

An automated airplane detection system for large panchromatic image with high spatial resolution [☆]

Zhenyu An^a, Zhenwei Shi^{a,*}, Xichao Teng^b, Xinran Yu^a, Wei Tang^a

^a*Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, P.R. China*

^b*College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, P.R. China*

Abstract

With a wide range of applications in different fields like airport management and military warfare, airplane detection has been a critical part in remote sensing image processing. In this paper, we focus on the airplane detection in large (usually larger than 10000×10000 pixels) panchromatic image (PI) with high spatial resolution (usually superior to 1m), and propose an automated airplane detection system. The system contains two main modules: In the first module, line segment detector (LSD) is applied to detect runway of an airport, thus segmenting airport region in a large PI and reducing airplane candidates. The other is used to detect airplanes in the segmented airport regions. We first use circle frequency filter to further locating airplane candidates, then accomplish precise detection task by combining Histograms of Oriented Gradients (HOG) descriptor and AdaBoost algorithm. Therefore, besides a practical airplane detection system, the other contributions of our approach include the following three parts: 1) it locates runway of an airport with LSD; 2) it classifies airplane candidates by using circle frequency filter;

[☆]The work was supported by the National Natural Science Foundation of China under the Grants 61273245 and 91120301, the 973 Program under the Grant 2010CB327904, the open funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (Grant No. BUAA-VR-12KF-07), and Program for New Century Excellent Talents in University of Ministry of Education of China under the Grant NCET-11-0775. The work was also supported by Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, P.R. China.

*Corresponding author. Tel.: +86-10-823-39-520; Fax: +86-10-823-38-798.

Email address: shizhenwei@buaa.edu.cn (Zhenwei Shi)

3) it precisely detects airplanes by exploiting HOG and AdaBoost algorithm. Experimental results on real data indicate the efficacy of the proposed system. The airport and airplane detection rates are higher than 94% and 96%, respectively. Meanwhile, the false alarm rate of airplane detection is superior to 0.05%. Moreover, the whole time cost for handling a large PI is less than 2.5 minutes, which implies that the system is a satisfactory choice for airplane detection in practical applications.

Keywords: Airplane detection, Line segment detector (LSD), Circle frequency filter, Histograms of oriented gradients (HOG), AdaBoost.

1. Introduction

Recently, with remote sensing data of high spatial quality being more easily obtained, new prospect has been opened in field of automatic detection in those images, offering opportunities to detect objects like airports, trees and roads. Among them, airplane detection is an outstanding interesting part for its wide applications. However, although some methods have been proposed for target detection in remote sensing images [13, 15, 21, 24, 25, 26], there are not too many systemic researches on airplane detection for its complexity and sensitiveness. In conventional researches, learning methods are usually applied to airplane detection. Different features of airplane are extracted, and then applied to train classifier and detect airplanes. Li et al. [16] proposed an airplane detection approach based on visual saliency computation and symmetry. Bo et al. [1] used shape values and shape features to detect airplanes.

Clearly, the above learning methods have some limitations for the detection in modern remote sensing images. Firstly, the conventional detection methods usually use only one simple feature, which is effective for small images with relatively simpler backgrounds. However, the modern remote sensing images have more complicated scenes, making traditional methods face with difficulties when handling large remote sensing images. A single feature is not capable to classify airplanes from backgrounds, thus resulting in a large quantity of false alarms. Secondly, to locate airplanes in an image, traversal pixels of the image is usually applied in conventional methods, meaning that nearly every pixel of the image should be checked whether it is target. However, for a large remote sensing image (usually larger than 100000×10000 pixels), the process of traversal usually brings large com-

putation complexity that people cannot tolerate. Therefore, if we directly apply the above airplane detection algorithms to these images, false alarm rate would be high and the time cost is intolerable. Inspired by these discussion, it could be concluded that, to accelerate the detection process in large images, we should first locate candidates of airplanes. Specifically, airport area, where airplanes usually appears, should be first classified. Note that, for a large remote sensing image with high spatial resolution (usually superior to 1m), airport only occupies relatively smaller area of whole image. Therefore, the airport detection could effectively reduce the time cost for airplane detection.

In fact, different airport detection methods have been proposed and they can be roughly classified into two groups: one is built on image segmentation [21] and the other is based on edge detection [3]. The former makes use of image segmentation and extracts regions of interest (ROIs), and the latter puts focus on the runway detection because it is the most remarkable feature of an airport. Traditional detection methods like hough transform, canny edge detector and sobel detector, have been used to detect runways. The methods proposed by Pi et al. [18] and Gan et al. [12] detect the edges of runway and segment airport by using region growing algorithm [14].

To our knowledge, although airport detection is a preprocessing of airplane detection, the two correlative detection tasks were individually discussed, and few papers nor researches have been proposed to accomplish the tasks simultaneously. Therefore, in this paper, we propose a practical automated airplane detection system for large panchromatic image (PI) as illustrated in Fig. 1. In the system, airport and airplane detections are synthetically considered. In our work, instead of conventional edge detection method, we first use Line Segment Detector (LSD) [23] to effectively locate the airport regions. Then an airplane detection algorithm is proposed by combining circle-frequency filter (CFF) [20] and Histograms of Oriented Gradients (HOG) [17, 27], where CFF is used to quickly locate airplane candidates and HOG is used to finally validate airplane locations.

The paper is organized as follows: In Section 2, airport detection based on LSD is discussed. In Section 3, we introduce the airplane detection method based on combining CFF and HOG in detail. In Section 4, numerical experiments on the real world data are discussed. Finally, the paper closes with conclusion in Section 5.

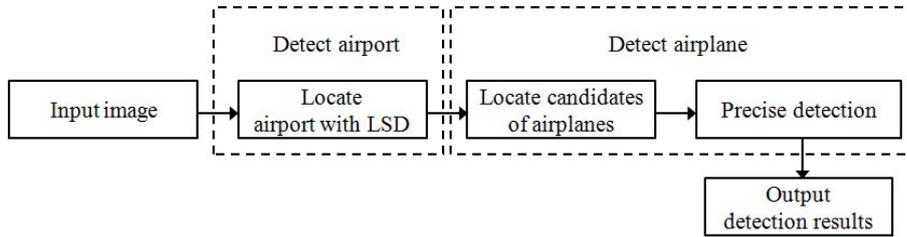


Figure 1: Proposed system for airplane detection.

2. Airport detection based on LSD

As discussed before, airport detection aims at locating airport area, thus narrowing the regions of airplane candidates. In the paper, we propose an airport detection method based on detecting runway with LSD algorithm as illustrated in Fig. 2. In the method, LSD is first applied to the downsampled image, and generates substantial line segments of different objects. Then, based on two strategies, we pay attention to line segment connecting for LSD usually obtains lots of fragmented line segments in intersections. Finally, we accumulate adjacent parallel line segments with similar orientations, and the area with most parallel line segments is just the obtained airport region. Details will be displayed in the following sections.

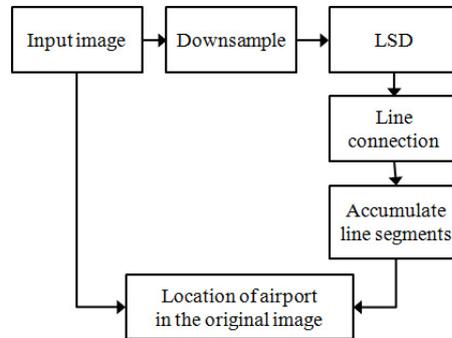


Figure 2: Process of detecting airport.

2.1. Brief introduction of LSD

LSD makes full use of pixel gradient orientation to detect line segments in an image. Firstly, pixels that share the similar gradient angles are gathered into potential line areas (also named line support regions), then a validation

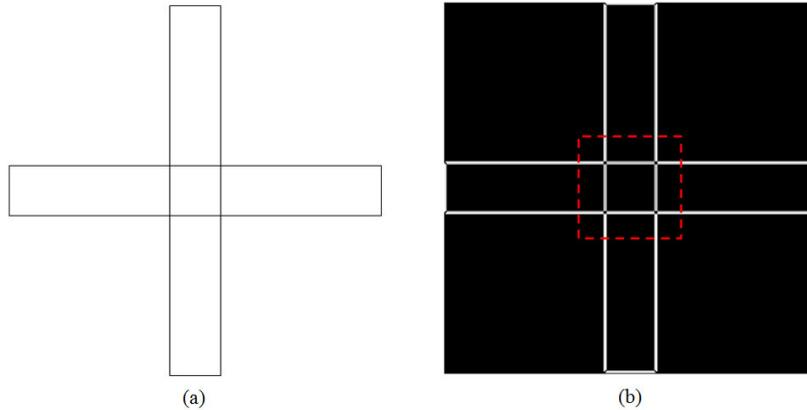


Figure 3: An example of LSD. (a) Test image for LSD. (b) Result after applying LSD.

step based on the *a-contrario* approach [7, 19] and the Helmholtz principle [8, 9] is implemented to find line segments of the image. Thus, the algorithm has three major steps:

1. Group pixels of image to line support regions in which pixels share similar gradient orientation within a specific tolerance angle;
2. Find a line segment that best approximates line support regions;
3. Verify each line segment based on *a-contrario* model.

Fig. 3 illustrates a simple example of LSD. From Fig. 3(b), we see that the line segments of Fig. 3(a) are effectively detected by applying LSD. Note that a tolerance angle (in step 1) of 22.5° is claimed to give the best result in the original LSD. In our case, runways in airports are always strictly straight whereas other objects like roofs and rivers are not, implying that the tolerance angle for airport detection could be much lower. Therefore, in our case, we reduce the tolerance angle, and test a range of angles from 5° to 22.5° . The results show that angles between 5° and 12.5° are satisfying.

2.2. Airport region locating based on LSD

As illustrated in Fig. 2, process for the airport detection has three major parts: Application of LSD, line segment connection and parallel line segment accumulation.

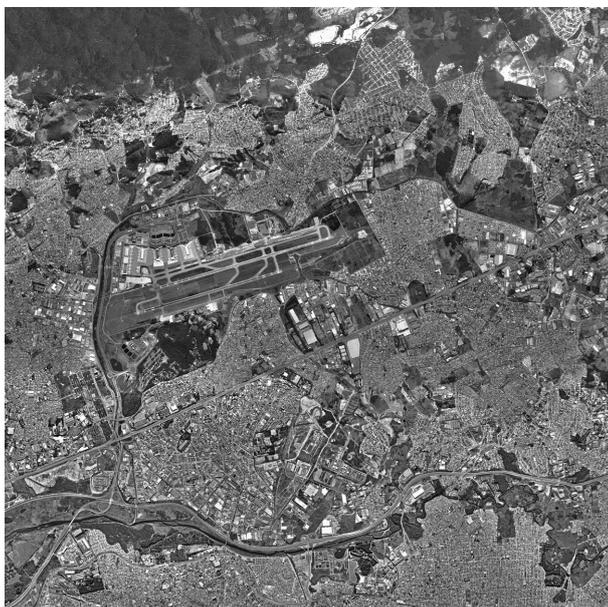


Figure 4: Original remote sensing image.



Figure 5: The line segments of Fig. 4 after applying LSD.

2.2.1. Application of LSD

For a PI with spatial resolution 1m as shown in Fig. 4, we first down-sample it by 10 times. By setting the tolerance angle 5° , we apply LSD to the downsampled image, thus obtaining line segments as shown in Fig. 5. Clearly, the line segments of runway (as shown in the top rectangle) are much more outstanding compared with those in other areas (shown in the bottom rectangle). Note that LSD is based on clustering pixels with similar gradient information. However, pixels in intersectional areas have quite different gradient information, thus causing the discontinuousness of line segments in these areas. This could be easily demonstrated in Fig. 3(b). Meanwhile, in the top rectangle of Fig. 5, fragmented line segments also confirm the above conclusion. Since the discontinuous line segments of runways are inevitable, and they are easily confused with other short line segments, connecting these adjacent fragmented line segments of runway is critical before validating the airport area.

2.2.2. Line segment connecting

Line segment connecting aims at making line segments of runways continuous. To accomplish the task, two strategies should be satisfied simultaneously. They are proposed from computational geometry and listed as follows:

1. The line segments are nearly collinear or parallel.
2. The line segments are close to each other.

According to strategy 1, we first obtain slopes of all the line segments. Then the line segments share similar slope values are potential line segments to be connected. As shown in Fig. 6, if we take **AB** as our reference, then **GH** is obviously excluded for γ (the slope of **GH**) has significant difference from α (the slope of **AB**). Line segments **CD** and **EF** are preserved for their slopes β and δ are all close to α .

Strategy 2 implies that, the distance of two line segments to be connected are required to be within a proper value. Here, the "proper value" have two meanings: The distance from one midpoint of a line segment to the other is less than a threshold which is usually quite small; Distance between midpoints of the two line segments should be less than a threshold. As illustrated in Fig. 7, if **AB** is the reference line segment, then **GH** should be

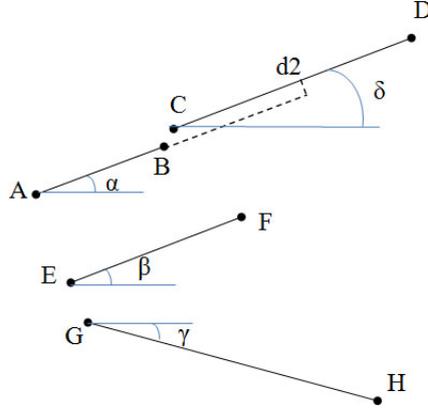


Figure 6: Strategy 1 for line segment connecting.

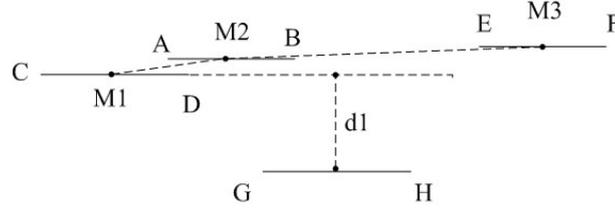


Figure 7: Strategy 2 for line segment connecting.

excluded for the distance $d1$ is too large, and \mathbf{EF} is also excluded because the length $\overline{M2M3}$ is too large. However, \mathbf{CD} is preserved for the length $\overline{M1M2}$ is enough small, and it should be connected to the line segment \mathbf{AB} . Therefore, strategy 1 guarantees that the connected line segments have similar directions, and strategy 2 guarantees that the connected line segments are enough close.

To connect \mathbf{AB} and \mathbf{CD} , the length of \overline{AC} , \overline{AD} , \overline{BC} and \overline{BD} are all first calculated. Then two points with the maximum length are connected. So \mathbf{AB} and \mathbf{CD} could be connected into one line segment \mathbf{BC} . Note that the connecting process is iterative, implying that, after connecting \mathbf{AB} and \mathbf{CD} , \mathbf{BC} will be used as a new line segment and maybe connected with other line segments. However, \mathbf{AB} and \mathbf{CD} are labeled and will not be used again.

With the proposed strategies, we could implement the line segment connecting process and obtain the result as illustrated in Fig. 8. Subscene (the airport area) is shown in Fig. 9. Obviously, the line segments are effectively



Figure 8: Result of whole image after line segment connecting.

connected and this will be helpful for the subsequent detection process.

2.2.3. Parallel line segment accumulating

From Fig. 8, it is clear that line segments in the airport regions are characterized by following factors:

1. Line segments in the airport area are parallel,
2. Line segments in the airport area are longer than those in other areas,
3. The airport area has denser and more line segments, whereas other areas like roads or rivers usually have only two line segments.

Based on the characteristics, if we accumulate adjacent parallel line segments with region growth, then we would obtain regions with different number of line segments. Among the regions, airport area has longer, denser and more lines segments than other areas. Therefore, we calculate total length of all parallel lines in each region, then the region with maximum length is just the airport area.

The accumulating process starts from randomly choosing a seed line segment, then any adjacent line segment that has similar orientation to the seed

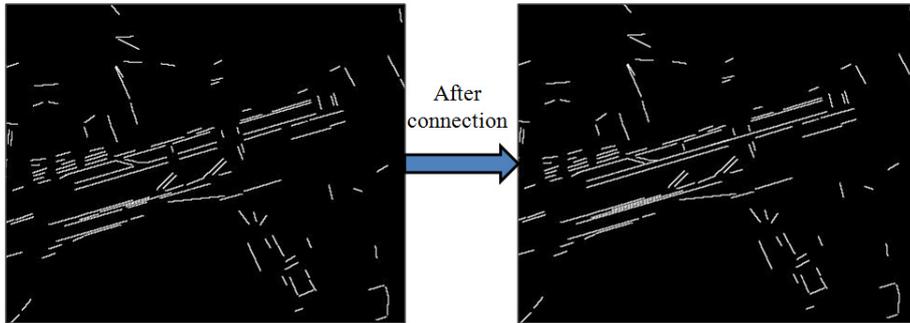


Figure 9: Result of subsene after line segment connecting.

line segment will be added to the region of seed line segment. This step is repeated until no line segments can be added. Finally, the region that has the densest line segments is chosen as airport region. The accumulating result is shown in Fig. 10. Note that, to reduce time cost, we choose a few longest line segments as seeds instead of starting from every line segment. After the accumulation, we could easily obtain the pixel coordinates of airport area in the original image by multiplying sampling ratio (in our case, it is 10), thus locating the airport region as shown in Fig. 11.

3. Airplane detection

After accurately obtaining airport region, we focus on airplane detection in this section. Although different features, like corners and lines have been used to detect airplanes [24, 25], most of methods exploit only partial characteristics of airplanes and would face problems when handling PI with complex scenes. Therefore, by synthesizing different features, we propose an airplane detection method based on CFF and HOG feature. The proposed method contains two main steps as shown in Fig. 12: First, we quickly obtain airplane candidates by using CFF. Then the trained classifier, Adaboost is used to validate airplanes with HOG descriptor.

3.1. Airplane candidate locating with circle-frequency filter

CFF is first proposed and successfully applied to face detection. In our work, it is used to quickly obtain the potential regions of airplanes. Obviously, an airplane usually has two wings and a long fuselage as shown in Fig. 13, and it is also symmetrical to the fuselage. Therefore, if we extract an array of pixels along a circle with a proper center (the center of airplane) and a proper



Figure 10: Result after line segment accumulating.



Figure 11: Result of airport detection .

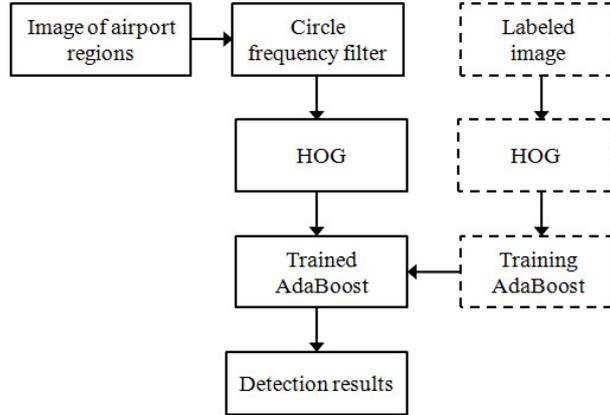


Figure 12: Process of airplane detection.

radius (the red circle as shown in Fig. 13), then the array approximates to a sine curve with period 4 as shown in the bottom row. Four peaks and valleys of the curve coincide with pixels of airplane and background, respectively. Clearly, the periodicity only happens when the proper circle is chosen, which means, proper center and proper radius are chosen. Then we could calculate the fourier transform amplitude of the array, and the amplitude with the largest value labels the candidate of airplane.

Considering a pixel $I(i,j)$ in an image, $f_k(k = 0, 1, \dots, N - 1)$ represents a circle array around the pixel with the radius r , where N represents the length of array, and r is smaller than half length of wingspan and larger than half width of fuselage. Then we easily obtain the amplitude of its Fourier transform [6] response by using the following equation:

$$F(i,j) = \left(\sum_{k=0}^{N-1} f_k \cos \frac{c\pi k}{N} \right)^2 + \left(\sum_{k=0}^{N-1} f_k \sin \frac{c\pi k}{N} \right)^2 \quad (1)$$

where c is a constant that represents the period in Fourier transform and it coincides with the period of array. As discussed before, the period of array is 4. Thus, the constant c is set 8 for the amplitude $F(i,j)$ is maximum only in this circumstance. Therefore, we obtain the whole CFF response of Fig. 11 as shown in Fig. 14.

From Fig. 14, we see that CFF has rapidly located the airplane candidates for the response of airplanes are stronger than other places like runways and flat roads. However, large number of background regions, like edges

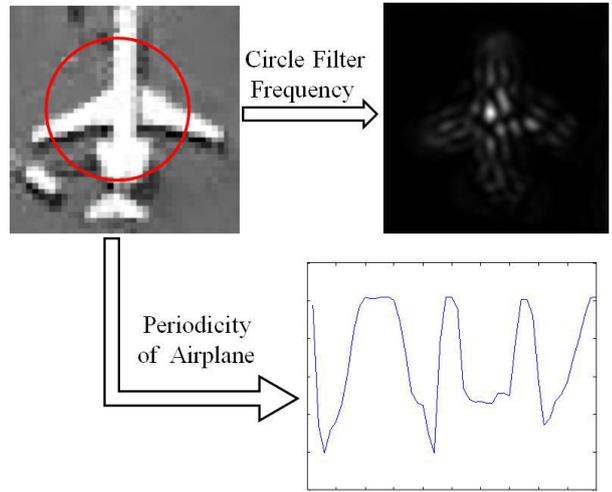


Figure 13: The periodicity of airplane and its CFF result.

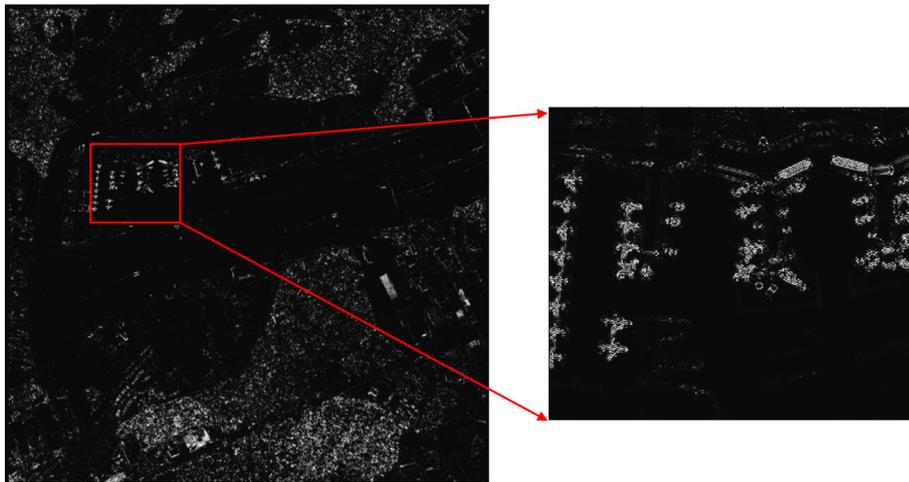


Figure 14: The circle frequency response of Fig. 11.

of buildings and corners of streets, also have strong contrast and confuse the detection results. To validate true result from the amount of potential airplane points, learning algorithm will be used to finally locate airplanes in the next section.

3.2. Precise detection based on HOG descriptor and AdaBoost

In this section, a more precise detection work based on HOG and AdaBoost is needed, thus validating the airplane candidates obtained by applying CFF. HOG descriptor was first proposed in 2005 and used in the pedestrian detection. Triggs [17] and Zhu et al. [27] extended the use of histograms with a dense scan approach. AdaBoost is also an effective learning algorithm to combine a set of simple weak classifiers and form a strong classifier with weighted majority vote. Compared with support vector machine (SVM) [5, 4], AdaBoost provides strong bounds on generalization and guarantees comparable performance. Therefore, in our system, the above two methods are applied to finally obtain airplanes. First, HOG descriptors of the labeled image patches are extracted, then they are used to train the classifier Adaboost. After that, the trained classifier is used to airplane candidates, and we could finally locate airplanes precisely by using region grow to cluster nearby points.

3.2.1. HOG descriptors and its application

HOG descriptor exploits gradient information and local shape information. In the original work for calculating HOG, each normalized sliding window with size 64×128 was divided into cells of size 8×8 pixels, and each group of 2×2 cells was integrated into a block with an overlap of one cell in both horizontal and vertical directions. Then, a nine-bin HOG in each cell is constructed, while each block contained a concatenated vector. Therefore, each block was represented by a 36-D feature vector that was normalized to an L2 unit length and each sliding window was represented by 7×15 blocks, thus forming a feature vector with 3780 dimensions.

In our work, we compute the HOG descriptor in a similar way but with a reference image window size of 40×40 pixels, reference block size of 16×16 pixels, reference cell size of 8×8 pixels. Therefore, in each block, we have 2×2 cells and features with 36 dimensions. The calculation will go through the whole window with step width of 8 pixels. So in each image window, we obtain HOG descriptor with 576 dimensions, and the whole steps for calculating HOG descriptor is shown as follows:

1. Compute the horizontal and vertical gradient of the image by Sobel filters.
2. Compute both the magnitude and orientation of the gradient.
3. For each block, split the block into 2×2 cells.
4. Compute a nine-bin histogram for each cell.
5. Normalize the histograms within a block of 2×2 cells.
6. Group all the normalized histograms into a single vector with $2 \times 2 \times 9$ dimensions.
7. Regard the block as a window, go through the whole image window, re-group all the histograms into a vector with 36×16 dimensions.

To calculate HOG descriptor, a database including 8000 samples is established as shown in Fig. 15. 3000 positive samples (shown in Fig. 15(a)) and 5000 negative samples (shown in Fig. 15(b)) are contained. Moreover, positive samples are collected from world-wide airport with fuselage angle ranges from 0° to 315° . Negative samples are randomly collected with a variety of backgrounds. In our work, HOG descriptors extracted from the above samples will be applied to train the AdaBoost in the next section.

3.2.2. Application of Adaboost

Given training database with positive and negative samples, the classifier, AdaBoost [10, 11] could be trained. Classification And Regression Tree (CART) [2] is employed as the weak learners, and detailed training procedure is shown in the following Algorithm 1:

Here, the iteration number $T = 30$.

As discussed before, the potential airplane regions could be obtained by applying CFF and the trained classifier will be directly used to validate the airplane candidates. We first calculate HOG descriptors of all pixels in the potential airplane region. Then the AdaBoost classifier is used to eliminate false points and obtain pixels of airplanes. Finally, we cluster adjacent pixels by region growing and accomplish the detection task.

In conclusion, the whole system for airplane detection has been established as shown in Fig. 16. Obviously, airplane candidate location is critical for a large PI with complex scenes. Therefore, LSD is first applied to find

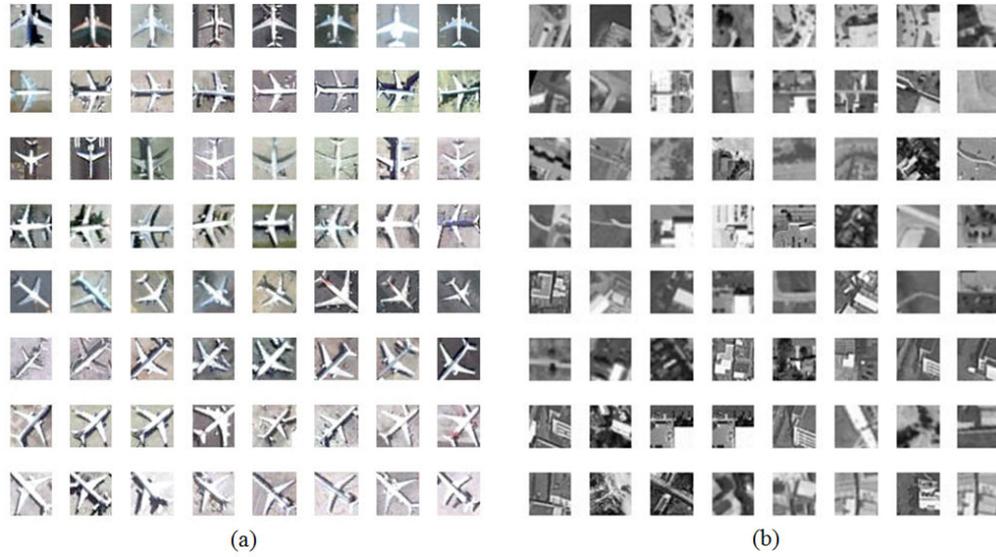


Figure 15: Part of training database. (a) Part of positive images. (b) Part of negative images.

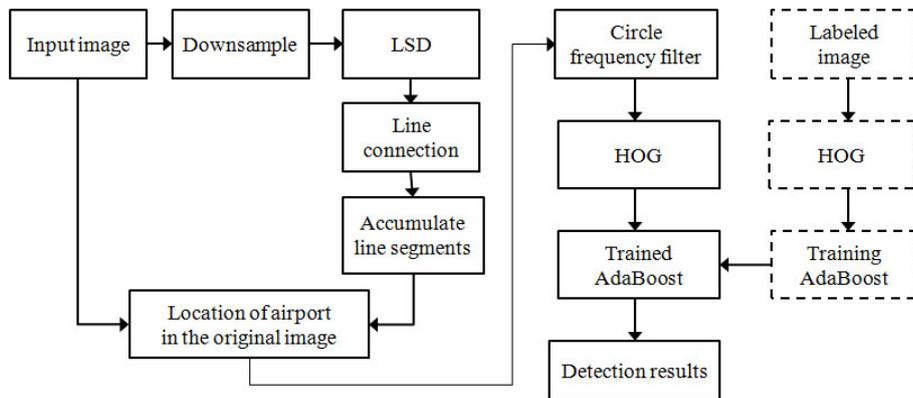


Figure 16: The whole process for airplane detection.

Algorithm 1 Training process of AdaBoost.

- 1: Given training data set $(x_1, y_1), \dots, (x_m, y_m)$, where $(x_i \in X, y_i \in Y = -1, +1)$, m is the total number of the training data set.
- 2: Initialize the weight distribution $D_t(i) = 1/m(1 \leq i \leq m)$.
- 3: Main Iteration: For $t = 1, \dots, T$
 - train the decision tree CARTs with the lowest error.
 - Calculate the error of classification results by the following expression:

$$\varepsilon_t = \sum_{i: h_t(x_i) \neq y_i} D_t(i) \quad (2)$$

where $h_t : X \rightarrow \{-1, +1\}$

- Update the weight distribution by the following expression:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \\ &= \frac{D_t(i)}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \end{aligned} \quad (3)$$

where

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right) \quad (4)$$

and Z_t is a normalized factor.

- 4: The final strong classifier obtained is:

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right) \quad (5)$$

airport in the proposed system, thus reducing large computation complexity. Then CFF is used to further narrow the number of airplane candidates. Therefore, the pre-processing for airplanes reduces a large amount of calculations. After that, HOG and AdaBoost are combined to effectively finally validate airplanes. In the next section, experiments on the real world data are implemented to confirm the effectiveness of the proposed system.

4. Numerical Experiments

The proposed system is implemented on about 50 remote sensing images which are obtained from Google Earth. All the images have sizes range from 13000×13000 pixels to 17000×17000 pixels, and the spatial resolution is 1m. The system is run under C/C++ program, and the computer has an Intel Core 3.2 GHz CPU, 4GB memory.

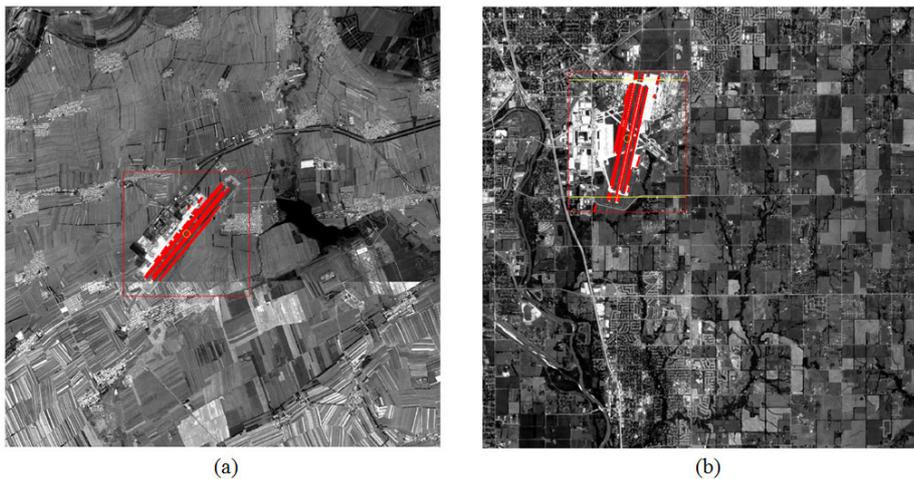


Figure 17: Two results of airport detection in our method.

In the first experiment, airport detection module of the system is tested. Fig. 17 shows two detection results, and both airports are correctly located. Note that, the airports in Fig. 17 have low contrasts, implying that the proposed method is effective in images with complex scenes. Some other methods based on other edge detection methods are also implemented and one result by using hough transform is shown in Fig. 18. We see that the obtained result is wrongly located because more interfering lines are obtained though hough transform. Moreover, detection results with different methods

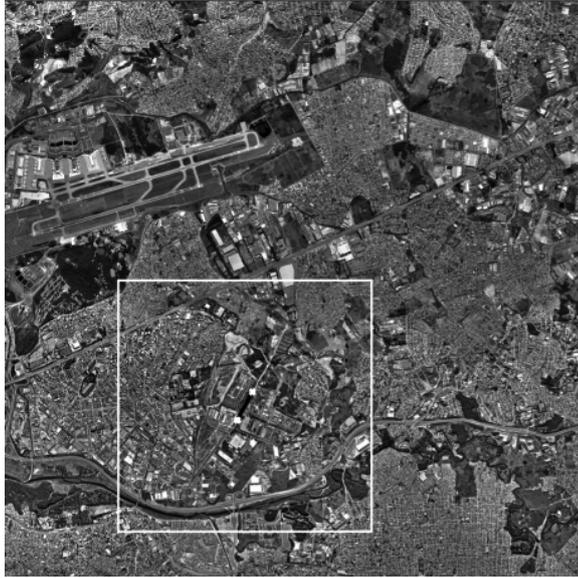


Figure 18: Result of airport detection with hough transform.

are listed in Tab. 1 and the detection rate in Tab. 1 is calculated by the following equation:

$$\text{Detection Rate} = \frac{\text{Number of detected airports}}{\text{Number of total airports}} \quad (6)$$

Clearly, our method costs least time and obtains the best detection results for the detection rate of our method is 94%, whereas other methods are much lower.

Table 1: Comparison of runway detection results between different methods

Method	Detection Rate	Average Time
Ours	94%	15.2s
Sobel	60.78%	20.5s
Prewitt	66.67%	20.2s
Robert	49.02%	21.2s
Canny	25.49%	16.5s

In experiment two, airplane detection module is implemented on the images with the above located airport regions, and all image sizes are 5000

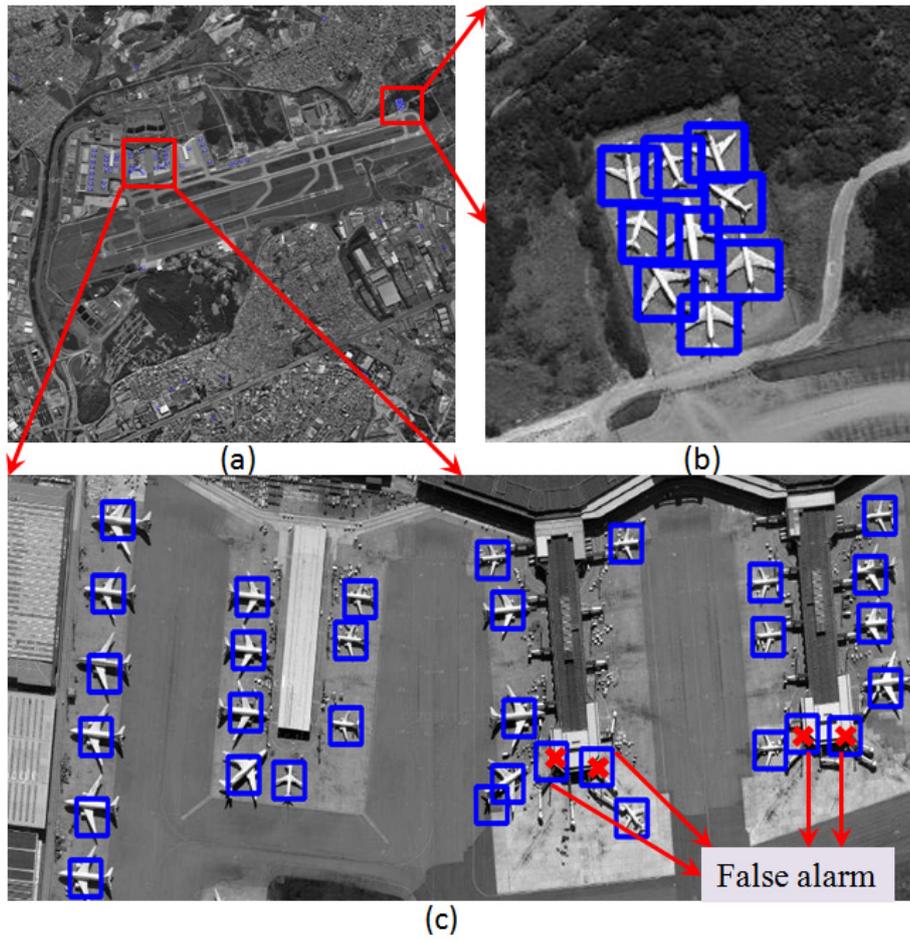


Figure 19: One of the final detection result. (a) The whole detection result of image. (b)-(c) Subscenes of detection results.

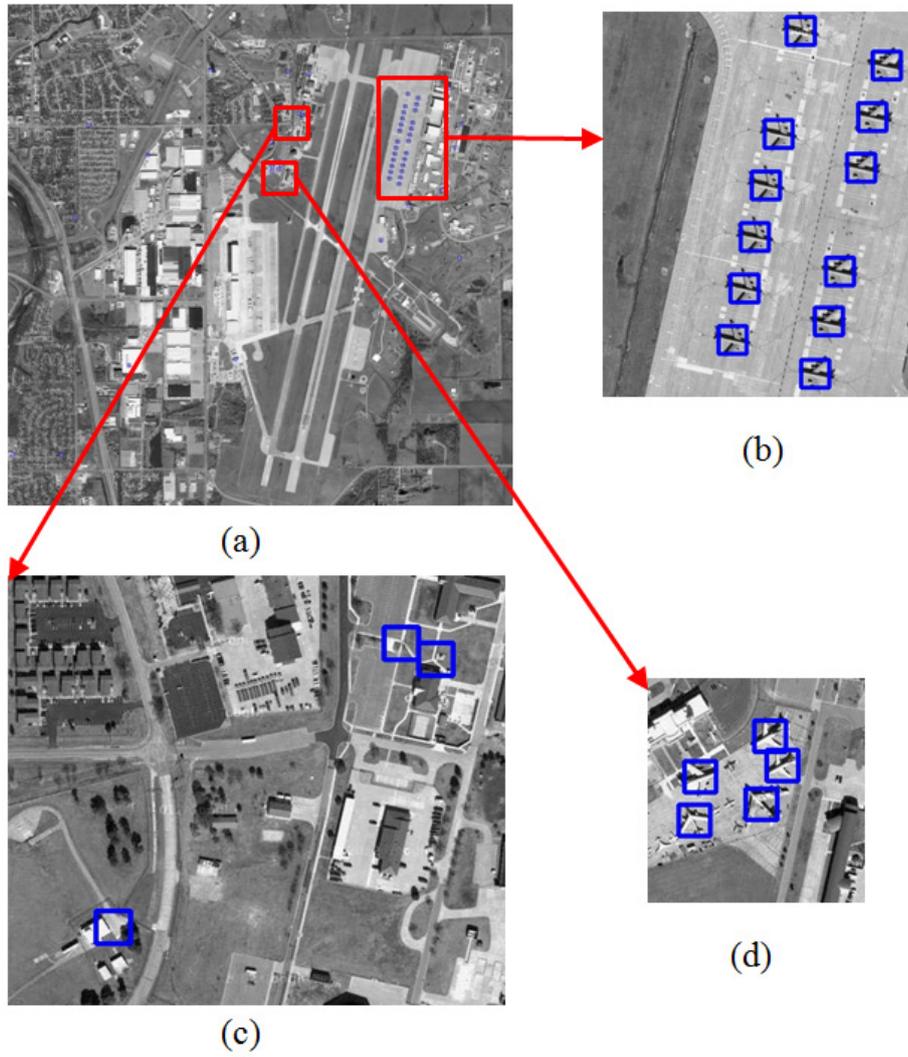


Figure 20: One of the airplane detection result. (a) The whole detection result of image. (b)-(d) Subscenes of the detection results.

$\times 5000$ pixels. The final detection airplanes are marked with rectangles as shown in Fig. 19 and Fig. 20. Fig. 19(a) and Fig. 20(a) are the detection results of whole images, Fig. 19(b)-(c) and Fig. 20(b)-(d) are the subscenes of detection results. In Fig. 19(b), nine airplanes are close to each other with different directions. However, all of them are correctly located without false alarm, indicating that the proposed method performs well for airplane detection. In Fig. 19(c), although four false alarms happen, we obtain all the airplanes, which means that the proposed system also works well for PIs with complex scenes. Since airplanes and backgrounds have high contrast in the parking apron, no false alarm happens and all airplanes are accurately detected in Fig. 20(b). Note that the proposed system also works well in the area with low contrast (in Fig. 20(d)) because the HOG descriptor helps to eliminate most backgrounds like roofs and roads. As discussed before, the training data for AdaBoost contains airplanes with eight directions. So the airplanes in Fig. 20(d) with different directions are also detected. However, In Fig. 20(c), there are three false alarms with the similar shape as airplanes, implying that false alarms are still inevitable in areas with too complex backgrounds. In order to quantitatively assess the method, two metrics including true-positive (TP) rate and false-positive (FP) rate are defined as follows:

$$\text{TP rate} = \frac{\text{Correctly detected airplanes}}{\text{Total number of airplanes}} \quad (7)$$

$$\text{FP rate} = \frac{\text{Pixel number of falsely detected airplanes}}{\text{Pixels number of whole image}} \quad (8)$$

In our experiments, the total number of airplanes is 296 and 285 of them are correctly detected. So the TP rate is superior to 96% and the FP rate is superior to 0.043%. It is also worth mentioning that, average time cost for the whole detection including airport detection and airplane detection modules is less than 2.5 minutes, indicating that the proposed system is a practical approach for the detection task in large PI.

5. Conclusions

This paper proposes an automated airplane detection system for the large PI with high spatial resolution. Two main modules which utilize characteristics of objects are contained in the system. Two fast algorithms, LSD and CFF, are applied in our system to respectively locate airport and airplane,

thus narrowing the airplane candidates from the complex backgrounds and helping to reduce a lot of unnecessary calculations. Finally, HOG descriptor and AdaBoost are combined to validate the airplanes. Experiments with real world data demonstrate the efficacy of the proposed system with the TP rate is more than 96%, and the FP rate is less than 0.05%. Moreover, the average time cost is less than 2.5 minutes. Therefore, the proposed detection system is a good choice when handling large PIs with high spatial resolution and complex scenes.

References

- [1] S. Bo and Y. Jing, Region-based airplane detection in remotely sensed imagery, 2010 3rd International Congress Image and Signal Processing (CISP), (2010) 1923-1926.
- [2] L. Brieman, F. Jerome, J. Charles and R. A. Olshen, Classification and regression trees, first ed. Chapman and Hall/CRC, (1984).
- [3] J. B. Burns, A. R. Hanson and E. M. Riseman, Extracting straight lines, IEEE Transactions on Pattern Analysis and Machine Intelligence 8(4) (1986) 425-455.
- [4] O. Chapelle, P. Haffner and V. N. Vapnik, Support vector machines for histogrambased image classification, IEEE Transactions on Neural Networks 10(5) (1999) 1055-1064.
- [5] C. Cortes and V. N. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273-297.
- [6] I. Daubechies, The wavelet transform, time-frequency localization and signal analysis, IEEE Transactions on Information Theory 36(5) (1990) 961-1005.
- [7] A. Desolneux, S. Ladjal, L. Moisan and J.-M. Morel, Dequantizing image orientation, IEEE Transactions on Image Processing 11(10) (2002) 1129-1140.
- [8] A. Desolneux, L. Moisan and J.-M. More, A grouping principle and four applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 25(4) (2003) 508-513.

- [9] A. Desolneux, L. Moisan and J. M. Morel, From gestalt theory to image analysis, a probabilistic approach, Springer (2008).
- [10] Y. Freund, Boosting a weak learning algorithm by majority, Information and Computation 121(2) (1995) 256-285.
- [11] Y. Freund, I. Raj, E. S. Robert and S. Yoram, An efficient boosting algorithm for combining preferences, The Journal of Machine Learning Research 4 (2003) 933-969.
- [12] B. Gan, X. Q. Wu and Y. J. Hu, Application of hough transform in segmentation of airdrome remote sensing images based on statistics, Computer Engineering 28(8) (2002) 264-265.
- [13] L. H. He and L. Carin, Airport detection in large aerial optical imagery, IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings. (ICASSP '04), (2004) 761-764.
- [14] A. Huertas, W. Cole and R. Nevatia, Detect runways in complex airport scenes, Computer Vision, Graphics, and Image Processing 51(2) (1990) 107-145.
- [15] Z. Lei, T. Fang, H. Huo and D. R. Li, Rotation-invariant object detection of remotely sensed images based on texton forest and hough voting, IEEE Transactions on geoscience and remote sensing 50(4) (2012) 1206-1217.
- [16] W. Li, S. M. Xiang, H. B. Wang and C. H. Pan, 2011. Robust airplane detection in satellite images. Image Processing (ICIP), 2011 18th IEEE International Conference on (2011) 2821-2824.
- [17] D. Navneet and T. Bill, Histograms of oriented gradients for human detection, 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2005) 886-893.
- [18] Y. M. Pi and X. B. Yang, Airport detection and runway recognition in SAR images, International Geoscience and Remote Sensing Symposium (IGARSS) '03. Proceedings (2003) 4007-4009.
- [19] A. Robin, L. Moisan and S. Le, An *a – contrario* approach for subpixel change detection in satellite imagery. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(11) (2010) 1977-1993.

- [20] K. Shinjiro and T. Nobuji, Circle-frequency filter and its application. IWAIT 2001, Int. Workshop on Advanced Image Technology (2001) 217-222.
- [21] C. Tao, Y. H. Tan, H. J. Cai, and J. W. Tian, Airport detection from large IKONOS images using clustered SIFT keypoints and region information, *IEEE geoscience and remote sensing letters* 8(1) (2011) 128-132.
- [22] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, 2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2001) 511-518.
- [23] R. G. 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) (2010) 722-732.
- [24] C. Wang, Z. Shan and X. L. Hu, Airplane detection in remote sensing image with a circle-frequency filter, *Proc. SPIE, International Conference on Space Information Technology* (2005) 1-6.
- [25] J. Yao and Z. F. Zhang, Semi-supervised learning based object detection in aerial imagery, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005) 1011-1016.
- [26] H. Zheng, X. M. Hu, X. S. Si and W. B. Yang, A novel object detection approach for satellite imagery based on danger theory, *International Conference on Intelligent Networks and Intelligent Systems* (2008) 445-448.
- [27] Q. Zhu, M.-C. Yeh, K.-T. Cheng and S. Avidan, Fast human detection using a cascade of histograms of oriented gradients, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2006) 1491-1498.