Development of a Building Detection System from an Aerial Image Based in Watershed Transformation and Linear Support Vector Machine

Michael Anthony Jay B. Regis¹*, Concepcion L. Khan², Jaime M. Samaniego², Serlie B. Jamias² and Vladimir Y. Mariano²

¹Visayas State University, Visca, Baybay City, Leyte, Philippines
²University of the Philippines Los Baños, College, Laguna, Philippines

Abstract
Object detection in an aerial image has always been a fundamental problem in remote sensing, more so with increasing population. With the advancement in sensor technology and falling prices in imaging hardware, it is now cheaper to acquire aerial images as compared to a decade ago. With increased quality and quantity of the images gathered, it is necessary to develop an automated object detection system to address tedious manual building detection. In this study, a two staged approach was executed to address automated building detection. First, we performed image segmentation to create meaningful regions of the image using a marker controlled watershed transform. Discrete Fourier Transform (DFT) coefficients were then derived from the grayscale histogram of each region to act as feature vector necessary for the next stage. Second, we trained linear support vector machines (SVM) using the acquired feature vector to identify the building and non-building regions of the test images. We evaluated the performance of the proposed method by using detection percentage, branching factor and receiver operating characteristic (ROC). We trained the linear SVM classifier with 872 building and 616 non-building images from 31 training images of the Calumpang aerial survey. Experimental results from 31 test images (of the same aerial survey) shows that the detection percentage and branching factor is 69.50% and 22.70%, respectively. Moreover, the area under the curve (AUC) of the ROC is 0.887 strongly suggesting that the proposed method is highly effective.

Keywords: Object detection, watershed transform, discrete Fourier transform (DFT), linear support vector machine (SVM), receiver operating curve (ROC)

Introduction

Aerial images are conventional and highly reliable illustration of the Earth’s surface. These images capture an extensive variety of man-made and natural objects. Such images are usually acquired through the use of an unmanned aerial vehicle (UAV) equipped with a mounted downward-facing camera. It offers a higher degree of spatial and temporal resolution compared to satellite images (Naithani, 1988).

A trained human interpreter can easily identify buildings found in aerial images because the human visual system see scenes as it is (e.g. buildings in an aerial image, trees in a landscape, books in shelf, etc). Interpreting a certain scene is carried out with almost no deductions and overt effort with effective answers normally available in a tenth of second (Davies, 2012). However majority of the neurons follow the subjective percept (Kreiman, Koch, & Fried, 2002) thus subjective interpretation of an image is most likely to happen (i.e. demonstrated when interpreting a bistable visual illusions like the Necker cube). So, there is a need for an unbiased computational method when interpreting aerial images.

Building detection is a well-researched field due to its military applications. Early researches were focused on techniques for
pixel-based classification, depending on spectral dimensionality rather than spatial context (Guindon, 1997). Further improvements incorporated image textures or pixel features within a fixed window (Haralick, 1979). More advance studies involved contextual knowledge of low level features (i.e. lines and edges) and high level ones (i.e. shapes and shadow information), then were further enriched by using machine learning algorithms (Maloof, Langley, Binford, Nevatia, & Sage, 2003).

Since buildings directly represent inhabited areas in remote sensing data (Bayburt, Büyüksalih, Baz, Jacobsen, & Kersten, 2008), an automated building detection system can be considered as one of the indispensable tools in mapping out human settlement in urban and sub-urban environments. This would entail in more systematic way to: (1) implement disaster risk evaluation and damage detection (Hosokawa, Jeong, Takizawa, & Matsuokac, 2008), (2) estimate locality population and growth (Baudot, 1993), and (3) automatic land use analysis and measurement of areas for public utilization (Muller & Zaum, 2005).

In this study, we propose the use of Red-Green-Blue (RGB) aerial images as input for successive stages culminating at the detection of buildings.

**Methodology**

The flowchart shown on Figure 1 illustrates the overview of the proposed building detection solution.

**Watershed Transform**

Image segmentation was initially employed to create meaningful regions in an image. It is considered as one of the most important problems in image processing since it would give the initial summary of information from an image before it is passed into another processing step (Forsyth & Ponce, 2012). Watershed transform is a segmentation algorithm that is classified as a clustering based approach. The elegant idea of this method comes from geography. If a landscape or any topological surface is flooded, watersheds would serve as regions in which rain water would be attracted (Figure 2). On another perspective, imagine a level landscape with holes immersed in a lake. Obviously, water would flow from the edges of the holes to create catchment basin out of these holes. When the water level has reached the highest level ground on the landscape, then the immersion would stop. As a result, regions known as watershed areas would be formed separated by dams (Roerdink & Meijster, 2001).
The common problem with using watershed transform especially when using randomized markers is oversegmentation (Bieniecki, 2004). To prevent oversegmentation, we implemented automatic marker extraction. It was done through thresholding the image using Otsu’s method (Otsu, 1979), cleaning using morphological opening, applying distance transform, thresholding the distance transformed image, and finding the external contours.

Afterwards, we applied merging of the regions through Bhattacharyya similarity measure of each histogram to combine similar regions.

**Linear Support Vector Machine**

Vapnik (1995) showed that support vector machines have the ability to recognize patterns. This machine learning technique derives the concept from a perceptron (i.e. artificial neural network) but improves it more by maximizing the geometric margin (Bell, 2014). A linear support vector machine (SVM) was operated by using a hyperplane to separate points into two classes. This would group points of the same class on the same side, while a maximum margin is obtained in order to minimize risk of misclassifying samples (Vapnik, 1995). The subset of data points that would define the hyperplane is called support vectors (Figure 3).

In this study, we used the Discrete Fourier Transform (DFT) for the linear SVM from the histogram of each region. DFT would analyze the spectral properties of a digital image (Cyganek & Siebert, 2009).

**Detection Percentage, Branching Factor, Receiver Operating Curve**

For quantitative evaluation of the performance for the proposed method, we first considered the ground-truth of the aerial image. Human visual labelling acted as the source of ground-truth since manual counting of the buildings on the test site would be very difficult. We made comparisons for manually detected buildings and automatically extracted ones. We considered the following metrics:

- **TP (True Positive)** - an object detected both manually and automatically
- **FP (False Positive)** - an object undetected manually but is detected automatically, also termed to as false alarm.
- **FN (False Negative)** – an object detected manually but is not detected automatically.

To evaluate the performance, we employed the performance computations of Lin and Nevatia (1998) which were the following:

\[
\text{Detection Rate (DR)} = \frac{TP}{TP + FN} \tag{1}
\]
Branching Factor ($BF$) = \frac{FP}{TP + FP} \quad (2)

Detection rate computed the rate of true positives on a given algorithm, which is same as the sensitivity. On the other hand, branching factor calculated the false positive rate, and is also known as 1-specificity. Moreover, typical machine learning binary classifiers were evaluated over a range of cost settings which is done using receiver operating characteristic (ROC). The idea is to plot the sensitivity in the y axis and 1-specificity in the x axis for a given binary classifier. Afterwards, an area under the curve (AUC) was computed based on the plotted data (Swets, 1988). If the AUC is $<0.5$ or is close to 1, it is considered effective. ROC was developed to evaluate the detection capability of radar systems during the World War II, but in the late 90’s, this evaluation model was used to test machine learning classifiers.

**Experimental Platform and Test Data**

The OpenCV application program interface (API) was used in this study. It is an open source and cross platform API written in C/C++ and is designed for computational efficiency. Moreover, it can take advantage of multicore processors for enhanced speedup (Bradski & Kaehler, 2008). The hardware environment used was an Intel Core I7 2670QM (i.e. 2nd Generation I-series, Quad Core Mobile processor running at 2.2GHz with turbo boost of up to 2.8GHz) alongside 8GB of RAM with a 64-bit Windows® 7 home premium operating system. Compatibility issues were addressed by compiling the OpenCV libraries with a 64-Bit Windows® version of GNU C++. Finally, Code::Blocks was used as the Integrated Development Environment (IDE) during coding.

The initial site used as training and test images were the aerial photographs from the Calumpang River delta in Batangas City, Philippines. The photo was taken last August 10, 2013 from an unmanned aerial vehicle courtesy of Itera Robota Inc.

**Results and Discussion**

The first step to detection is creating a grayscale image (Figure 4) from the 1000 x 750 resolution RGB aerial image in .PNG format. From the grayscale image, binarization is done using Otsu’s method (Figure 5) wherein watershed regions would start from the white areas of the binary image.

![Grayscale Image](Figure 4)

![Binarized Image](Figure 5)

Morphological erode (Equation 3), then dilate (Equation 4) known also as morphological opening were then employed to remove salt and pepper noise, join separate regions and isolate individual regions (Figure 6). This operation uses a 7x7 pixel structuring element anchored at coordinates (4,4) and is iterated twice.
erode(x,y) = 
\[
\min_{(x',y') \in \text{kernel}} \text{src}(x + x', y + y')
\]
(3)

dilate(x,y) = 
\[
\max_{(x',y') \in \text{kernel}} \text{src}(x + x', y + y')
\]
(4)

Further isolation of region markers were executed through distance transform as shown in Figure 7, since it would create a gradient in which each pixel is equal to distance nearest to zero pixel in the image. The mask for the transformation is automatically defined by specified function and the distance measure used is Cartesian Metric Distance (Equation 5).

\[
p(r) = \frac{r^2}{2}
\]

(5)

The transformed image was further normalized and thresholded (Figure 8). External contours that acted as watershed markers were then derived from this image (Figure 9). Watershed transformation as shown in Figure 10 was applied once the markers were finally derived.

Figure 6: Morphological Transform of Image

Figure 7: Distance Transformed Image

Figure 8: Thresholded image from distance transform

Figure 9: Finding external contours as markers

Figure 10: Watershed transform before region merging

Oversegementation is still common even after a marker controlled watershed segmentation. To address that, region merging was done first of all by converting
each RGB region into its HSV counterpart (Equations 6a, 6b, 6c). A normalized histogram was then derived from each HSV region to act as the input data for the similarity measure. Each histogram was compared to each other using Bhattacharyya distance (Equation 7). If the similarity measure between two histogram \( H_1 \) and \( H_2 \) is 0.9999 then both histograms are merged. Figure 11 shows the output after region merging.

\[
V \leftarrow \max(R, G, B) \quad (6.a)
\]

\[
V \leftarrow \begin{cases} 
\frac{V - \min(R, G, B)}{\max(R, G, B)} & V \neq 0 \\
0 & \text{Otherwise}
\end{cases} \quad (6.b)
\]

\[
H \leftarrow \begin{cases} 
60(G - B) & \text{if } V = R \\
120 + 60(B - R) & \text{if } V = G \\
240 + 60(R - G) & \text{if } V = B
\end{cases} \quad (6.c)
\]

\[
d(H_1, H_2) = \frac{1}{\sqrt{(H_1)(H_2)}} \sum_{I=1}^{N} \sqrt{H_1(I) \times H_2(I)} \quad (7)
\]

Supervised learning was then executed by picking building regions and non-building regions from the 31 training images taken from Calumpang River delta aerial survey. From the 1488 regions of the 31 test images, 872 were selected as building regions while 616 were selected as non-building regions. To separate the building and non-building classes, a linear hyperplane with its appropriate support vectors is computed with termination criteria as follows: maximum iterations must not exceed 1x10^7 and tolerance is 1x10^{-6}. To avoid recomputation, the training data with the support vectors and decision function were saved in a xml file and were later retrieved when the identification process was invoked. Another 31 test images that were different from the training images were selected from the same aerial survey. Figure 12 shows the output of one of the test images after detection.

Building coordinates were then marked from the source image to act as ground truth (Figure 13). Given that a building coordinate is inside the detected building region, it is counted as true positives (TP). On the other hand building coordinates outside the building regions are counted as false negatives (FN). Finally building regions which has no buildings are considered false positives (FP). Each image has its own evaluation result in a text file containing TP, FP, FN and building coordinates.

Table 1 gives the summary of the performance statistics for all test images. An overall DP (also known as sensitivity) of 69.50% indicates good performance in building detection. On the other hand, a BF (otherwise known as 1-specificity) of 22.70% shows that there is a considerable rate of false alarm detected by the system.
Table 1: Evaluation results of the system

<table>
<thead>
<tr>
<th>Test Data (N=31)</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>DP</th>
<th>BF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>1682</td>
<td>494</td>
<td>738</td>
<td>69.50%</td>
<td>22.70%</td>
</tr>
</tbody>
</table>

Figure 12: Output Image, visible areas are building regions, dark areas are non-building regions.

Figure 13: Evaluation Image, crosshairs indicate buildings that are detected manually.

Individual detection percentages and branching factor for each test image were then plotted on a ROC curve graph. The blue curve that is shown in Figure 14 indicates the ROC curve of the building detection system while the green line shows the reference line of the acceptable area. The area covered under the reference line is 0.5 and is considered the lower bound for acceptability. In this study, the area under the curve (AUC) is 0.887, thus, it can be concluded that the system is effective as a binary classifier.

Figure 14: ROC curve graph of the Building Detection System.

Conclusion and Recommendation

Numerous methods already exist in building detection from an aerial image. A method for detecting building using the combination of watershed segmentation and linear support vector machines was proposed in this study. By introducing a general framework in automatic marker extraction for watershed transform, the method pinpointed specific regions in the image. But localization of building regions still needs refinement since there are numerous cases in which a building region is beyond the bounds of building edges. Moreover, oversegmentation within building regions or multiple regions in a homogenous building is another common occurrence after watershed transform. Region merging may have addressed this problem, but it is also responsible for adding false positives since
some building regions are combined with non-building region.

On the other hand, using the DFT coefficients from the grayscale histogram as feature vector for the SVM classifier is quite effective. The linear SVM classifier trained with 872 building regions and 616 non-building regions successfully detected 69.50% of the buildings on the test images. Moreover, an AUC of 0.887 in ROC strongly suggests that the system is effective when detecting building regions.

In the future iteration of this study, continued refinement especially during marker selection must be done accordingly. Moreover, even though markers represent knowledge about building locations, threshold parameters for marker extraction is a drawback that needs to be addressed. It could be that edge strength around buildings areas may be added as an image mask in order to define more clearly the building edges thus improving localization. Contextual knowledge between thresholds and spatial information (e.g. location of buildings with respect to vegetation and roads) can be useful to improve the threshold parameters. Complex rules may also be necessary when merging regions so that it would not generate too many false positives. Furthermore, other features may be tested as feature vector to improve the performance during training. It is also necessary that different kernels (e.g. Radial Basis Function, Polynomial, Sigmoid) would be tested during training and classification in order to determine which kernel is effective when classifying. So far, RGB image is the only input accepted by the method. More advanced bands such as Near Infrared or Ranged data can be applied on top of the existing RGB image in order to enhance the capability of the existing method.

References


Bayburt S., Büyüksalih G., Baz I., Jacobsen K., & Kersten T. (2008). Detection of changes in Istanbul area with medium and high resolution space images. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVII, Part B7


