Exploiting Publicly Available Cartographic Resources for Aerial Image Analysis

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

Cartographic databases can be kept up to date through aerial image analysis. Such analysis is optimized when one knows what parts of an aerial image are roads and when one knows locations of complex road structures, such as overpasses and intersections. This paper proposes self-supervised computer vision algorithms that analyze a publicly available cartographic resource (i.e., screenshots of road vectors) to, without human intervention, identify road image-regions and detects overpasses.

Our algorithm segments a given input image into two parts: road- and non-road image regions. It does so not by learning a global appearance model of roads from hand-labeled data, but rather by approximating a locally consistent model of the roads' appearance from self-obtained data. In particular, the learned local model is used to execute a binary classification. We then apply an MRF to smooth potentially inconsistent binary classification outputs.

To detect overpasses, our method scrutinizes screenshots of road vector images to approximate the geometry of the underlying road vector and use the estimated geometry to localize overpasses.

Our methods, based on experiments using inter-city highway ortho-images, show promising results. Segmentation results showed on average over 90% recall; overpass detection results showed 94% accuracy.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Segmentation—*Partitioning*; I.4.8 [**Scene Analysis**]: [Objec Recognition]

General Terms

Algorithms

Keywords

Ortho-Image Analysis, Road Image Region Segmentation, Overpass Detection

ACM SIGSPATIAL GIS'12, November 6-9, 2012. Redondo Beach, CA, USA, Copyright (c) 2012 ACM ISBN 978-1-4503-1691-0/12/11...$15.00
.

1. INTRODUCTION

In the GIS community, an essential part of maintaining existing cartographic databases is extracting relevant objects and spatial patterns from aerial images [1, 3, 4, 6, 9]. Despite being potentially out of date, the topological relations among spatial objects (e.g., two road segments are connected through an intersection) appearing on aerial images do not vary over a long period of time, even after natural disasters [25]. Overhead views of interesting areas from aerial images provide better vantage points, for delineating the geometry of the underlying road network, than those of sensor measurements, from perspective vision sensors, or range finders installed on a ground survey vehicle. In addition, maintaining road-vectors through aerial image analysis is more attractive and cheaper than doing so with manual surveys because aerial images with high resolution have become publicly available.[1]

An important goal of aerial image analysis is to identify image sub-regions where vehicles can virtually drive on. This is because knowledge of road image regions enables aerial image analysis tasks to scrutinize only sub-regions of an image, instead of searching for relevant objects over the entire image. For example, by defining where to look, the information about road image regions facilitates the aerial image analysis tasks of localizing intersections [4] and overpasses [21, 22], of detecting vehicles [8, 13, 11], of summarizing the geometry of a road network [7, 24], of improving the road network geometry extraction using in-vehicle data such as GPS signal logs or wheel-turning-speed [2, 26].

Knowing what parts of an aerial image are roads is obviously helpful in executing various aerial image analysis tasks. However, it is challenging to precisely identify road image-regions appearing on aerial images because their appearances vary due to the different conditions of image acquisition processes and road surface materials. For example, the line of sight between an acquisition vehicle and the ground can perceptually and computationally change the photometric characteristics of road image-regions. Even in a given aerial image, road surfaces may be covered with different materials, such as asphalt or concrete. Such variation in road surfaces cause an inconsistency in color and texture of road images-regions.

The most dominant approach to recognizing spatial objects appearing on aerial images is to learn the global photometric and geometric properties of interesting objects from hand-labeled images and then to apply the learned patterns to images unseen during the training [7, 8, 11, 12,

[1]Aerial imagery with a foot resolution about US and its territories are publicly available from United States Geological Survey (USGS), `http://www.usgs.org`

17]. Despite impressive results from such methods trained with manually labeled data and some inventions of making manual data acquisition cheap, manual annotations are still labor-intensive and error-prone.

In this paper, instead of relying on hand-labeled data, we exploit a publicly available cartographic resource to learn, without human intervention, an appearance model of road image-regions. Particularly, we make use of screenshots of existing road-vectors to acquire a local appearance model of road image-regions and use the learned model to identify road image regions. An analysis of road vectors' screenshots also provides us with a geometric understanding of the underlying road network. We use the reconstructed geometry to detect overpasses appearing on aerial images.

Our main contribution is to demonstrate a good use of publicly available cartographic resources to obtain important pieces of cartographic information from aerial image analysis. To this end, we developed self-supervised computer vision algorithms to identify road image-regions and a geometry analysis algorithm to detect overpasses appearing on aerial images.

2. RELATED WORK

By analyzing a screenshot of a road vector, our algorithms automatically learn an appearance model of road image-regions and obtain a geometric model of the underlying road network for identifying road image-regions and for detecting overpasses. In this regard, we first review some of previous work on aerial image segmentation and overpass detection using road vector information. We also investigate some of the self-supervised learning methods for computer vision tasks that minimize the amount of manual effort for training classifiers.

Our method partitions a given aerial image into two groups: road and non-road regions. This method builds upon previous superpixel based image segmentation methods [11, 12]. Representing an image with a set of pixel-groups is attractive for aerial image analysis, because such a representation enables algorithms to treat homogeneous pixels together e.g., pixels belonging to road lane [5, 7, 8, 11, 12, 17]. Such image representations with a set of pixel-groups are also computationally efficient because they reduce the complexity of images from hundreds of thousands of pixels to only a few hundred pixel-groups. We utilize the oversegmentation method introduced in [11, 12] to convert an aerial image into a superpixel image.

Results of aerial imagery segmentation have been extensively utilized to facilitate the GIS-related tasks in achieving their goals: intersection [4] and overpass detection [21, 22], vehicle detection [8, 13, 11], road network geometry understanding [2, 7, 26], and building contour extraction [20]. Our approach is different from these methods in that our methods do not use any hand-labeled data to execute classification-based aerial image segmentation.

We detect interesting road structures, such as intersections and overpasses, to identify potentially complex road geometry. In particular, information about the location and boundary of an overpass sheds light on the hierarchical order of road-lanes passing through the overpass. Researchers have, to recover such 3-dimensional road structures, directly accessed a road vector or utilized 3D data such as air-borne point clouds [21, 22]. By contrast, our method, without using any of these specialized data, detects overpasses by using screenshots of road-vectors. Particularly, we analyze screenshots of road-vectors depicted on map images to approximate a geometry of the underlying road network. The recovered geometry is used to detect overpasses and analyze extracted lines to identify the boundaries of the detected overpasses.

In a common setting, pattern recognizers require hand-labeled data to learn latent patterns embedded in the data. However, manual labeling is expensive and error-prone. To minimize use of hand-labeled data, there have been efforts to utilize a data source, the contents of which are less uncertain, to assign class labels to other data sources [18, 24]. Nair and Clark exploited motion information in video to automatically collect training examples for the task of detecting people from a video of an office corridor scene [18]. Seo and his colleagues analyzed the spatial arrangements between extracted lines to produce a set of nominal parking spot images that are used as training examples for existing machine learning techniques [24]. In this paper, we analyze a screenshot of a road-vector to learn an appearance model of road image-regions and to obtain a geometric skeleton of a road network.

3. EXPLOITING PUBLICLY AVAILABLE CARTOGRAPHIC RESOURCES

3.1 Harvesting Image Cues through Bootstrapping

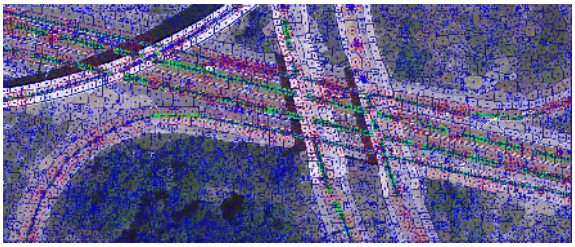
Our algorithms take two images as the input: a highway ortho-image and the input image's road-vector screenshot. Figure 1(a) shows an example of a road-vector screenshot image and Figure 1(b) shows an example of a highway ortho-image with some other information.

A road-vector screenshot is a screencapture image that depicts, with distinctive colors (i.e., yellow), the underlying road-network of the 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 our task of identifying road image-regions. Such is not the case. First, the sketches (or drawings) 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. Some image regions of true road-lanes are marked as "road," while others right next to them are labeled as "non-road." For example, Figure 1(b) shows green blobs about yellowish road-vector paintings appearing on Figure 1(a). These are extracted after identifying yellowish image regions on Figure 1(a), removing map symbols, and overlaid onto a superpixel representation of the input image. Most of the superpixels belonging to the true road regions are not covered by extracted road vector fragments. This holds for the opposite case as well – indicating non-road regions as road regions, e.g., a road-vector painting over trees or buildings. Thus, when a screenshot of a road-vector is used, extra care must be taken.

To handle such a noisy class assignment, we employ probabilistic methods to reduce ambiguity and to minimize inconsistency in classification outputs. We first describe how we obtain low-level image features such as a superpixel image and then explain how we perform an aerial image segmentation based on a binary classification in the following section.

To obtain a superpixel image, we first, as a preprocessing step, applied a histogram equalization to normalize intensity and a bilateral filter [27] to preserve natural edges, and

(a) A screenshot of the road-vector of the input image. This image depicts the geometry of underlying road-network with other cartographic information such as names of highways.



(b) A superpixel image. The elongated green polygons (or blobs) are fragments of a road-vector screenshot. Superpixels' boundaries are delineated by (blue) lines and their centroids are marked with (blue) dots. Red circles indicate superpixels whose areas are significantly overlapped by road-vector fragments.

Figure 1: A screenshot image of road-vector is analyzed to produce two image features: a super-pixel image and fragments of road-vectors. This figure is best viewed in color.

then compute image gradients. We apply the watershed segmentation algorithm to the image of gradient magnitude to obtain a coarse segmentation. We then reiterate this coarse segmentation process until the dimensions (or areas) of individual superpixels are large enough [12]. We terminate this iteration when the current number of superpixels is smaller or equal to the predefined proportion (e.g., 15%) of the initial number of segments produced by the watershed algorithm. Figure 1(b) shows an example of a superpixel image. 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 [10].

3.2 Road Image-Region Segmentation

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- (R) and non-road region (NR). Suppose that $\mathbf{X}$ is an input ortho-image and is represented by n superpixels, $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, where X_i is ith superpixel. A true segmentation function, $f^*(X_i)$, unknown a priori, which assigns a given superpixel, X_i, with the true class label, $Y_i^* \in \{R, NR\}$. Our task is then to learn a segmentation function, $f(X_i)$, which minimizes an empirical error, $\epsilon = \sum_{i=n} [f^*(X_i) \neq f(X_i)]$. In a common setting, one manually assigns a part of superpixels with their class labels to prepare training data and learns a classifier to obtain a function $f(X_i)$ by using the training data [5, 8, 11, 17]. The number of training examples may vary, but can be roughly determined based on the dimensionality of data and the complexity of the problem.

In this paper, we take a different approach to executing this binary classification. Instead of relying on numerous human-labeled examples, we utilize a road-vector screenshot image, to automatically prepare a training data. In particular, we treat a superpixel as a positive example (i.e., a road-region superpixel) if its area is significantly overlapped, e.g., more than 90%, overlapped with road-vector paintings (or fragments); otherwise we treat it as a negative example (i.e., non road-region superpixel). Note that such a weak classification assignment is noisy – some superpixels belonging to true roads are initially labeled as non-road regions and vice versa.

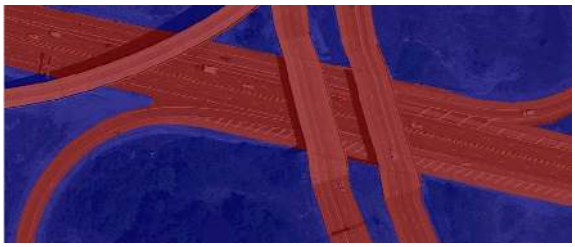
To execute the superpixel segmentation, we first represent each superpixel as a feature vector. A feature vector consists of color and texture information. We use a histogram to represent color values in a superpixel and a dictionary of textons [14, 15] to represent texture value. In particular, for color representation, we used four different representation: RGB, HSV, Lab, and gray intensity. For texture representation, we utilize a dictionary of textons. To produce a texton dictionary, we applied Leung and Malik's filter bank [14][2] to 23 highway ortho-images collected separately from test images, to obtain the initial filter responses and then clustered these filter responses into the predefined number of groups (i.e., 64). A superpixel's texture is represented by a texton if the Euclidean distance between the filter response of the superpixel and the texton is smallest [15].

For each test image, $\mathbf{X}$, we first obtain superpixels and analyze a screenshot of a road-vector to assign superpixels with weak class-labels by thresholding overlapping areas between superpixels and road-vector fragments. This results in $\mathbf{X} = \mathbf{X}_{W+} \cup \mathbf{X}_U$, where $\mathbf{X}_{W+}$ is a set of weakly labeled positive superpixels, and $\mathbf{X}_U$ is a set of unlabeled superpixels. Note that $\mathbf{X}_{W+} \neq \mathbf{X}_R$, where $\mathbf{X}_R$ is a set of true road-region superpixels. Our problem is then to assign an unlabeled (or a weakly-labeled) superpixel with a class label, $f(X_{U,j}) = Y_{U,j} \in \{R, NR\}$. We developed three different approaches to solve this image classification problem: use of area-overlap, superpixel clustering and superpixel classification.

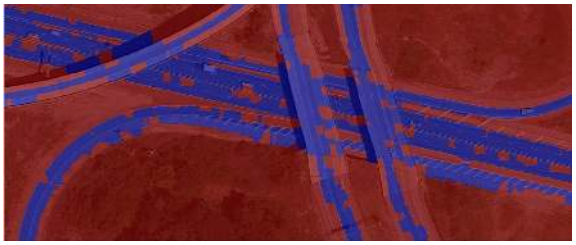
3.2.1 *Use of Area-Overlap for Road Region Segmentation*

An image segmentation based on area-overlap is to use the initial class-label assignments as a segmentation result. Figure 2(b) shows an example of this result. Qualitatively speaking, the outputs mostly cover road image-regions and reveal the skeleton of the underlying road network as well. However, a large part of the true road image-regions are missed, meaning that most of the superpixels belonging to the true road-regions are not recovered. To quantitatively evaluate such a classification output, we compute the area of overlap between segmentation output and the ground truth.

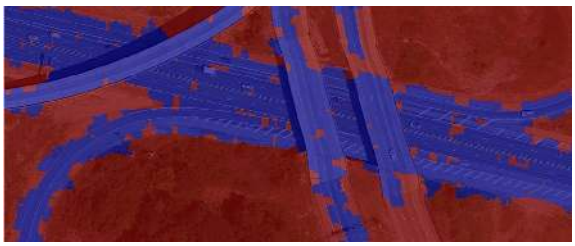
[2]A Matlab implementation of LM filter bank is publicly available from `http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html`



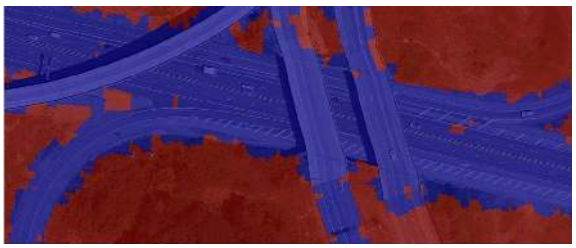
(a) Ground truth segmentation.



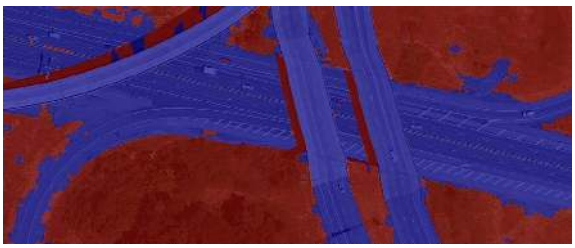
(b) Results by a plain-area-overlap method. ($p = 0.979, r = 0.594, F1 = 0.739$)



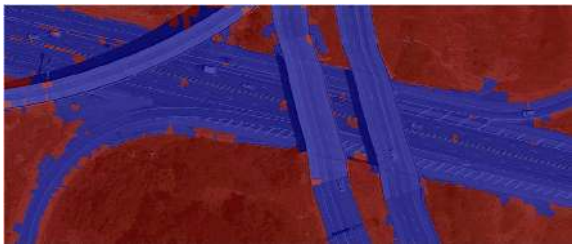
(c) Results of a plain-area-overlap w/ 1st order neighbor expansion. ($p = 0.924, r = 0.830, F1 = 0.874$)



(d) Results of plain area overlap w/ 2nd order neighbor expansion. ($p = 0.817, r = 0.951, F1 = 0.879$)



(e) Results by a spectral clustering. ($p = 0.849, r = 945, F1 = 0.895$)



(f) Results of GMM with MRF. ($p = 0.894, r = 0.969, F1 = 0.930$)

Figure 2: Results of road image region segmentation. The blue regions represent identified road image regions and the red regions represent non-road image-regions. Three numbers in parenthesis show quantitative evaluation of results, i.e., p stands for precision, r for recall, $F1$ for F1 score. Viewed best in color.

Figure 2(a) shows the ground truth of the input image. A segmentation output, r_o, is overlaid with ground truth annotation, r_g, to measure the area of the overlap, $\frac{Area(r_o \cap r_g)}{Area(r_o \cup r_g)}$. The overlapped areas are further analyzed by measuring the following performance matrices: *precision*=$\frac{Area(r_o \cap r_g)}{Area(r_o)}$ and *recall*=$\frac{Area(r_o \cap r_g)}{Area(r_g)}$. In particular, the precision measures how accurate a segmentation output is and the recall measures how much the true road image-regions are correctly recovered. The precision and recall values can be summarized by computing F-score (F1), $F1 = \frac{2 \times precision \times recall}{precision + recall}$.

The outputs of area-overlap method are highly accurate in terms of precision, but it recovers only 60% of the true road image regions. Such under-estimation of the true road image regions can cause problems like excluding relevant spatial objects, such as vehicles, from further investigation.

An obvious way of improving this result is to expand the segmentation output area by including superpixels neighboring the weakly labeled positive superpixels. A first-order neighboring superpixel to an input superpixel is one that directly attaches to a target superpixel, partially sharing its boundaries with the target superpixel. Similarly, a second-order superpixel is the first-order superpixel to the first-order superpixel of the input superpixel. Figure 2(c) and Figure 2(d) show results of expanding the initial area overlap results by including the first and second-order of neighboring superpixels. The recall rate of the simple overlap method is in fact improved as increasing the initial overlapped areas, from 0.839 to 0.963. However, at the same time, the accuracy of the outputs dropped, meaning that the number of incorrectly classified superpixels increased as well.

3.2.2 Spectral Method for Superpixel Clustering

Another intuitive way of obtaining road image-region identification is to group input superpixels into two groups based on their similarities. To this end, we use a spectral clustering that partitions input superpixels using eigenvectors of a superpixel feature matrix [19, 28]. We chose a spectral method for our segmentation problem over conventional iterative clustering methods such as k-means and EM (Expectation and Maximization) for two reasons 1) an iterative method only guarantees a suboptimal partitioning result, which is sensitive to the starting point, and 2) the computational time of these iterative methods to generate an output is linearly proportional to the dimension of the data. By contrast, instead of directly dealing with input feature vectors, a spectral method for clustering is to use an eigenvector (e.g., the second largest eigenvector[3] for our case) of a graph's Laplacian to define an optimal cut of a given data [19, 28]. In particular, we first compute an affinity matrix to measure the similarity between superpixels:

$$\mathbf{W}(i,j) = e^{-d(X_i,X_j)/2\sigma^2}$$

where $d(X_i, X_j) = ||X_i - X_j||^2$ defines a Euclidean distance between two superpixels, X_i and X_j, and σ defines a range of effective affinity between two superpixels. One can think of this affinity matrix as a graph, where each superpixel is a node and two superpixels, if $\mathbf{W}(i,j)$ is greater than zero, are linked over an edge. We derive two more matrices from this affinity matrix: the degree matrix of the affinity matrix, $\mathbf{D}(i,i) = \sum_j W(i,j)$ and the Laplacian of the graph (i.e., the affinity matrix), $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Then we do an eigen decomposition of the graph Laplacian, $\mathbf{L}$, to obtain its eigenvectors, $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{|e|}$, and corresponding eigenvalues, $\lambda_1 > \lambda_2 >, ..., > \lambda_{|e|}$. We treat each component in the second largest eigenvector, $\mathbf{e}_2$, as newly generated data points converted from superpixels and apply k-means to this eigenvector to segment superpixels into two groups. The second largest eigenvector is an indicator function that outputs 1 for a given superpixel belonging to a dominant cluster and zero for otherwise, resulting in superpixels in a newly created one-dimension sample space forming tight clusters based on their initial affinities [28]. Note that clustering an eigenvector is different from clustering superpixels in that each superpixel is represented in a one-dimensional space $X_i \in R^1$ rather than in an m-dimensional original feature space. Figure 2(e) shows the results of spectral clustering. The segmentation results are improved both in accuracy and coverage.

[3]The second largest eigenvector is the eigenvector which the magnitude of its corresponding eigenvalue is second largest.

3.2.3 Superpixel Classification with Markov Random Fields

The last method for road image-region segmentation is a combination of a probabilistic binary classifier and a spatial smoothing method. To minimize the effect of superpixels' incorrect class assignments, we learn a probabilistic classifier, a Gaussian Mixture Model (GMM), and assign individual superpixels with probabilities for being a part of road image regions, $f(X_i) = P(X_i|Y_i)$, where $Y_i \in \{R, NR\}$. Regardless of the initial weak class assignment, we then re-assign superpixels with class labels based on a log likelihood value, $\log\left\{\frac{P(X_i|Y_R)}{P(X_i|Y_{NR})}\right\}$. Since our GMM is only trained by local examples, its outputs may be sub-optimal in terms of classification accuracy. To improve the outputs of the GMM, we run a pairwise Markov Random Fields (MRF) and infer the most probable segmentation of the input image using loopy belief propagation. In particular, we model a superpixel image by a pairwise Markov Random Fields. It is an undirected graphical model that factorizes the underlying joint probability distribution $P(X,Y)$ by a set of pairwise cliques. In such a graphical representation, a superpixel is a node and the first-order neighboring superpixels are considered to be directly linked. A link (or an edge) between superpixels represents dependence (or homogeneity of their class labels) between them. The joint probability distribution is factorized as

$$P(X,Y) = \frac{1}{Z}\prod_{i=1}^{N}\Phi(X_i,Y_i)\prod_{j\in N(i)}\Psi(Y_i,Y_j)$$

where $\Phi(X_i, Y_i)$ is a node potential, $\Psi(Y_i, Y_j)$ is an edge potential, Z is the partition function that ensures a probability density of this model, $N(i)$ is the set of neighboring superpixels. The node potentials are GMM's outputs. The edge potential is computed by Potts model.

$$\Psi(Y_i,Y_j) = \exp\left\{-\beta(Y_i - Y_j)^2\right\}$$

where β is a penalty factor for class label assignment disagreement between nodes. For inferencing the most likely class labels of individual superpixels in a given highway ortho-image, we use loopy belief propagation because it is easy to implement. Figure 2(f) shows the results of this method. This method outperforms all other methods in terms of highest accuracy and largest coverage of the true road image-regions. We further investigate the performance of these methods later and discuss in detail the findings in Section 4.

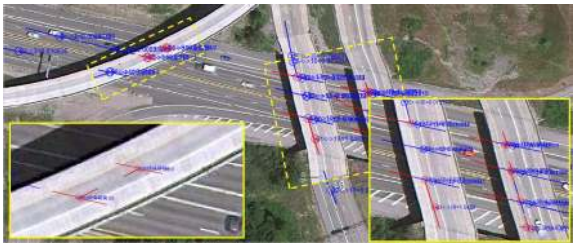
3.3 Overpass Detection

To accurately delineate the geometry of the underlying road network, it is necessary to recognize road structures such as overpasses and intersections, which indicate complex road geometries. For example, the presence of an overpass suggests the fact that multiple roads pass each other in the same image region.

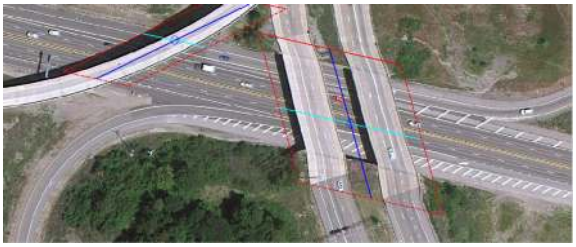
An input of the overpass detection algorithm is the road-vector screenshot. To extract the useful geometric information of the underlying roads from a road-vector screenshot, we first extract image regions of road-vector sketches (e.g., yellow or yellowish paintings in Figure 1(a)) and produce a binary image. This contains only these fragments of road-vector without any map symbols and road markings. We then further analyze each of the road-vector fragments, to obtain their geometric properties, such as extremity and bifurcation (or merging) points. Because a road-vector fragment is an elongated polygon bounded by a closed path, the skeleton of a fragment is useful in acquiring these pieces of information. A skeleton of a polygon is a series of linear lines linking ridge points which are local extrema sitting in the middle of a polygonal shape. We apply a distance transform to each of the road-vector fragments and identify these ridge points. Figure 3(a) 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.

For each of the road-vector fragments, we extend both 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., $\pi/3$) to localize potential overpasses. Figure 3(b) shows a result of overpass localization where a multiple of two (red) intersection lines indicate potential overpass regions. The final process of detecting overpasses is to identify the boundary of a detected 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(c) 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 [21, 22]. Our method require no 3-dimensional geometry-related computation.

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



(b) Results of overpass localization. Every intersection of two red lines indicates a potential location of an overpass. A blue line is a line extended from a road-vector fragment.



(c) 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: These figures show the sequence of overpass detection. Viewed best in color.

4. EXPERIMENTS

This section details experiments conducted to investigate the robustness and the accuracy of our methods of segmenting road image-regions and of detecting overpasses appearing on test ortho-images. In what follows, we first explain the experimental setup and evaluation methods, then show experimental results, and finally discuss the findings.

From Google's map service[4], we collected 50 inter-city highway ortho-images that are sampled from the route between the Squirrel Hill Tunnel to the Pittsburgh International Airport. We also saved corresponding road-vector screenshots of the ortho-images and manually drew boundaries of road image-regions in each of the collected images for the ground truth. While we were manually generating the ground truth images, we ensured that our annotations correctly and reasonably separated drivable image-regions from non-drivable image-regions. Drivable image-regions include road-lanes, their shoulders, parking spaces directly attached

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

to any of road-lanes, and emergency pull-overs. Due to shadows and occlusions, it was often necessary to interpolate or extrapolate the boundaries of drivable regions. We did this when the areas of occlusions and shadows were not significantly large. Figure 2(a) shows an example of ground truth annotation.

We have a list of methods that require optimal parameters for producing desirable results. While extracting lines, we remove any lines the lengths of which are greater than half of the input image width (e.g., 600 pixels), because such long lines usually fail to align with any highway contours. In executing road-region segmentation, we apply Leung-Malik's filter bank [14], a multi-scale and multi-orientation filter bank consisting of 48 filters, and thereby obtain 64 different textons to represent each superpixel. For the spectral clustering, we found the method produced the best results when σ is $\frac{1}{3}$ of the maximum distance in an affinity matrix. For GMM with MRF, we found our road-region segmentation method produced the best results when β is set to 0.2. For overpass detection, we set the angle threshold at $\pi/3$ so as to detect greater intersection angles between road-vector fragments.

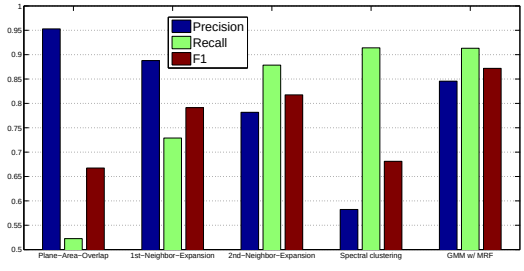


Figure 4: A comparison of the segmentation results. bar graph shows micro-averaged performances.

4.1 Experimental Results

We compared the performance of five methods' segmentation outputs. Although there are many well-defined and complex evaluation methods for comparing performance of multiway-segmentation results [16], to measure the accuracy of our binary segmentation results, we chose an overlap score between the area of a segmentation output and the area of the ground truth. We chose such because it is straightforward to compute and provides an objective comparison among methods.

Figure 4 shows the macro-averaged performance measures of the segmentation outputs. This macro-averaged performance is obtained by computing per image performance measures first, and then averaging the corresponding measures.

The "plain-area-overlap" method outputs highly accurate segmentation results (i.e., its average precision value was greater than 0.95), meaning that, for most of time, the road-vector sketches correctly indicated road image-regions. However, because they do not completely cover the true road image-regions, its recall values are the lowest. This confirms our initial assumption that we cannot directly use a road-vector screenshot to precisely define a road image-region. As we expanded the initial image regions of the road-vector sketches by including neighboring superpixels, its recall value was drastically increased and its coverages of the true road image-regions were increased as well. At the same time, however, their accuracies are inversely proportional to increasing recall rates, indicating that a simple expanding of the initial overlapped image-regions does not correctly recover the true road-image. Over 90% of the true road-image regions are covered by the outputs of "spectral clustering." This is encouraging in that our spectral clustering implementation consumed the least computational time and memory of our five segmentation methods. However, its outputs were not the most desirable. Due to similarities between superpixel features, the spectral clustering grouped vegetation near highway lanes and tops of buildings when road-vector fragments included parts of such objects. This resulted in including some non-drivable image-regions and producing many false positives. By contrast, the accuracies of those methods about area-overlap with neighbor-expansion did not drop drastically because their area expansion were always centered in the initial road-vector sketches. The combination of GMM and MRF (GMM w/ MRF) worked best in terms of the accuracy and coverage of segmentation outputs. Although our GMM learned the appearance model of drivable image-regions from noisy signals, it was capable of identifying most of the drivable region's superpixels. Our MRF implementation did a reasonable job of smoothing spatial label inconsistencies. Similar to those of spectral clustering, GMM misclassified some of the non drivable-regions based on similarities in superpixels' appearances. Some image regions were in fact drivable regions (one could visually identify that their surfaces were covered by the same or similar materials, e.g., asphalts). However, due to the rules of our ground truth annotation, those image regions are not annotated as drivable regions. Figure 5 shows some of the segmentation results by GMM w/ MRF.[5]

Figure 6 shows some of the overpass detection results. Eighteen out of 50 test ortho-images contained a total of 33 overpasses. Our algorithm successfully detected the locations and boundaries of 31 of these. Figure 6(b) shows an example of overpass detection in which our algorithm was able to detect an intersection as well. Figure 6(h) shows an overpass detection result that overestimated the boundary of the overpass. We believe that an overestimated of overpass boundary should be permissible, but an underestimation would be problematic in terms of understanding the geometry of the underlying road network. Figure 6(i) and 6(j) show two failure cases of our overpass detection. In particular, the complex true road geometry and the image distortion in Figure 6(i) made it challenging to correctly identify the boundaries of overpasses. Our method was able to correctly localize overpasses, but failed to precisely detect their boundaries. The boundaries of the detected overpasses in Figure 6(j) are underestimated, resulting in some overpass image-regions were left undiscovered.

5. CONCLUSIONS

This paper presents algorithms that analyze a publicly available cartographic resource, i.e., screenshots of a road vector, to identify road image regions and detect overpasses. From experiments using inter-city highway ortho-images, our method showed a promising result: Segmentation results showed on average over 90% recall; overpass detection results showed 94% accuracy.

For future work, we would like to investigate whether it is possible to generalize a global appearance model of road regions from those locally learned models. Although we believe our test data poses sufficient challenges for road image

[5]The complete experimental results of these 50 test images are available from [23].

region segmentation and overpass detection, we would like to test our algorithms with more aerial images.

6. ACKNOWLEDGMENTS

This work is funded by the General Motors-Carnegie Mellon University Autonomous Driving Collaborative Research Laboratory (AD-CRL).

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] L. Cao and J. Krumm. From gps traces to a routable road map. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 3–12, 2009.

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

[4] 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.

[5] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *Proceedings of Computer Vision and Pattern Recognition*, pages 1964–1971, 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] T. Geraud and J.-B. Mouret. Fast road network extractioin in satellite images using mathematical morphology and markov random fields. *Journal on Applied Signal Processing*, 16:2503–2514, 2004.

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

[9] 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.

[10] 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.

[11] S. Kluckner, M. Donoser, and H. Bischof. Super-pixel class segmentation in large-scale aerial imagery. In *Proceedings of Annual Workshop of the Austrian Association for Pattern Recognition*, 2010.

[12] 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.

[13] 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.

[14] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[15] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[16] 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.

[17] V. Mnih and G. E. Hinton. Learning to detect roads in high-resolution aerial images. In *Proceedings of European Conference on Computer Vision*, pages 210–223, 2010.

[18] V. Nair and J. J. Clark. An unsupervised, online learning framework for moving object detection. In *Proceedings of Computer Vision and Pattern Recognition*, pages 317–324, 2004.

[19] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clusterings: Analysis and an algorithm. In *Proceedings of Neural Information Processing Systems*, pages 849–856, 2001.

[20] M. Ortner, X. Descombes, and J. Zerubia. Building outline extraction from digital elevation models using marked point processes. *International Journal of Computer Vision*, 72(2):107–132, 2007.

[21] 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.

[22] 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.

[23] Y.-W. Seo. *Augmenting Cartographic Resources and Assessing Roadway State for Vehicle Navigation*. PhD thesis, The Robotics Institute, Carnegie Mellon University, April 2012. tech. report CMU-RI-TR-12-13.

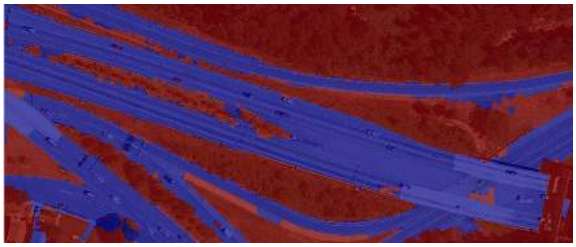
[24] Y.-W. Seo, C. Urmson, D. Wettergreen, and J.-W. Lee. Augmenting cartographic resources for autonomous driving. In *Proceedings of ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 13–22, 2009.

[25] 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.

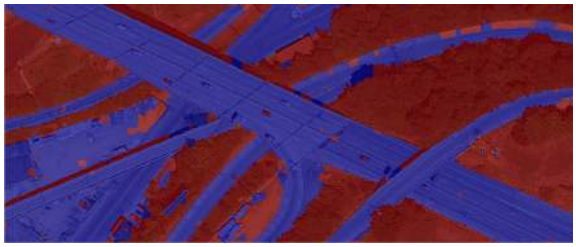
[26] A. Tabibiazar and O. Basir. Kernel-based modeling and optimization for density estimation in transportation systems using floating car data. In *Proceedings of International IEEE Conference on Intelligent Transportation Systems*, pages 576–581, 2011.

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

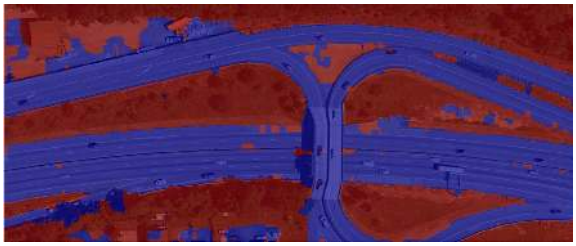
[28] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *Proceedings of International Conference on Computer Vision*, pages 975–982, 1999.



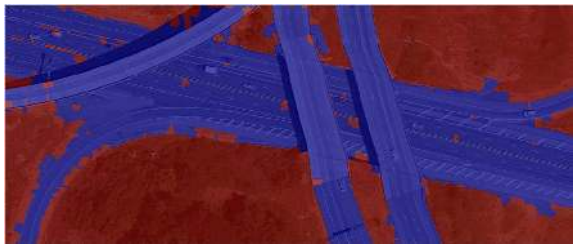
(a) $p = 0.944, r = 0.907, F1 = 0.925$.



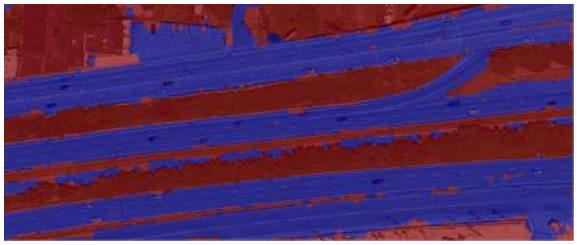
(b) $p = 0.754, r = 0.945, F1 = 0.839$.



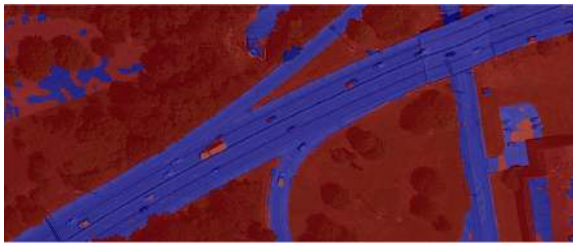
(c) $p = 0.844, r = 0.941, F1 = 0.890$.



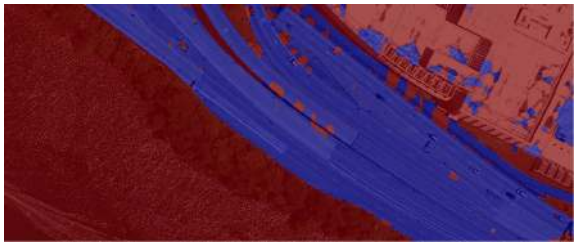
(d) $p = 0.894, r = 0.969, F1 = 0.930$.



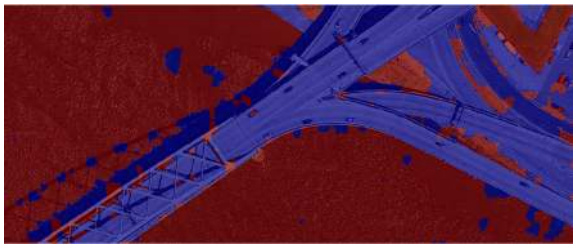
(e) $p = 0.940, r = 0.848, F1 = 0.892$.



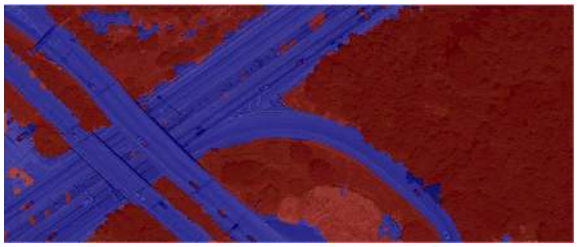
(f) $p = 0.910, r = 0.799, F1 = 0.851$.



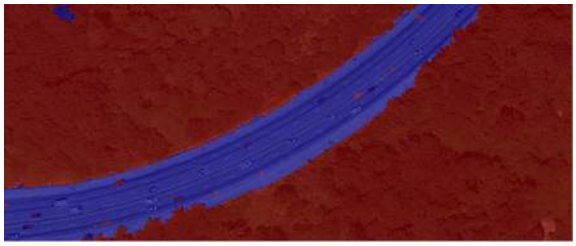
(g) $p = 0.850, r = 0.961, F1 = 0.902$.



(h) $p = 0.747, r = 0.947, F1 = 0.835$.



(i) $p = 0.853, r = 0.968, F1 = 0.907$.



(j) $p = 0.975, r = 0.985, F1 = 0.980$.

Figure 5: Some of the road image-region segmentation results by GMM w/ MRF are shown. Viewed best in color.

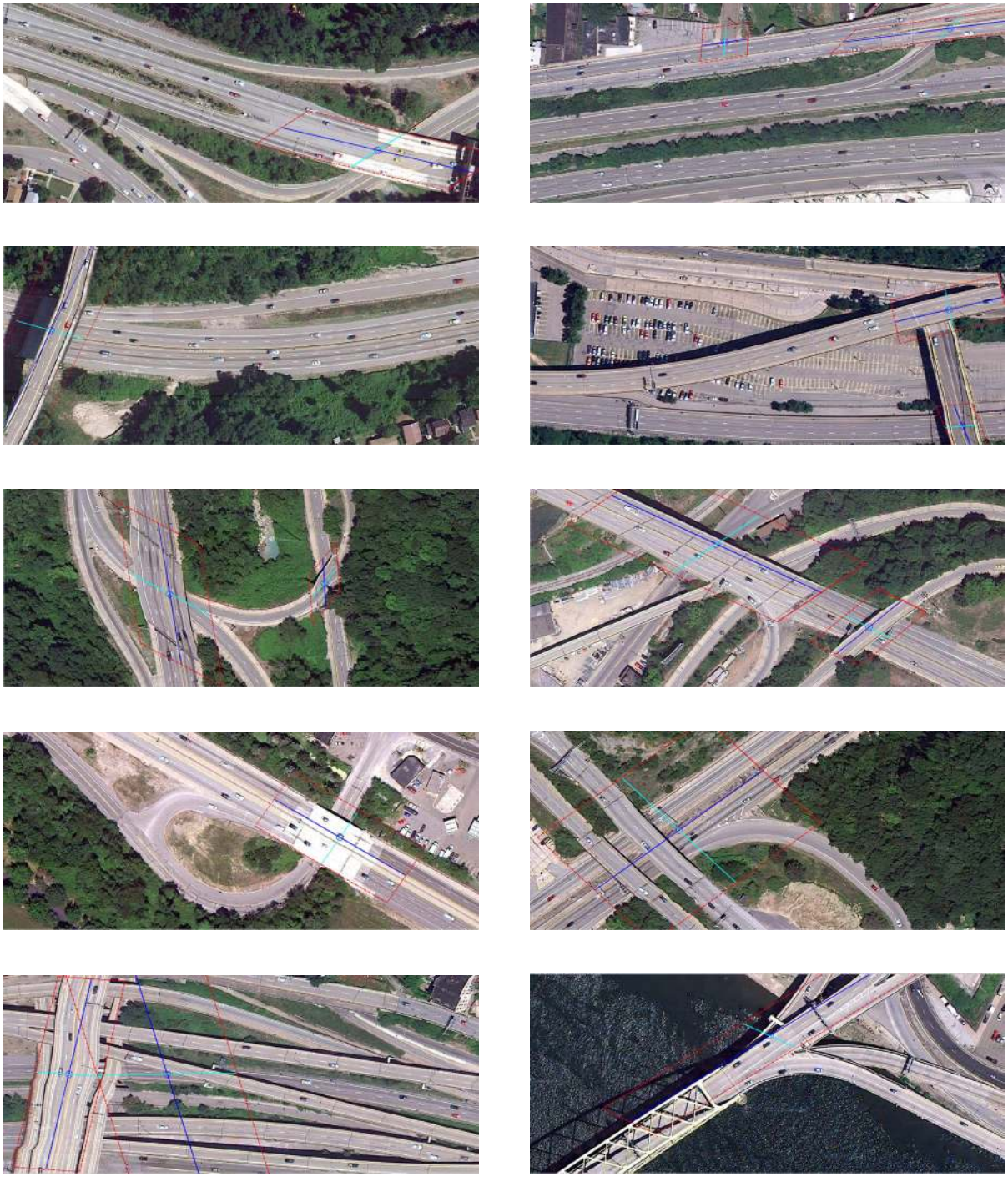


Figure 6: Some of the overpass detection results are shown. Rectangles with dashed (red) lines indicate the boundaries of the detected overpasses and two solid (blue and cyan) lines represent two axes of the boundary rectangles. Note that our algorithm intentionally drew some of the boundary lines at outside of image bounds for the completeness. These figures are best viewed in color.