

# Learning to Count Buildings in Diverse Aerial Scenes

Jiangye Yuan  
Computational Sciences & Engineering Division  
Oak Ridge National Laboratory  
Oak Ridge, Tennessee 37831  
yuanj@ornl.gov

Anil M. Cheriyaadat  
Computational Sciences & Engineering Division  
Oak Ridge National Laboratory  
Oak Ridge, Tennessee 37831  
cheriyadatam@ornl.gov

## ABSTRACT

Determining the number of buildings in aerial images is an important problem because the information greatly benefits applications such as population estimation, change detection, and urbanization monitoring. In this paper, we address this problem by learning the relationship between low-level image features and building counts. Building footprints from public cartographic databases are used as labeled data. We first extract straight line segments from images. A classifier is then trained to identify line segments corresponding to building edges. Although there exist mismatches between resulting line segments and building edges, we observe a strong linear relationship between building numbers and line numbers for similar types of buildings. Based on this observation, we predict the building count for a given image using the following method. We find top  $k$  images with the most similar appearances from training samples and learn a linear regression model from this image set. The building count is computed based on the model. Our method avoids the difficulty in building detection and produces reliable results on large, diverse datasets.

## Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;  
D.2.8 [Software Engineering]: Metrics—*complexity measures, performance measures*

## General Terms

Algorithms, Experimentation

## Keywords

Building count, Aerial images, Straight line extraction

## 1. INTRODUCTION

The number of buildings in an area is highly desirable information in many geospatial related applications ranging from disaster management to urban planning. An effective and economical data source for acquiring such information

is aerial images including satellite images and airborne images. A human interpreter can conveniently count buildings in images, but it is a tedious and time-consuming task. To automate this process, one option is to employ building detection methods that are designed to detect individual buildings in aerial images. Unfortunately, despite decades of research, reliably identifying individual buildings in diverse aerial scenes is still challenging [2]. The main reason is that building appearances vary significantly not only due to different roof materials, building designs, and lighting conditions, but occlusions by shadows and other surrounding objects. Published work on building detection generally establishes some prior criteria for building appearances and identifies objects that satisfy the criteria [19, 17, 7, 13]. Although showing promising performance on certain sample images, such approaches have not been shown to work on large datasets containing diverse scenes. Note that many building detection methods utilize LiDAR data that provides 3D information and can achieve much more reliable performance [15, 21, 1]. However, LiDAR data are considerably more expensive than images. In this work, we will only rely on optical images.

We approach the problem of counting buildings from a new perspective. Instead of identifying buildings in images, we propose to learn the relationship between building counts and low level features and infer building counts directly based on low level features. In particular, we use straight line segments as the low-level features because buildings are typically characterized by straight edges formed by the contrast between building roofs and other objects. By using building footprints from public cartographic resources as labeled data, we adaptively learn a linear regression model to predict building counts. Our strategy has two major advantages. First, low-level features in images are much easier and more reliable to extract than high-level information such as individual buildings [9]. Second, we can leverage a massive amount of ancillary data to apply our method to very large-scale datasets. Unlike many machine learning tasks that suffer from insufficient labeled data, public cartographic databases provide abundant human-labeled building footprints that are easily accessible. Exploiting such data to enhance image understanding capabilities has cultivated a number of recent research efforts. [10, 12, 20].

The contributions of this work can be summarized as follows.

- We propose an improved line segment extraction method.

The method is computationally efficient and produces lines well aligned with edges even when an edge has a low contrast and appears noisy.

- After collecting a large number of samples, we make an important observation that the number of buildings is linearly correlated to the number of extracted line segments when buildings have similar appearances.
- We design a classification method that identifies line segments corresponding to building edges based on image appearances of surrounding areas.
- We develop a method to predict building counts by learning a linear regression model from similar images. The method counts buildings accurately for images containing diverse types of buildings.

The rest of the paper is organized as follows. Section 2 describes the data sources used in this work. Section 3 presents our line extraction method. The method for estimating building counts are discussed in detail in Section 4. In Section 5 we conduct experiments on large datasets and provide quantitative evaluation. We conclude in Section 6.

## 2. DATA SOURCES AND PREPROCESSING

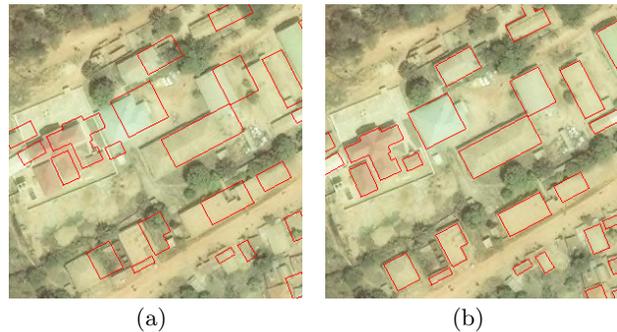
In our work, we use geo-referenced orthorectified images with 3 color bands. Although more spectral bands can potentially improve results, in this work we focus on RGB color images. In order to develop a learning method to count buildings, we need labeled data for training and testing. OpenStreetMap (OSM)<sup>1</sup> provides an ideal data source for such a purpose. OSM maps are publicly available and has detailed building footprints for many cities around the world. Moreover, as a volunteered geographic information platform, OSM has over one million contributors to create and edit geographic data [5], and therefore the map coverage will keep expanding.

Because OSM maps are generated using data sources different from our images, there may exist inconsistency between maps and images. One type of inconsistencies is mismatched features. For example, a map shows a building which is not in the corresponding image, or vice versa. This issue is mostly caused by the time difference between maps and images. Such inconsistencies are often limited in properly selected datasets.

Another type of inconsistency is misalignments between maps and images, which results from different projections and accuracy levels among data sources. Figure 1(a) shows an example of building footprints overlaid with the corresponding image. There are noticeable misalignments between building footprints and the image. Such misalignments lead to inaccurate training samples for line classification and building count estimation and need to be corrected.

We apply a simple preprocessing to reduce the inconsistencies. We assume that in a local neighborhood the building footprints can be aligned with image content through a translation. Despite the lack of theoretical justification,

<sup>1</sup><http://www.openstreetmap.org/>



**Figure 1: Misalignment correction. (a) Building footprints overlaid with the corresponding images. (b) The result after correction.**

this assumption leads to satisfactory results in practice. For an image window containing closely located buildings, we compute the image gradient and perform a cross-correlation between building footprints and gradient magnitude. If the building footprints and images are correctly aligned, the correlation coefficient should reach its maximum. The correction result of the data in the example can be seen in Figure 1(b).

## 3. STRAIGHT LINE EXTRACTION

It is a common practice to rely on some low level image features for finding buildings, such as corners and edges [11, 14]. In this work, we use straight line segments because a major discriminative characteristic of buildings from an aerial view are straight edges. For line segment extraction, Burns *et al.* proposed an important method based on line support regions [3], where each connected region with similar gradient orientations is segmented and line parameters are estimated based on the region. In the paper, we follow this framework and design a new approach to estimate line parameters, which generates accurate results with enhanced efficiency.

We use a  $7 \times 7$  derivative of Gaussian filter with  $\sigma$  equal to 1.2 to compute derivatives in the horizontal and vertical directions, which provide the gradient direction and magnitude at each pixel. For pixels with gradient magnitude larger than a threshold, their gradient directions are quantized into 8 equally divided bins between  $0^\circ$  and  $360^\circ$ . Each connected region containing pixels with the same directions forms a line support region (i.e., a region containing a line segment). The direction quantization may cause a line to be broken. To address this issue, the directions are quantized into another 8 bins between  $22.5^\circ$  and  $(360 + 22.5)^\circ$ , and a different set of line support regions are produced based on the quantization. The lines extracted from two sets of line support regions are integrated through a voting scheme.

Given a line support region, we need to determine the location, length, and orientation of a line segment. In Burns' method, line orientations are estimated by fitting planes to pixel intensities in line support regions, and locations and lengths are obtained by intersecting a horizontal plane with the fitted planes. This method gives accurate results but is

computationally expensive. In order to improve efficiency, a number of studies estimate line parameters based on boundary shapes of line support regions [16, 18]. However, region boundaries do not always reflect the actual orientations and locations of lines. For example, a line support region can be elongated perpendicularly to the actual line in the region when the edge is short and blurred. To overcome the drawbacks while keeping a low computational cost, we exploit the technique of Harris edge and corner detector [6] to determine line orientations. For a line support region, if we shift the region and compute the pixel difference, the largest difference occurs when the shift is perpendicular to the main edge in the region, and the smallest difference occurs when it is along the edge, which corresponds the line orientation. We construct a structure tensor

$$A = \begin{bmatrix} \sum_W I_x^2 & \sum_W I_x I_y \\ \sum_W I_x I_y & \sum_W I_y^2 \end{bmatrix}, \quad (1)$$

where  $I_x$  and  $I_y$  are the derivatives in horizontal and vertical directions in a line support region  $W$ . The shift vector resulting in smallest difference, which indicates the line orientation, is the eigenvector corresponding to the smaller eigenvalues of  $A$ . Such a line orientation is derived from the gradients of all pixels in the region and thus is more robust to noise.

After obtaining the orientation, we need to locate the line segment such that it is best aligned with the edge in the line support region. Here we examine the overall gradient magnitude a line passes and choose the one that gives the maximum value. We use a fast implementation based on Hough transform. A line is represented as  $r = x \cos \theta + y \sin \theta$ , where  $\theta$  can be calculated from the line orientation. Each pixel location  $(x, y)$  in the line support region is plugged into the equation to obtain an  $r$  value, which is assigned to a quantization bin with a weight of its gradient magnitude. The bin with the maximum value gives the desired  $r$  value, which together with the orientation defines a unique line. The part of the line overlapping with the line support region determines the length of the line segment.

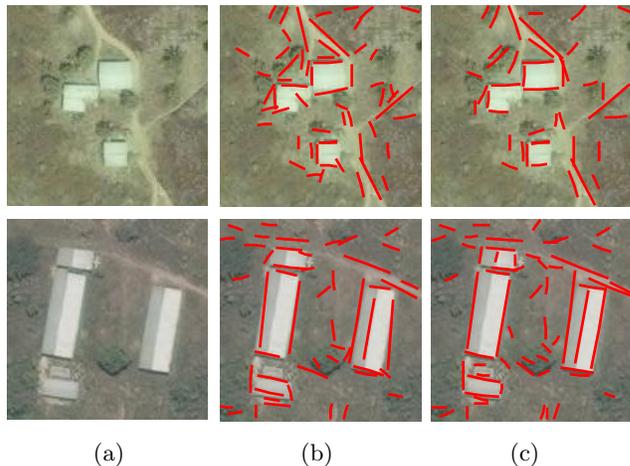
Figure 2 shows the lines extracted using our method and the method in [16], which is based on line support regions and applies Fourier transform to region boundaries for line parameter estimation. It can be seen that the lines in our results are better aligned with the edges, especially when the edges are blurred and noisy. The two methods have comparable computation time.

## 4. LEARNING TO COUNT BUILDINGS

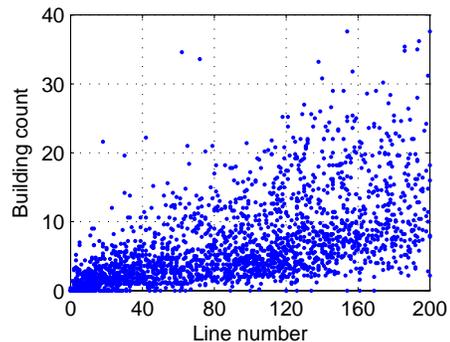
Our goal is to learn the relationship between line segments and buildings, which is utilized to estimate building counts. It is a supervised learning with OSM building footprints used as labeled data. We now describe the method in detail.

### 4.1 Line-building relationship

To investigate the relationship between lines and buildings, we conduct the following experiment. We collect over 2000 building images of 0.6 meter resolution from different places in the world. The size of each image is  $150 \times 150$  pixels, corresponding to a  $60\text{m} \times 60\text{m}$  area. Such an area generally contains multiple buildings with approximately homogeneous structures. For each image, we extract line segments and



**Figure 2: Comparison of line segment extraction results on two images. (a) Input images. (b) The results from [16]. (c) Our results.**



**Figure 3: Scatter plot of line and building numbers.**

manually count buildings. We plot the number of line segments and the number of building for all images, as shown in Figure 3. There is no clear relationship between the two variables.

Next, we perform the same analysis on images with similar buildings. We select a number of exemplar images with different building appearances and assign other images to their similar images. To measure image similarity, we use spectral histograms as image descriptors, which consist of histograms of different filter responses [8]. Spectral histograms have shown to be capable of differentiating image appearances with properly selected filters. We use RGB color bands and filter responses of three Laplacian of Gaussian filters to compute spectral histograms and Euclidean distance as a distance metric. As a result, images are grouped based on appearances. We have also experimented with bag-of-words representations built on SIFT features [4], but the results of spectral histograms are more visually meaningful. We plot the number of line segments and the number of buildings for images in each group. Figure 4 shows the plots of three groups. Four example images from each group are displayed

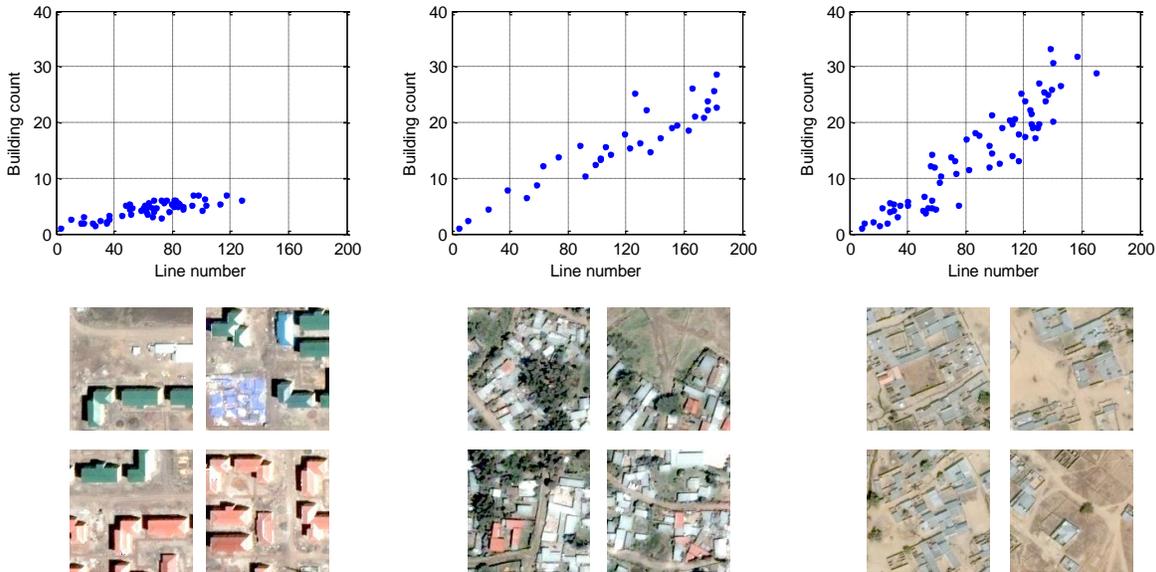


Figure 4: Scatter plots of line and building numbers for different image groups.

below each plot.

A striking observation from Figure 4 is that there is a strong linear relationship between line and building numbers. We calculate the Pearson correlation coefficient, which measures the strength of the linear relationship between two variables and equals to 1 in the case of a perfect linear relationship. The Pearson correlation coefficients for the three groups are 0.85, 0.91, and 0.86, respectively. Linear relationships are also observed for other groups. The main reason for such a line-building relationship is that buildings with similar structures tend to exhibit similar numbers of edges from an aerial view. Although extracted line segments do not perfectly match building edges, the mismatches appear consistent and do not severely affect the linear relationship. There are a few images that noticeably deviate from the linear relationship. We find that in those images many non-building line segments are counted, which often correspond to roads and trees. A stronger linear relationship can be expected if non-building line segments are filtered out.

Based on this observation, we use a simple linear regression model to associate building numbers with line segment numbers,  $y = \beta x$ , where  $x$  is the line segment number,  $y$  the building count, and  $\beta$  the regression coefficient. This model provides an effective solution for counting buildings with similar appearances. We only need to select several small areas to manually count building numbers and extract straight line segments, which are used to estimate  $\beta$  through the least square approach. The building number in the entire area is equal to the number of extracted line segments multiplied by  $\beta$ .

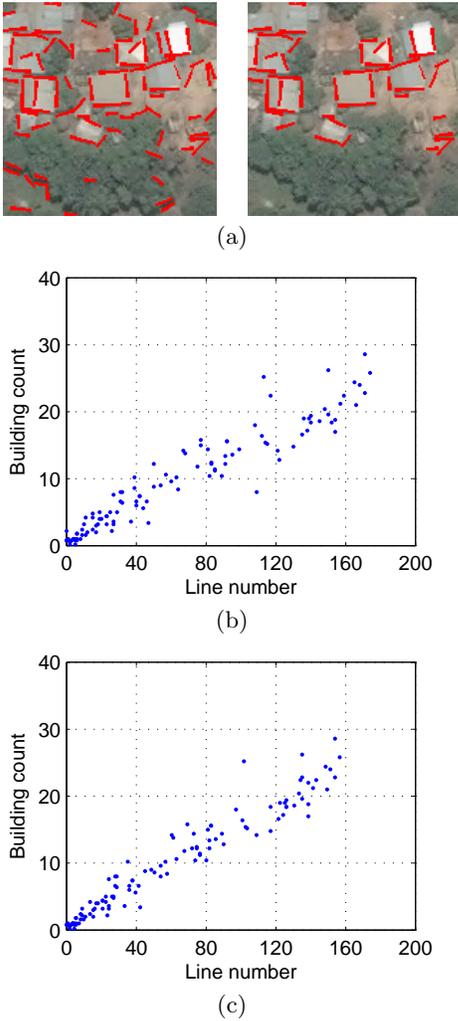
Another observation we have from Figure 4 is that for different groups the line-to-building ratio is different. That is, the linear regression models may have different regression coef-

ficients for different types of buildings. For example, for the leftmost group in the figure the building number increases slowly as the line number increases because each building correspond to more lines in that group. Therefore, we cannot apply a single model to all types of buildings.

## 4.2 Line segment classification

Line segments from non-building areas should not contribute to any building counts. Removing those line segments can strengthen the linear relationship between line segment numbers and building counts. Here we aim to identify line segments corresponding to building edges. We train a multi-layer perceptron (MLP) to classify line segments based on surrounding image appearances.

Based on the line segments extracted from images and the corresponding building footprints (with alignments corrected as described in Section 3), we label each line segment as 1 if its maximum distance to a building edge is smaller than 3 meters and half its length and 0 otherwise. The feature used for classification is spectral histograms. Note that spectral histograms can be used to compare image content regardless of region sizes. For each line segment, the feature is computed from the region within a certain distance to the line segment. A distinctive attribute of building edges is co-occurrence of perpendicular edges. To encode such information in the classifier, we convert RGB values to grayscale values and apply two derivative of Gaussian filters, one with the same orientation as the line segment and the other perpendicular to the line segment. The two filter responses together with RGB color bands are used to compute spectral histograms, where each band is represented by a histogram with 11 equally divided bins. We use two neighborhood sizes to capture information at multiple scales. The MLP has 110 input nodes to take all feature dimensions, one hidden layer with 70 nodes, and 1 output node. Since building lines are



**Figure 5: Line segment classification.** (a) Left: extracted line segments. Right: Line segments classified as building lines. (b) and (c) Scatter plots of line and building numbers before and after line segment classification.

often much fewer than non-building lines, the errors during training are weighted based on the size ratio between two classes so that the result is not biased toward the large class. After training, the MLP classifier gives the posterior probability of a line segment belonging to building edges.

Figure 5(a) illustrates the result of line segment classification on an image. Line segments in non-building areas are greatly reduced. Figure 5(b) and (c) show two scatter plots of line and building numbers for one of the image groups mentioned in Section 4.1, where the number of filtered line segments has a higher degree of linear dependence to the building number. The Pearson correlation coefficient increases from 0.89 to 0.92 by filtering line segments.

### 4.3 Building count estimation

As discussed earlier, a single linear regression model cannot apply to different types of buildings. To deal with this

issue, we propose to select images similar to the input image from training samples and establish a linear regression model based on similar images to estimate building counts. Training samples comprise images in the training set, corresponding building counts obtained from building footprints, and line segments extracted from the images. To measure image similarity, we use spectral histograms as image descriptors and Euclidean distance as a distance metric. To compute spectral histograms, we use RGB color bands and filter responses of three Laplacian of Gaussian filters with different  $\sigma$  values. After obtaining the  $K$  most similar images from the training pool, their line segment numbers and building counts are taken to estimate the regression coefficient using the least-squares approach. Line segments are extracted from the input image and filtered by the trained MLP. The building number of the input image is immediately obtained based on the regression coefficient and the line segment number.

We use a method based on  $K$ -nearest-neighbor ( $K$ -NN) search to adaptively learn a line-building relationship because such a method is well suited for our task. Since there are potentially infinite types of buildings, learning a model for each type is intractable.  $K$ -NN can naturally deal with a very large number of classes. Moreover, new training samples can be easily added without the need of retraining.

The complete procedure of our method can be described as the following three steps.

1. Compile a training set that includes images and the corresponding building footprints. Building counts of each image is determined based building footprints.
2. Extract line segments for images in the training set. Label each line segment based on whether it is aligned with edges in building footprints. Use spectral histograms as features to train a MLP for line segment classification. Record the number of line segments filtered by the MLP.
3. Given an input image, extract line segments and count those classified as building edges by the trained MLP. Find the  $K$  most similar images from the training set and use their line numbers and building counts to derive a linear regression model, which produces the building count based on the line segment number.

## 5. EXPERIMENTS

We conduct experiments on two datasets, which will be referred to as Dataset I and Dataset II. Two datasets correspond to very different geographic areas.

Dataset I covers the urban areas in San Francisco, CA. We collect two  $5000 \times 5000$  image tiles with spatial resolution of 0.3 meters. We randomly select 400 images of size  $250 \times 250$  within each image tile. Two sets of images are used for training and testing respectively. The OSM building footprints for the corresponding areas are quite complete. When counting buildings on maps, we count a partial building as one if the part contains more than half area of the entire building or an area larger than 50 square meters. According to the map data, the number of buildings in these images ranges

**Table 1: Percentage of correctly counted images with different error tolerance for Dataset I**

Error tolerance	2	3	4	5
Accuracy	66.1%	79.0%	88.6%	92.9%

**Table 2: Average count error with different  $K$  values on Dataset I**

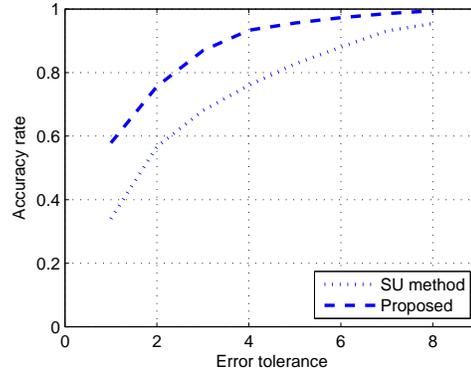
$K$	3	4	5	6
Count error	3.08	2.78	2.51	2.51

from 0 to 31, and the average is 12. In the experiment, pixels with gradient magnitude larger than 40 are selected to identify line support regions, and line segments shorter than 3 meters are removed in order to reduce noise. For line segment classification, we use the MATLAB neural network toolbox to construct and train a MLP. For searching  $K$  most similar images,  $K$  is set to 5.

We show some example images from the test set and their building counts from map data and our method in Figure 6(a). The counts from our method are rounded to integers. As can be seen, building appearances vary to a large extent. Moreover, many buildings are adjacent to each other, where individual buildings are very difficult to detect. The counts from our method are very close to those from maps. To better show the diversity of scenes, we apply the method to two areas corresponding to highly different city blocks. Each area is divided into  $250 \times 250$  image windows for processing, and the total count is obtained by aggregating the results. Line segments are extracted for the entire area. For each image window, we only count the line segments with the centroids inside the window so that large buildings with long line segments are not double counted. The results are shown in Figure 7.

To quantitatively measure the results, we calculate the count error by comparing the counts from our method and maps. The average count error is 2.51. To provide a more detailed measurement, we compute the percentage of correctly counted images at different levels of error tolerance (the maximum allowable deviation from the count based on maps), which are reported in Table 1. Our method produces correct counts for 66.1% images with an error tolerance of 2. The accuracy rate reaches 92.9% with an error tolerance of 5. We also calculate the average count errors using different  $K$  values in  $K$ -NN search (see Table 2). We can see that the results are not overly sensitive to this parameter value.

Dataset II covers the small city of Kissidougou in southern Guinea. The spatial resolution of images is 0.6 meters. We use a  $4500 \times 2550$  image tile corresponding to the south part of the city for training and a  $4500 \times 3900$  image tile corresponding to the north part for testing. We randomly select 510 images from the training image tile and 780 images from the test image tile, where each image is of size  $150 \times 150$  pixels. We use the same parameter setting as for Dataset I except adjusting the gradient magnitude threshold to 20 because of the different image resolution and quality. For this dataset, our counting result has an average count error



**Figure 9: Accuracy rates of the SU and proposed methods.**

of 1.73. Example results are presented in Figure 6(b). We also select two different areas and give the total counts, as shown in Figure 8.

We do not find any previous work that explicitly aims at counting buildings from images. However, this task is closely related to building detection. If buildings in an image are detected, counting buildings is trivial. On the other hand, if we apply our method to each small window of an image so that the counts are localized at a fine scale, the result is close to that of building detection. For comparison, we select a leading building detection method proposed by Sirmacek and Unsalan [13], which will be referred to as the SU method. The method extracts SIFT keypoints and constructs graphs based on the keypoints. The buildings are identified through subgraph matching, which can handle occluded buildings. We use the code distributed by the authors.

Since the SU method cannot detect buildings that are closely spaced, it fails to produce reasonable results for Dataset I that contains dense buildings. We apply the SU method to Dataset II and calculate the percentage of correctly counted images as described earlier. Figure 9 presents the plot of the accuracy rates for both methods. As can be seen, our method outperforms the SU method by a significant margin. By examining the results, we find that the SU method tends to miss buildings with a low contrast to the surrounding areas because there is often no SIFT feature extracted for those buildings. In our method, line segments can be extracted for those buildings and they contribute to the final count.

## 6. CONCLUSIONS

We have presented a method that automatically counts buildings in aerial images. We observe that the number of buildings in images are linearly correlated to the line segment number. By using building footprints from public cartographic databases as labeled data, we adaptively learn a linear regression model to estimate building counts in a given image. We test the method on two large datasets containing diverse building scenes and obtain very promising results.

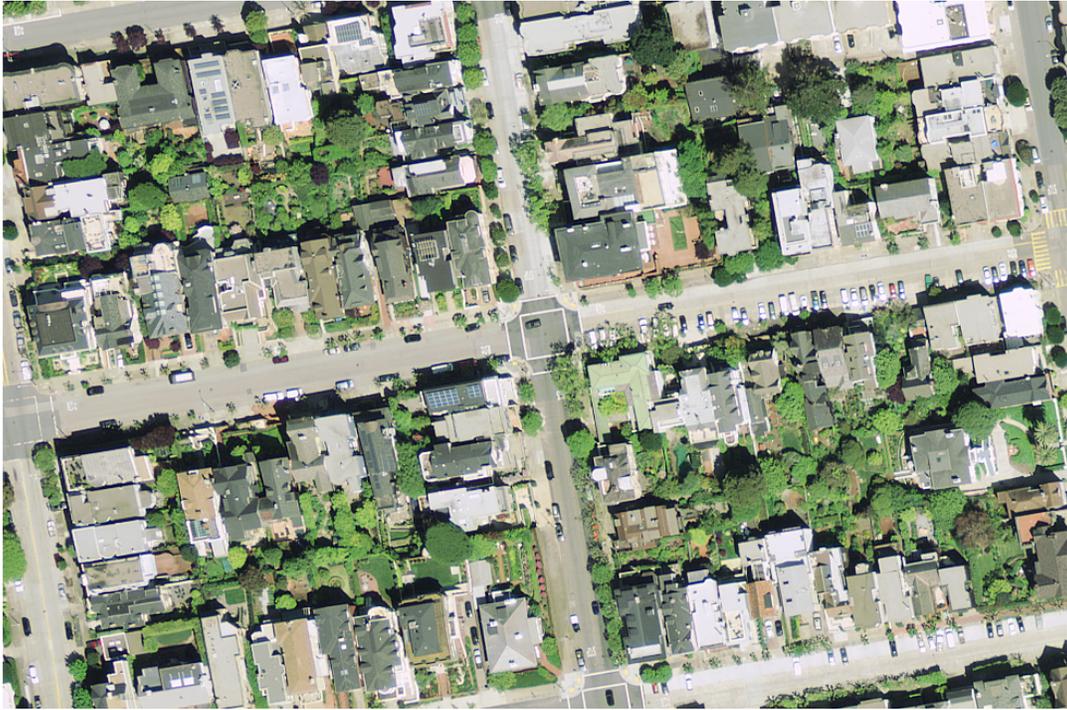


(a)

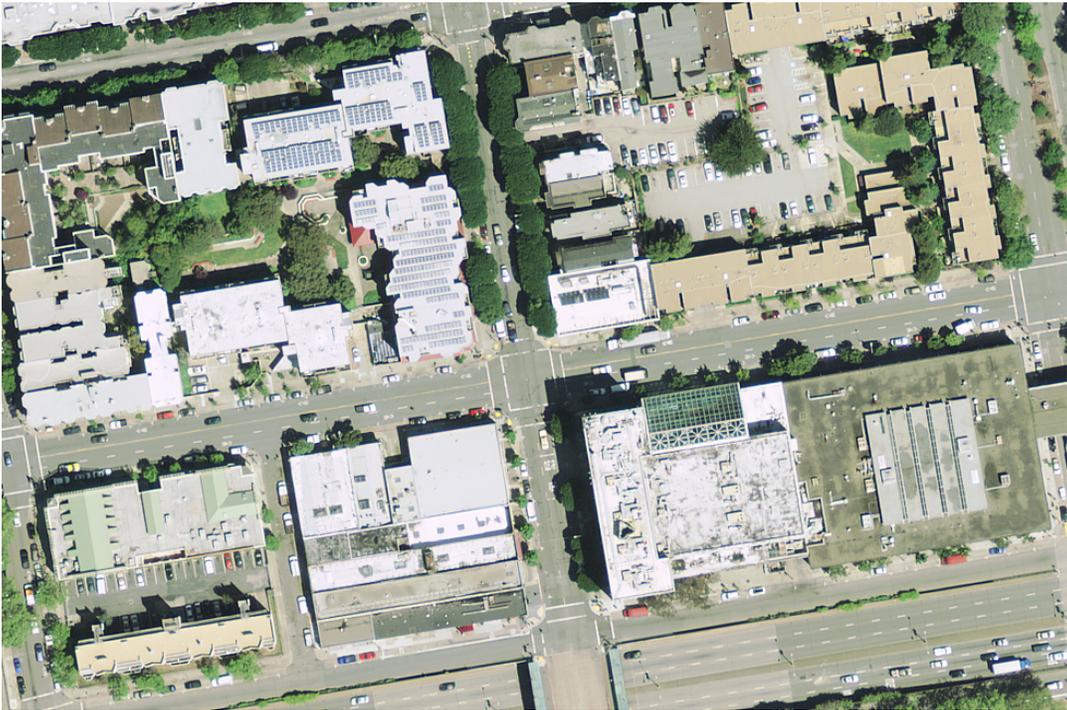


(b)

Figure 6: Example building count results for individual image windows. (a) Dataset I. (b) Dataset II. Buildings counts from maps and our automatic method are shown below each image. M stands for maps, and A our automatic method.



M: 112 A:121



M: 47 A:55

Figure 7: Building count results for different city blocks in Dataset I.



M: 153 A:166



M: 185 A:201

**Figure 8: Building count results for different areas in Dataset II.**

There are several directions for future work. First, based on the experiments we find that many incorrect counts come from images containing multiple types of buildings, where the learned model cannot correctly describe the line-building relationship. To reduce such errors, a plausible solution is to first segment the image based on texture information so that similar buildings form a segment and then estimate building counts for each segment. The choice of segmentation methods needs to be investigated. Second, the output of the current method is the number of buildings. In future studies, we plan to derive more information for buildings based on low-level features. For example, it appears feasible to estimate building sizes based on spatial distributions of line segments.

## 7. ACKNOWLEDGMENTS

This manuscript has been authored by employees of UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. Accordingly, the United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The authors would like to acknowledge the financial support for this research from the US Government for the development of LandScan USA model and database.

## 8. REFERENCES

- [1] M. Awrangjeb, M. Ravanbakhsh, and C. S. Fraser. Automatic detection of residential buildings using LiDAR data and multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(5):457–467, 2010.
- [2] T. Blaschke. Object based image analysis for remote sensing. *ISPRS journal of photogrammetry and remote sensing*, 65(1):2–16, 2010.
- [3] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):425–455, 1986.
- [4] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop on statistical learning in computer vision*, 2004.
- [5] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.
- [6] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conferencel*, 1988.
- [7] K. Karantzas and N. Paragios. Recognition-driven two-dimensional competing priors toward automatic and accurate building detection. *IEEE Transactions on Geoscience and Remote Sensing*, 47(1):133–144, 2009.
- [8] X. Liu and D. Wang. A spectral histogram model for texture modeling and texture discrimination. *Vision Research*, 42(23):2617–2634, 2002.
- [9] H. Mayer. Object extraction in photogrammetric computer vision. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2):213–222, 2008.
- [10] V. Mnih and G. E. Hinton. Learning to detect roads in high-resolution aerial images. In *Proceedings of European Conference on Computer Vision*. 2010.
- [11] M. Molinier, J. Laaksonen, and T. Hame. Detecting man-made structures and changes in satellite imagery with a content-based information retrieval system built on self-organizing maps. *IEEE Transactions on Geoscience and Remote Sensing*, 45(4):861–874, 2007.
- [12] Y.-W. Seo, C. Urmson, and D. Wettergreen. Exploiting publicly available cartographic resources for aerial image analysis. In *Proceedings of ACM SIGSPATIAL international conference on Advances in geographic information systems*, 2012.
- [13] B. Sirmacek and C. Unsalan. Urban-area and building detection using sift keypoints and graph theory. *IEEE Transactions on Geoscience and Remote Sensing*, 47(4):1156–1167, 2009.
- [14] B. Sirmacek and C. Unsalan. A probabilistic framework to detect buildings in aerial and satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 49(1):211–221, 2011.
- [15] G. Sohn and I. Dowman. Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(1):43–63, 2007.
- [16] C. Unsalan and K. L. Boyer. Classifying land development in high-resolution panchromatic satellite images using straight-line statistics. *IEEE Transactions on Geoscience and Remote Sensing*, 42(4):907–919, 2004.
- [17] C. Ünsalan and K. L. Boyer. A system to detect houses and residential street networks in multispectral satellite images. *Computer Vision and Image Understanding*, 98(3):423–461, 2005.
- [18] R. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.
- [19] Y. Wei, Z. Zhao, and J. Song. Urban building extraction from high-resolution satellite panchromatic image using clustering and edge detection. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, 2004.
- [20] J. Yuan and A. M. Cheriyyadat. Road segmentation in aerial images by exploiting road vector data. In *Proceedings of International Conference on Computing for Geospatial Research and Application*, 2013.
- [21] Q.-Y. Zhou and U. Neumann. Fast and extensible building modeling from airborne lidar data. In *Proceedings of ACM SIGSPATIAL international conference on Advances in geographic information systems*, 2008.