2000. http://www.fhwa.dot.gov/planning/fctoc.htm.

[23] F. H. A. U.S. Department of Transportation. *Mitigation Strategies for Design Exceptions*, 2007. http://safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/

[24] F. H. A. U.S. Department of Transportation. *Manual on uniform traffic control devices for streets and highways*, 2009. http://mutcd.fhwa.dot.gov/.

[25] V. V. Vazirani. *Approximation Algorithms*. Springer, 2004.

[26] J. Zhou, W. F. Bischof, and T. Caelli. Road tracking in aerial images based on human-computer interaction and bayesian filtering. *Journal of Photogrammetry and Remote Sensing*, 61(2), 2006.

[27] Q. Zhu and P. Mordohai. A minimum cover approach for extracting the road network from airborne lidar data. In *Proceedings of ICCV Workshop on 3D Digital Imaging and Modeling*, pages 1582–1589, 2009.