A METHOD TO ENHANCE THE DEEP LEARNING IN AN AERIAL IMAGE

Kuang-Pen Chou[1], Dong-Lin Li[2], Chin-Teng Lin[3] and Wen-Chieh Lin[4]

1&4: Department of Computer Science, National Chiao Tung University, Taiwan
2: Department of Electrical Engineering, National Chiao Tung University, Taiwan
3: Computational Intelligence & Brain Computer Interface Centre, University of Technology Sydney, Australia

ABSTRACT

In this paper, we propose a kind of pre-processing method which can be applied to the deep learning method for the characteristics of aerial image. This method combines the color and spatial information to do the quick background filtering. In addition to increase execution speed, but also to reduce the rate of false positives

***Index Terms*—**drones, aerial image, deep learning, background filter.

1. INTRODUCTION

In recent years, it has become a new trend to monitor the environment through drones, and the use of cameras to capture interesting objects has become one of the most popular research areas in computer vision [1]-[3]. However, High resolution and small targets are not conducive to the traditional algorithm to achieve effective detection. Therefore, many scholars try to use the deep learning algorithm to detect and track [4]-[6] , but the deep learning has high recognition rate is also accompanied by a lot of computing time. The characteristics of the aerial image also cause deep learning to spend a lot of time on handling the background, such as plenty of vegetation, rivers, sky and roads etc. Therefore, in order to avoid this situation, high-speed and effective background filtering is the problem to be must deal with when deep learning applies to aerial image.

Filtering techniques are an important part of image processing systems, in particular when it comes to image enhancement and restoration. Traditionally, there are many ways to filter out the background, such as spatial or frequency filter, color and grayscale estimation, and even through the continuous time to obtain the foreground information, the fastest two of among are use of spatial filter and color transform. Spatial filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. According to this feature, many studies have been proposed to estimate whether the texture of a particular region of the image is meaningful, or for background [7]-[10] .

Color Transform is mainly used in face detection, such as [11][12]. It projects the image into HSV, YCbCr, or other 2-dimensional or 3-dimensional color space, and then use Bayesian rules or some classification method to detect face pixels. In [13], Rojas and Crisman proposed a method, which can project the road pixel onto a color plane, and can thus detect vehicles from road background.

This paper is organized as follows: Section 2 presents the principle of color transform. Section 3 presents how to filter most background by our proposed pre-processing. Experimental result is presented in Section 4 along with accuracy discussion followed by conclusions in Section 5

2. COLOR TRANSFORM

Our research mainly use the color transform method, proposed by Tsai et al.[14]. First step is to collect many images of the detected objects, such as in [13][15], they collected N images of freeway and parking lots. In [16], Ohta et al. used the concept of Karhunen-Loève Transform [17], “ The eigenvector of the original data covariance matrix is used as the transformation matrix to reduce the relevance of the data, leaving a representative information”. He calculated the covariance matrix Σ of the RGB value from these images, and derive the eigenvectors e_i and eigenvalues λ_i of Σ , where $i = 1,2,3$ The three color features are then generated.

$$C_i = e_i^r R + e_i^g G + e_i^b B \quad \text{for } i = 1,2,3 \tag{1}$$

where $e_i = \left(e_i^r, e_i^g, e_i^b\right)$. Ohta et al. use this method to perform region segmentation, and found that the color feature C_1 with respect to the largest eigenvalue is the transform equation for grey level

$$C_1 = \frac{1}{3}R + \frac{1}{3}G + \frac{1}{3}B \tag{2}$$

And other two feature C_2 and C_3 are orthogonal to C_1,

$$C_2 = \frac{R - B}{2} \tag{3}$$

and

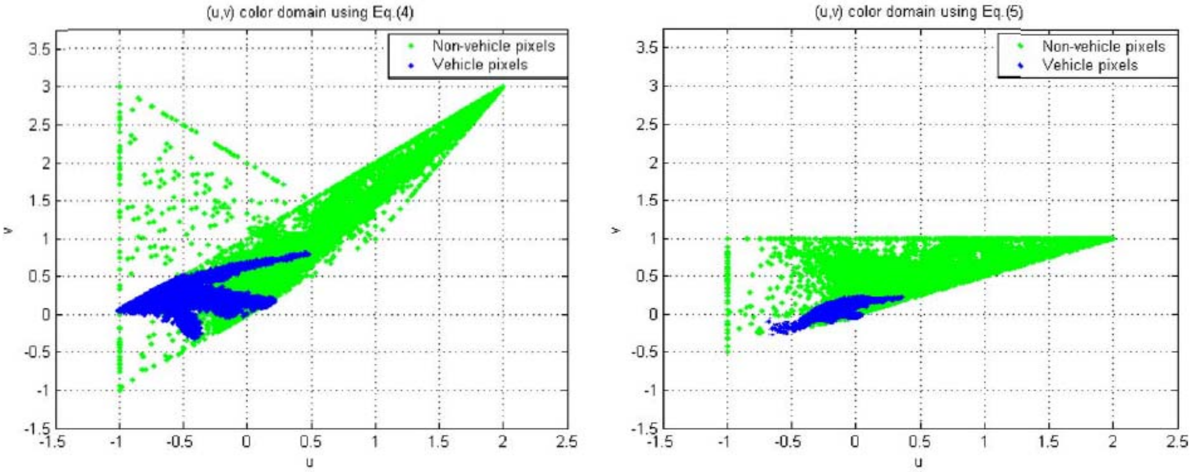

Fig. 1 Distribution of vehicle and nonvehicle color projected into (u,v) domain
(In the left is the color domain of equation (5) and (6), in the right is the color domain of equation (7) and (8))

$$C_3 = \frac{2G - R - B}{4} \tag{4}$$

In [13], Rojas et al. used the same method, and found that the color distribution of road will be centered on the axis of the (2) vector. Therefore, if the color of all the road is projected to the plane, which is orthogonal to (2), these colors will focus on a small circle on this plane.

In [11], Tsai et al. used this method to find out the color of vehicles. The color plane (u,v) of vehicles is orthogonal to the axis extended by (1/3,1/3,1/3), which are

$$u_p = \frac{2Z_p - G_p - B_p}{Z_p} \tag{5}$$

and

$$v_p = \max\left\{\frac{B_p - G_p}{Z_p}, \frac{R_p - B_p}{Z_p}\right\} \tag{6}$$

where $\left(R_p, G_p, B_p\right)$ is the color of pixel p and $Z_p = \left(R_p, G_p, B_p\right)/3$ is used for normalization. However, in the actual operation, if *R,G,B* three components are calculated separately, each of them would be easily affected by noise, and generate some false alarm. Therefore, Tsai et al. convert the B_p and R_p in equation (5) and (6) to reduce false alarm. And the equation becomes

$$u_p = \frac{2Z_p - G_p - B_p}{Z_p} \tag{7}$$

and

$$v_p = \max\left\{\frac{Z_p - G_p}{Z_p}, \frac{Z_p - B_p}{Z_p}\right\} \tag{8}$$

From Fiq. 1, we can find out that the vehicle pixels converted by equation (7) and (8) is more concentrated than the vehicle pixels converted by equation (5) and (6).

After projecting the color into (u,v) domain, Bayesian rule is used to determine whether a pixel is vehicle or not. Assume that RGB value in (u,v) domain forms a multivariate Gaussian distribution, where m_v and m_n are the mean value of vehicle and nonvehicle pixel in (u,v) domain, and Σ_v and Σ_n are their covariance matrix. Then, given a pixel x, the probability belonging to a vehicle pixel is

$$p\left(x \mid \text{vehicle}\right) = \frac{1}{2\pi\sqrt{\left|\Sigma_v\right|}} e^{-d_v(x)} \tag{9}$$

where

$$d_v = \left(1/2\right)\left(x - m_v\right)\sum\nolimits_v^{-1}\left(x - m_v\right)^T \tag{10}$$

Similarly, the pixel x belonging to a nonvehicle pixel is

$$p\left(x \mid \text{non-vehicle}\right) = \frac{1}{2\pi\sqrt{\left|\Sigma_n\right|}} e^{-d_n(x)} \tag{11}$$

where

$$d_v = \left(1/2\right)\left(x - m_n\right)\sum\nolimits_v^{-1}\left(x - m_n\right)^T \tag{12}$$

According to Bayesian rule, if

$$p\left(x \mid \text{vehicle}\right) > p\left(x \mid \text{non-vehicle}\right) \tag{13}$$

then pixel x can be determine as vehicle, and equation (2-8) can be replaced by

$$\begin{aligned} &p\left(x \mid \text{vehicle}\right) P\left(\text{vehicle}\right) > \\ &p\left(x \mid \text{non-vehicle}\right) P\left(\text{non-vehicle}\right) \end{aligned} \tag{14}$$

$P\left(\text{vehicle}\right)$ and $P\left(\text{non-vehicle}\right)$ are the initial probability of vehicle and nonvehicle. Plugging (9) and (11) into (13) and take its log form, then the classification equation is: if

$$d_n\left(x\right) - d_v\left(x\right) > \lambda \tag{15}$$

Then pixel x is vehicle, where

(a) (b)

Fig. 2 Training sample of color transform (a) Tree images (b) Non-tree images

$$\lambda = \log\left[\sqrt{\frac{|\Sigma_v|}{|\Sigma_n|}}\left(\frac{P(\text{non-vehicle})}{P(\text{vehicle})}\right)\right] \tag{16}$$

3. PROPOSED PRE-PROCESSING

3.1 Color Transform

In this research, we use the same color transform method as in [14]. We cut 3390 tree images and 5796 non-tree images from CBCL StreetScenes Challenge Framework dataset. In Fig. 2 are some training samples of color transform. To reduce the amount of data and noise, we set the size of all images as 35 x 35 and calculate the statistics of their RGB value. After that, use the following equation to calculate its covariance matrix

$$\Sigma = \begin{bmatrix} E((R-\mu_R)(R-\mu_R)) & E((R-\mu_R)(G-\mu_G)) & E((R-\mu_R)(B-\mu_B)) \\ E((G-\mu_G)(R-\mu_R)) & E((G-\mu_G)(G-\mu_G)) & \cdots E((G-\mu_G)(B-\mu_B)) \\ E((B-\mu_B)(R-\mu_R)) & E((B-\mu_B)(G-\mu_G)) & E((B-\mu_B)(B-\mu_B)) \end{bmatrix} \tag{17}$$

where μ is the average value of this color, and E() is the calculation of expected value. Then, calculate the eigenvector e_i and eigenvalue λ_i of Σ. The result has shown that the eigenvector with respect to the largest eigenvalue is the same as mentioned in Section 2, which is (0.33, 0.33, 0.33). And the other 2 eigenvector are (0.78, -0.6, -0.18) and (0.24, 0.56, -0.8), both of which are orthogonal to the first eigenvector. We therefore derive the transform equation

$$u_p = \frac{0.78R_p - 0.6G_p - 0.18B_p}{Z_p} \tag{18}$$

and

$$v_p = \frac{8B_p - 2.4R_p - 0.56G_p}{Z_p} \tag{19}$$

This transform equation can be used to transform the RGB value of each pixel p into (u,v) domain, where $Z_p = (R_p + G_p + B_p)/3$ is used for normalization. Mentioned in [14], the noise of the original image would affect the accuracy of judgment in actual operation, which is the noise would increase if RGB 3 color components are separately used than together. Therefore, we also replace the first component in both equation into grey level Z_P, and derive the new transform equation

$$u_p = \frac{0.78Z_p - 0.6G_p - 0.18B_p}{Z_p} \tag{20}$$

And

$$v_p = \frac{8Z_p - 2.4R_p - 0.56G_p}{Z_p} \tag{21}$$

As in Fig. 3, the color of tree pixels would gather in a smaller area than the color of non-tree pixels. Then, we use Bayesian rules as in [14] to determine whether each pixel is tree or not. Assume that the RGB component forms a Gaussian distribution in (u,v) domain, m_t and m_{nt} are its average value of tree and non-tree pixels in (u,v) domain, Σ_t and Σ_{nt} are their covariance matrix. Then, we define the probability of a pixel x belonging to a tree pixel as a normal distribution

$$p(x \mid \text{tree}) = \frac{1}{2\pi\sqrt{|\Sigma_t|}} e^{-d_t(x)} \tag{22}$$

where

$$d_t = \left(\tfrac{1}{2}\right)(x - m_t)\textstyle\sum_t^{-1}(x - m_t)^T \tag{23}$$

Similarly, its probability belonging to a non-tree pixel is

$$p(x \mid \text{non-tree}) = \frac{1}{2\pi\sqrt{|\Sigma_{nt}|}} e^{-d_{nt}(x)} \tag{24}$$

where

$$d_t = \left(\tfrac{1}{2}\right)(x - m_{nt})\textstyle\sum_{nt}^{-1}(x - m_{nt})^T \tag{25}$$

After the derivation of, equation (26)

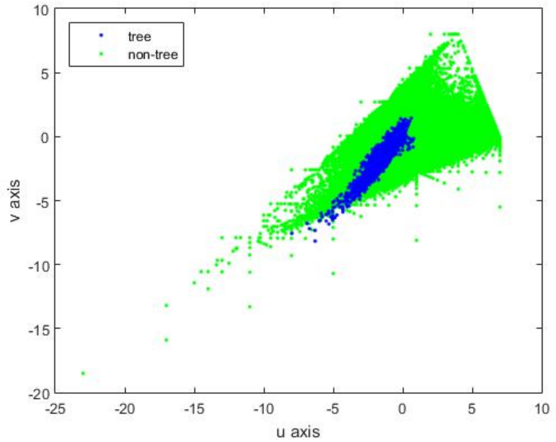

Fig. 3 Color distribution of tree and non-tree pixels in (u,v) domain

$$d_{nt}(x) - d_t(x) > \lambda \quad (26)$$

can be used to determine whether pixel x is tree, where

$$\lambda = \log\left[\sqrt{\frac{|\Sigma_t|}{|\Sigma_{nt}|}}\left(\frac{P(\text{non-tree})}{P(\text{tree})}\right)\right] \quad (27)$$

$P(\text{tree})$ and $P(\text{non-tree})$ are tree and non-tree initial probability.

3.2 Smooth Area Detection

To filter out road and sky these background, we collect road and sky images from CBCL StreetScenes Challenge Framework dataset [18], and calculate the statistic of the variance distribution in each images, as in equation (28)

$$\text{var} = \frac{\sum(X - \mu_X)^2}{area} \quad (28)$$

According to the statistic, we found that the variance of these images mainly concentrate close to zero. We also calculate the vehicle variance distribution in this dataset [18], and the result shows that the variance is very widely distributed, but the minimum value is around 500. Therefore, we take this minimum as our threshold, and use sliding window to scan through the whole image. If the variance is smaller than the threshold, it is smooth area, and would be filtered out. The result is shown in Fig. 3. While calculating the variance in each image, we use integral image [19] to speed up the process, and avoid repeated calculation.

4. EXPERIMENTAL RESULT

In this section, we show the experimental result on the real aerial images form a drone. The testing samples a total of 2307 aerial images to four different scenes. Fig 4. shows the results was preprocessed by our proposed method. It can clearly see a lot of vegetation, road and ocean and other smooth background has been filtered. The overall filtration ratio was 71.4%. This means that regardless of which deep learning algorithm is used, we proposed pre-processing has been effectively reduced to be processed image area.

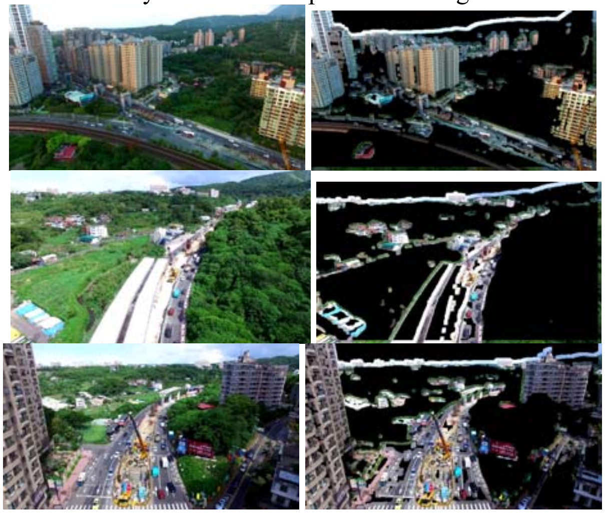

Fig. 4 the results was preprocessed by our proposed method

5. CONCLUSIONS

In this paper, a pre-processing framework was introduced which can work well in most scenes. Although the adoption of high-resolution images in the application of surveillance can reserve more image details, the computation cost will be increased largely. In summary, the proposed method can filter most non-information areas to save computation power.

6. REFERENCES

[1] A. Capolupo, S. Pindozzi, C. Okello, N. Fiorentino and L. Boccia, “Photogrammetry for environmental monitoring: The use of drones and hydrological models for detection of soil contaminated by copper” *Science of The Total Environment,* Vol 514, pp. 298-306, 2015

[2] D J. Busset, F. Perrodin, P. Wellig, B. Ott, K. Heutschi, T. Rühl and T. Nussbaumer “etection and tracking of drones using advanced acoustic cameras” *SPIE Security+ Defence. International Society for Optics and Photonic*s, 2015.

[3] P Nguyen, M Ravindranatha and A Nguyen “Investigating cost-effective rf-based detection of drones” Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use, Singapore, June 26, 2016

[4] T-Y Lin, Y. Cui, S. Belongie and J. Hays "Learning deep representations for ground-to-aerial geolocalization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[5] L. Zhang, X. L. Zhang and D. Bo "Deep learning for remote sensing data: A technical tutorial on the state of the art." *IEEE Geoscience and Remote Sensing Magazine* Vol.4 pp. 22-40, 2016.

[6] G.L. Oliveira, A Valada, C Bollen, W. Burgard and T. Brox, "Deep learning for human part discovery in images." Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 2016.

[7] H. B Barlow "Temporal and spatial summation in human vision at different background intensities." *The Journal of physiology* vol.141 pp 337-350, 1958.

[8] A.V. Lugt "Signal detection by complex spatial filtering." *IEEE Transactions on information theory* vol.10 pp. 139-145, 1964.

[9] G.A. Swartzlander "Peering into darkness with a vortex spatial filter." *Optics Letters* vol.26, pp 497-499, 2001.

[10] J.S. Kirar and R.K. Agrawal "Composite kernel support vector machine based performance enhancement of brain computer interface in conjunction with spatial filter." *Biomedical Signal Processing and Control* vol.33 pp. 151-160, (2017.

[11] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis", IEEE Transactions on Multimedia, vol. 1, no. 3, pp. 264-277, 1999.

[12] J. Kovac, P. Peer and F. Solina, "Human skin color clustering for face detection", The IEEE Region 8 EUROCON 2003. Computer as a Tool., 2003.

[13] J. Rojas and J. Crisman, "Vehicle detection in color images", Proceedings of Conference on Intelligent Transportation Systems, 1997.

[14] L. Tsai, J. Hsieh and K. Fan, "Vehicle Detection Using Normalized Color and Edge Map", IEEE Transactions on Image Processing, vol. 16, no. 3, pp. 850-864, 2007.

[15] G. Healey, "Segmenting images using normalized color", IEEE Transactions on Systems, Man, and Cybernetics, vol. 22, no. 1, pp. 64-73, 1992.

[16] Y. Ohta, T. Kanade and T. Sakai, "Color information for region segmentation", Computer Graphics and Image Processing, vol. 13, no. 3, pp. 222-241, 1980.

[17] K. Karhunen, Über lineare Methoden in der Wahrscheinlichkeitsrechnung, vol. 37, 1947.

[18] S. Bileschi, " CBCL streetscenes challenge framework ", Cbcl.mit.edu, 2007. Available: http://cbcl.mit.edu/software-datasets/streetscenes/.

[19] P. Viola and M. Jones, "Robust Real-Time Face Detection", International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, 2004.