AIRPORT DETECTION IN LARGE AERIAL OPTICAL IMAGERY

Dehong Liu, Lihan He and Lawrence Carin

Department of Electrical and Computer Engineering
Duke University, Durham, NC 27708-0291

ABSTRACT

A method to detect airports in large aerial optical imagery is considered. Combining texture segmentation and shape detection, this method shows advantages in analyzing large aerial imagery. First, large aerial images are segmented and interpreted according to textural features using a fast kernel matching pursuits (KMP) algorithm. As a result, attention is then paid on small regions of interest, extracted from the large images. Second, for each region of interest, a corresponding binary image is generated via the Canny edge operator, yielding a modified Hough transform image with which we search for elongated rectangles with desired dimensions (characteristic of runways). Those detected rectangles are declared as runways and the corresponding region of interest as an airport. Application in a dozen aerial images from southern California demonstrates the effectiveness of the algorithm.

1. INTRODUCTION

Target detection and recognition has been a research topic in computer vision for many years. An airport, as a key transportation target, has also attracted much attention. Previous research [1][2] has investigated recognition of airports from aerial imagery, based primarily on small localized areas or small imagery size (256×200). The aim of this paper is to search for airports within cluttered backgrounds in large optical imagery (approximately 6500×7500 pixels).

As we demonstrate below, airports located in dense urban and suburban areas are often difficult to distinguish from the background, composed of roads and buildings. Furthermore, to deal with large aerial imagery of size around 6500×7500 pixels, algorithm design is a significant challenge to meet requirements of computation time and memory.

Fortunately, an airport scene is typically characterized by runways of particular length and width. Such properties distinguish airports from their surrounding environment, and such will play a key role in the algorithm presented here. In a large image it is often difficult to exhaustively search for such airport features. We therefore develop a texture-based pre-screening step that first locates potential regions of interest (ROIs), with these localized regions examined subsequently for the detailed characteristics of an airport.

In this paper, a new airport detection method combining texture segmentation and shape analysis is presented. Large aerial images are first segmented and classified into different textures, with a sparse kernel classifier based on the kernel matching pursuits (KMP) algorithm [3]. According to the texture segmentation results, ROIs are extracted for further shape analysis. To verify the existence of an airport, elongated rectangles are searched in each ROI using a modified Hough transform. The method is demonstrated using a dozen airborne optical images from southern California. The algorithm as well as example results are described in the following sections.

2. TEXTURE SEGMENTATION

Optical aerial imagery may often be interpreted in terms of texturural features. The first stage of our airport detector is based on textural features, with which we seek to delineate those ROIs that are likely to be characteristic of an airport.

2.1. Texture features

Here images are cut into small-sized chips. Each chip is considered as a type of texture depending on its features. For each chip, a total of six features are extracted to form a feature vector. Gray-scale images are considered, and the textural features used are (a) the mean pixel value within the chip, (b) the standard deviation, (c) the mean of the two-dimensional gradient, (d) the standard deviation of the gradient, (e) the Zernike moment [4] and (f) the circular-Mellin coefficient [5].

Specifically, for a chip $I(x,y)$ in a Cartesian coordinate system or $I(\rho,\theta)$ in a polar coordinate system with size $M \times N$, its features can be computed as follows.
Mean:

$$F1 = \frac{1}{M \cdot N} \sum_x \sum_y I(x,y) . \qquad (1)$$

Standard deviation:

$$F2 = \sqrt{\frac{1}{M \cdot N - 1} \sum_x \sum_y \left(I(x,y) - F1 \right)^2} \ . \tag{2}$$

Mean of gradient:

$$F3 = \frac{1}{M \cdot N} \sum_x \sum_y G(x,y) \ , \tag{3}$$

where $G(x,y) = \left| I(x,y) - I(x-1,y) + j(I(x,y) - I(x,y-1)) \right|$.

Standard deviation of gradient:

$$F4 = \sqrt{\frac{1}{M \cdot N - 1} \sum_x \sum_y \left(G(x,y) - F3 \right)^2} \ . \tag{4}$$

Zernike moment [4]:

$$F5 = \frac{n+1}{\pi} \iint_{\rho \le 1} I(\rho,\theta) V^*(\rho,\theta) d\rho d\theta \ , \tag{5}$$

where Zernike polynomials

$V(\rho,\theta) = R(\rho)\exp(jm\theta)$,

$$R(\rho) = \sum_{s=0}^{(n-|m|)/2} \frac{(-1)^s (n-s)!}{s!\left((n+|m|)/2 - s\right)!\left((n-|m|)/2 - s\right)!} \rho^{n-2s} \ .$$

We use $m = n = 1$, and then $V(\rho,\theta) = \rho \exp(j\theta)$.

Circular-Mellin coefficient [5]:

$$F6 = \int_\rho \int_\theta I(\rho,\theta) h^*(\rho,\theta) \exp(2\rho) d\rho d\theta \ , \tag{6}$$

where $h(\rho,\theta) = \exp(-\rho)\exp(j2\pi \cdot p\rho)\exp(jq\theta)$, with $p = q = 1$, and therefore $h(\rho,\theta) = \exp(-\rho)\exp[j(2\pi\rho + \theta)]$.

Thus corresponding to each texture chip, there is a feature vector $\mathbf{F} = [F1, F2, F3, F4, F5, F6]$, which can be classified using a learning-machine classifier.

2.2. KMP Classifier

In order to classify the feature vectors of different texture chips, a kernel machine classifier is commonly used. However, due to the large amount of training and testing data, classifiers such as the support vector machine (SVM) and the relevance vector machine (RVM) will require much time and computer memory, and may be impractical. Here a sparse kernel classifier using the kernel matching pursuits (KMP) algorithm is applied to overcome these difficulties [3].

Suppose the training samples are $\{\mathbf{x}_i, y_i\}_{i=1}^K$ where $\mathbf{x}_i$ is an observed datum (feature vector) and $y_i \in \{1,2,...,L\}$ is its target (texture) label, with the functional relationship $y = f(\mathbf{x},\mathbf{w})$. The aim of kernel machine classifier design is to learn the optimal parameters $\mathbf{w}$ that minimize the expected risk.

Suppose the *n*th order kernel machine function is:

$$f_n(\mathbf{x}) = \sum_{i=1}^n w_{n,i} k(\mathbf{c}_i, \mathbf{x}) + w_{n,0} = \mathbf{w}_n^T \boldsymbol{\phi}_n(\mathbf{x}) \tag{7}$$

where $\boldsymbol{\phi}_n(\cdot) = [1, k(\mathbf{c}_1,\cdot), k(\mathbf{c}_2,\cdot),..., k(\mathbf{c}_n,\cdot)]^T$, $k(\cdot,\cdot)$ is a kernel function, $\mathbf{W}_n$ are the weights that combine the basis functions, and the subscript n denotes the number of basis functions being used.

Then the weighted sum of squared errors between the expected output and the KMP output is

$$e_n = \left(1 \Big/ \sum_{i=1}^K \beta_i\right) \sum_{i=1}^K \beta_i [y_i - \boldsymbol{\phi}_n^T(\mathbf{x}_i)\mathbf{w}_n]^2 \ , \tag{8}$$

where β_i is a constant responsible for quantifying the importance of the *i*th training sample $(\mathbf{x}_i, y_i)$. The value of $\mathbf{w}$ that minimizes (8) is found to be:

$$\mathbf{w}_n = \mathbf{M}_n^{-1} \sum_{i=1}^K \beta_i \boldsymbol{\phi}_n(\mathbf{x}_i) y_i \ , \tag{9}$$

where $\mathbf{M}_n = \sum_{i=1}^K \beta_i \boldsymbol{\phi}_n(\mathbf{x}_i)\boldsymbol{\phi}_n^T(\mathbf{x}_i) \overset{Def}{=} \{\beta_i \boldsymbol{\phi}_{n,i} \boldsymbol{\phi}_{n,i}^T\}_i$.

Then *(n+1)*th order KMP is inductively written as:

$$f_n(\mathbf{x}) = \mathbf{w}_{n+1}^T \boldsymbol{\phi}_{n+1}(\mathbf{x}) = \mathbf{w}_{n+1}^T \begin{bmatrix} \boldsymbol{\phi}_n(\mathbf{x}) \\ \phi_{n+1}(\mathbf{x}) \end{bmatrix} \tag{10}$$

where $\phi_{n+1}(.) = k(\mathbf{c}_{n+1},\cdot)$, and the *(n+1)*th order parameters $\mathbf{W}_{n+1}$ are

$$\mathbf{w}_{n+1} = \begin{bmatrix} \mathbf{w}_n + \mathbf{M}_n^{-1}\{\beta_i \boldsymbol{\phi}_{n,i}\phi_{n+1,i}\}_i b^{-1}[\{\beta_i \boldsymbol{\phi}_{n,i}^T \phi_{n+1,i}\}_i \mathbf{w}_n - \{\beta_i \phi_{n+1,i} y_i\}_i] \\ -b^{-1}\{\beta_i \boldsymbol{\phi}_{n,i}^T \phi_{n+1,i}\}_i \mathbf{w}_n + b^{-1}\{\beta_i \phi_{n+1,i} y_i\}_i \end{bmatrix} \tag{11}$$

where $b = \{\beta_i \phi_{n+1,i}^2\}_i - \{\beta_i \phi_{n+1,i} \boldsymbol{\phi}_{n,i}^T\}_i \mathbf{M}_n^{-1} \{\beta_i \boldsymbol{\phi}_{n,i} \phi_{n+1,i}\}_i$,

$$\mathbf{c}_{n+1} = \mathbf{x}_{i_{n+1}} = \arg\max_{\substack{j \ne i_1,...,i_n \\ 1 \le j \le K}} \delta e(k, \mathbf{x}_j) \ .$$

From (11), the parameter $\mathbf{w}$ can be optimized with an iterative computation. When $(e_{n-1} - e_n)/e_{n-1} < 0.01$ the KMP results are terminated and the parameters $\mathbf{w}$ are obtained to design the KMP classifier (other stopping criteria may be considered).

In the KMP algorithm, after selecting a given training example as a basis, the form of the kernel is optimized with a simple gradient search in the parameters of the kernel (e.g. the variance of a radial basis function kernel). In this manner the kernel is "tuned" to the respective training vector chosen as a basis. We have found that, compared with SVM and RVM, this algorithm achieves good generalization at low computational costs.

Considering the typical textures in aerial imagery, a seven-ary KMP classifier is designed (discussed further when presenting results). With the KMP classifier, chips in large aerial imagery will be labeled as different textures. As a result, those chips not likely to be parts of an airport area (i.e. with the wrong textural properties) are eliminated and ROIs are extracted for further shape analysis.

3. SHAPE DETECTION

After segmentation, attention is then focused on the limited ROIs deemed to be characteristic of airports. Textural segmentation is not adequate by itself to delineate airports, and therefore the ROIs are subjected to further processing. Specifically, it is necessary to carry out shape detection in the ROIs. We have found that detection of runway edges, in the form of an elongated rectangle, is an effective method to verify whether a given ROI is representative of an airport.

3.1. Modified Hough transform

The Hough transform is one of the most popular methods to extract straight lines from digital images. A large number of papers dealing with Hough transform applications may be found, such as [6]. Using the Hough transform, the straight line $y = kx + t$ can be expressed in polar coordinates as:

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta \qquad (12)$$

From (12), if a set of edge points (x_i, y_i) that lie on a straight line having parameters (ρ_0, θ_0), then each edge point plots to a curve in (ρ, θ) space and all these curves must intersect at the point (ρ_0, θ_0). Hence a local maximum in a (ρ, θ) space histogram represents a straight line. This approach achieves good results in most cases. However, in large aerial imagery, extra straight lines are often detected due to the large number of edge points (roads, buildings, etc.).

To reduce the influence of noise points, the increment of a point (x_i, y_j) at the (ρ_0, θ_0) histogram bin is changed from a constant to an absolute inner product $\left|\left\langle \vec{r}_{ij} \cdot \vec{r}_{\theta 0} \right\rangle\right|$, in which $\vec{r}_{ij}$ represents the unit vector with the direction of the local line at point (x_i, y_j) and $\vec{r}_{\theta 0}$ stands for unit vector with the orientation of the line $\rho_0 = x \cdot \cos\theta_0 + y \cdot \sin\theta_0$.

3.2. Elongated rectangle detection

Compared with highways or roads, airport runways show straight and smooth characteristics, with specific dimensions of about 1000~2000 meters long and 25~85 meters wide and thus form an elongated rectangle shape in aerial imagery. Such elongated rectangles are characteristic of runways.

Before performing the Hough transform, a binary image is first created by edge detection for each ROI. Since the gray level variation may be small in some aerial images, the Canny edge operator [7], which can find both strong and weak edges using two thresholds on image gradient, is used such that runway edges can be detected even at relative lower gray contrast.

Using the aforementioned modified Hough transform, a histogram of the binary image is computed. Based on this transform histogram, elongated rectangles are searched in the ROI. First, pairs of long straight parallel lines are detected as the long edges of runways. In the histogram image, each pair of parallel lines corresponds to two local maxima at the same θ, while with a distance $(\rho_1 - \rho_2)$. If the distance is between 25 and 85 meters, then short parallel lines between the two parallel lines with almost orthogonal direction ($\pm 10^\circ$ is allowed in practice) and distance between 1000 and 2000 meters will be detected as the short edges. Then the two pairs of parallel lines form an elongated rectangle representing the location of a runway. If there is not such an elongated rectangle found in a ROI, then the ROI is declared as clutter (not an airport).

4. EXAMPLE RESULTS

Using the algorithm mentioned above, experiments have been carried out on a dozen aerial images from southern California. Seven textures are considered: water, mountains, buildings, urban areas, foliage, fields and runways (airport). Examples of each of these seven textures are taken from distinct imagery, for KMP training. Example results are shown in the following figures, wherein (automatic) classification results are presented. Fig. 1 shows an aerial image, which includes an airport. The image size is 6570×7620 pixels, with one-meter ground resolution. The texture segmentation result of this image is shown in Fig. 2. Two airport-like ROIs are extracted. Further shape analysis shows that one ROI includes an airport while the second does not (Fig. 3). The final algorithm results for two other images are demonstrated in Figs. 4 and 5.

5. CONCLUSIONS

An automatic airport-detection algorithm is proposed for large aerial optical imagery. This method carries out both texture segmentation and shape analysis. It overcomes difficulties in computation time and memory, and obtains good results in finding ROIs and airport runways from large complex imagery. This technique has been carefully tested and validated using a dozen images from southern California.

6. REFERENCES

[1] David M. McKeown, Wilson A. Harvey and John McDermott, "Rule-based interpretation of aerial imagery," IEEE Transactions on Pattern Analysis and Machine Intelligence, 7(5), pp. 570-584, 1985.
[2] Junwei Han, Lei Guo, and Yongsheng Bao, " A method of automatic finding airport runways in aerial images," Proceedings of 2002 6th International Conference on Signal Processing, pp. 26-30, 2002.
[3] P. Vincent and Y. Bengio. "Kernel matching pursuits," Machine Learning, vol. 48, pp. 165-187, 2002.
[4] Alireza Khotanzad and Yaw Hua Hong, "Invariant image recognition by Zernike Moments," IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(5), pp. 489-497, 1990.
[5] G. Ravichandran, M. M. Trivedi, "Circular-Mellin features for texture segmentation," IEEE Transactions on Image Processing, 4(12), pp. 1629-1640, 1995.
[6] V. Venkateswar and Rama Chellappa, "Extraction of straight lines in aerial images," IEEE Transactions on Pattern analysis and Machine Intelligence, 14(11), pp. 1111-1114, 1992.
[7] J. F. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), pp. 679-698, 1986.

Fig. 1. Aerial image of Los Alamitos, California

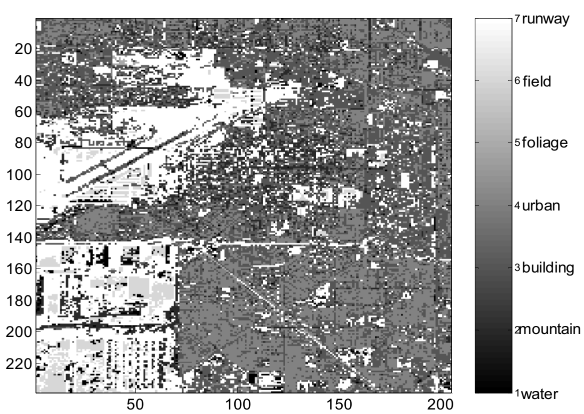

Fig. 2. Texture segmentation result using seven-ary KMP classifier with two ROIs (white)

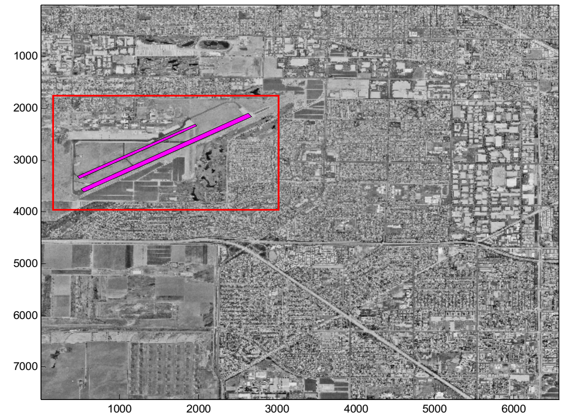

Fig. 3. Airport detection result, large rectangle denotes airport area, two elongated rectangles represent runways

Fig. 4. Airport detection result of aerial image of Camarillo, California

Fig. 5. Airport detection result of aerial image of Whiteman, California