Automatic Detection of Roads using Aerial Photographs and Calculation of the Optimal Overflight Route for a Fixed-wing Drone

Miguel Pérez P, Holman Montiel A, Fredy Martínez S
Facultad Tecnológica, Universidad Distrital Francisco José de Caldas
Bogotá, Colombia

***Abstract*—Currently, fixed-wing drones have become indispensable tools for the surveillance of large areas of land, justified by their better cost/benefit ratio, great flight autonomy, and payload capacity. In particular, the identification of roads, traffic control, monitoring of wear on asphalt layers, risk identification, and safety improvement are applications that are being implemented in these unmanned aerial vehicles. Tracking a road requires systems capable of detecting artificial marks through images employing aerial photographs that allow the implementation of optimal overflight routes. This research work presents a solution to the problem of road tracking from aerial photographs and implements an image processing algorithm and morphological techniques that calculate and traces the ideal route for the drone to track automatically, regardless of its orientation and the type of road.**

***Keywords*—Automatic road tracking; decorrelation stretching; aerial imagery; optimal overflight route calculation; UAV**

I. Introduction

Transportation infrastructure, and in particular roads, are of vital importance for the growth and development of a country, and even more so for a country like Colombia, since 80% of the country's cargo is moved by road [1]. Surveillance of these important roads is normally done through simple methods such as cameras or with the public forces located in different segments of these roads. These methods are costly due to the many elements or people that must be available [2], [3]. Currently, the need to implement new, less expensive methods for surveillance has forced researchers to investigate systems that use fixed-wing drones for this task [4]. Such systems take advantage of the great autonomy and high payload capacity of these drones [5] for the process of monitoring and surveillance of roads. This monitoring process cannot be teleoperated due to the extension of the territories to be monitored; for this reason, research is aimed at autonomous systems that use digital image processing algorithms, obtained by the Unmanned Aerial Vehicles (UAV) themselves; to automatically detect the road and thus be able to extract elements of interest autonomously [6], [7].

Asphalt roads can be considered extraneous or artificial features that extend for many kilometers in a natural landscape [8], [9]. A process used to highlight these landmarks in an image is called decorrelation stretching (DCS) [10], [11], [12]. The DCS technique is complemented with color detection [13], [14] and image filtering [15] algorithms; which extract the violet areas corresponding to the road, leaving it in a binarized image. It should be clarified that the images used in this research are aerial photographs of roads taken by fixed-wing drone overflight; where there are great differences between them (altitude taken from the photo, direction, light conditions and brightness, topography, landscape, etc.) which demonstrates the robustness and efficiency of the developed system. Once the road is detected, the system can calculate the extreme points [16] and trace the direction of this obtaining the optimal route of overflight of the UAV [17] that allows following the road for many km.

To recapitulate, the proposed system incorporates a decorrelation stretching algorithm and filtering and color detection algorithms to automatically detect a road in an aerial image. In addition, it proposes an algorithm that allows computing the optimal overflight path of the detected road by implementing the endpoints of the detected road. These contributions are described in detail in the following sections, which are organized as follows: Sections II and III present the general definitions and methodology used respectively, and section IV presents the results obtained.

II. Methodology

This research contributes to a joint operation of decorrelation stretching algorithms (DCS), image filtering, and color detection algorithms together with image processing libraries developed entirely in MATLAB. They allow detecting and calculating a flyover route for roads using aerial images. A description of each of the elements used is presented below.

A. Decorrelation Stretching (DCS)

Decorrelation stretching is a process that improves the color differences found in an image by removing the inter-channel correlation found in the image pixels. Decorrelation stretched images provide an overview that improves spectral reflectance variations; that is, it improves color differences in images with high inter-channel correlation [11], [12].

Therefore, using DCS is useful for intensifying the color difference between roads and the surrounding landscape in an image, which facilitates the detection of roads. Fig. 1 shows an image of a road subjected to the decorrelation stretching process implemented in this research; in it, you can see the color difference between Fig. 1 (a) and (b), in addition, you can see the violet hue that takes the road to be detected.

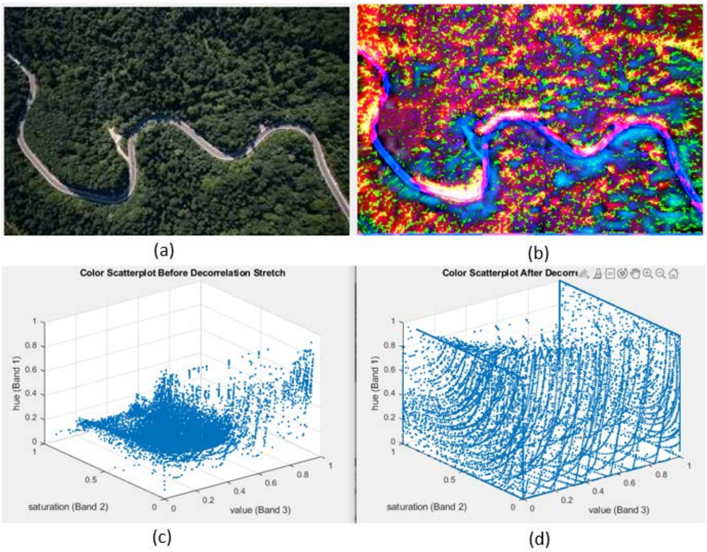

Fig. 1. (a) Aerial View of the Road from Stock Photo 2024801798 [18], (b) Image Processed by the Decorrelation Stretching Algorithm, (c) Point Spread Cloud of the Original Image, (d) Point Spread Cloud of the Resulting Image.

The decorrelation stretching is a linear operation in pixels in which the parameters depend on the values of the original image statistics and the target image and can be expressed using the following equations [19].

$$b = T * (a - m) + m\_target \quad (1)$$

a and b are vectors of n bands times 1, T is the linear transformation matrix and m contains the mean of each band in the image, m_target are vectors of n bands times 1 containing the desired output mean in each band. The linear transformation matrix T is calculated by performing a proper decomposition of the correlation matrix as follows:

$$Corr = inv(SIGMA) * Cov * inv(SIGMA) \quad (2)$$

$$S(k,k) = \frac{1}{\sqrt{LAMBDA(k,k)}} \quad (3)$$

Where $S(k,k)$ is a diagonal matrix containing the stretching factors for each band; finally, the matrix T used in (1), is calculated through:

$$T = SIGMA\_target\ V\ S\ V'\ inv(SIGMA) \quad (4)$$

Although the decorrelation stretching process is usually implemented in RGB format images [20]; the operations proposed here can be applied to any multiplane matrix. Therefore, the algorithm can be applied to images in other formats. For this research the image was processed in HSV format; since it takes advantage of the high stability of this format to the changes of brightness and luminosity in the image; which benefits the detection [21].

In Fig. 1 (c) and (d), you can see the cloud of points representing the dispersion of pixel values in both the original and the resulting image; image (c) is much less dispersed than (d); which demonstrates the color differences between the two images, eliminating the high correlation between channels.

B. Color Detection and Image Filtering Algorithm

Color detection is a technique used to separate objects containing colors of interest into the different existing color spaces; it is important to clarify that color spaces are nothing more than a representation of how machines understand the existing color in the real world [22]. An image in RGB format is made up of three matrices representing the level of red, green, and blue in the image; therefore, the combinations of these three matrices give rise to a wide range of colors ranging from black to white. A color segmentation algorithm reads the three matrices that make up the image and extracts from them the pixels containing the target color; usually, the output of this algorithm is a single plane or binarized matrix which contains in white the pixels corresponding to the target color and in black the other pixels; which are considered as the background of the image (see Algorithm1).

Algorithm 1. Color Detection

```
Start Variables;
create an array of zeros the size of the image
For a=1: up to the number of rows
    For b=1: up to the number of columns
        If (Image (a,b,R)> threshold)
            If (Image (a,b,B)> threshold)
                If (Image (a,b,G)< threshold)
                    transform into 1 pixel with the target color
                End If
            End If
        End If
    End For
End For
```

Algorithm 1 shows the implemented color detection; in this algorithm, it is observed that the function goes through the color image and transforms the pixels corresponding to the target color. This procedure is achieved through a function that defines the transformation threshold, see Fig. 2.

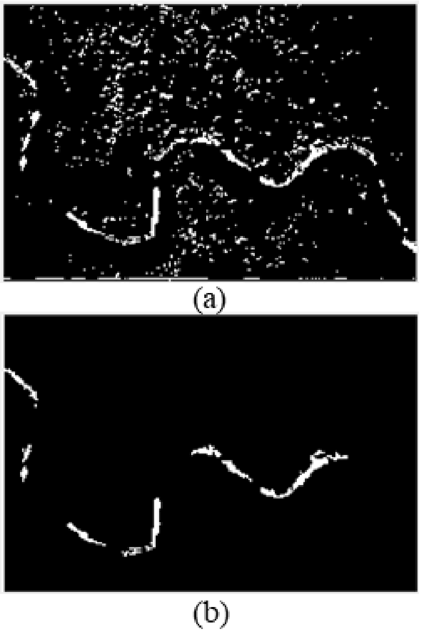

Fig. 2. (a) Binarized Image Employing the Color Detection Algorithm, (b) Filtered Image.

From Fig. 2(a) the need for the image filtering algorithm is evident. The process of image filtering is that which removes unwanted objects or noise from the original image. There are innumerable image filtering methods or functions [23]; but one of the most widely used is bwareaopen (BinaryImage, minPixels) object size filtering. This function removes objects from a binarized image that contain less than the minimum number of pixels specified by minPixels; such a filtering process finds the connected components of the binarized image and calculates the area (in pixels) for further removal. Another filter implemented is the imfill function (BW, locations), which performs a flood fill operation on the background pixels, and is used to eliminate noise in the detected objects. It should be noted that this function removes all black objects surrounded by white elements in the image. Fig. 2 (b) shows the result of the image filtering algorithms implemented in this research.

III. Implementation

The developed system is composed of two main operational blocks: the first block oversees detecting and extracting the existing roads in the processed image; this block makes use of the technique of stretching by decorrelation and conventional image filtering, see Fig. 3. The second block aims to develop an algorithm that calculates the optimal overflight route for a UAV to follow, this algorithm will be explained in full in the next section and is a creation entirely developed by the authors of this research.

A. *Overflight Route Calculation for Autonomous Navigation*

Calculating the overflight route that allows autonomous navigation for extended periods is essential in video surveillance systems [24] [25]; this calculation must draw a straight line between the start and end points to maximize the capabilities of the fixed-wing drone and minimize its fuel consumption. The problem is that most roads are not straight (see Fig. 4 (a)), therefore the system must calculate the orientation (horizontal, vertical, diagonal, etc.) and direction of the road to calculate an overflight route (in a straight line) that allows overflying the entire road.

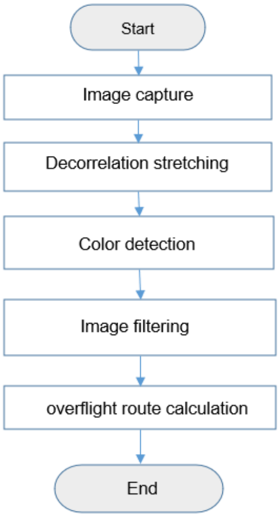

Fig. 3. Block Diagram of the Developed System.

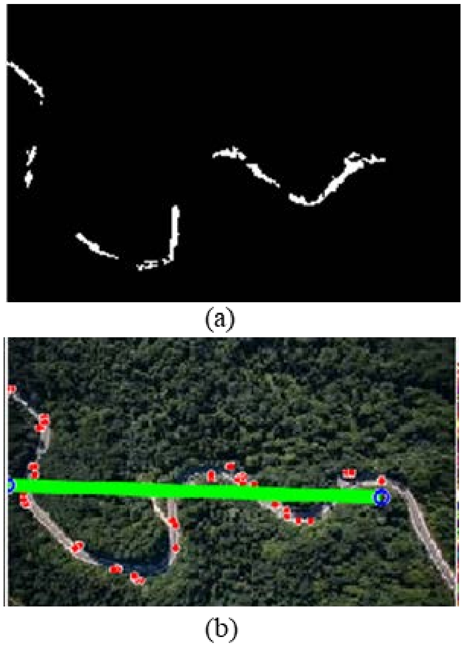

Fig. 4. (a) Road Detected by the Algorithm, (b) Automatic AUV Overflight Route Calculation.

Fig. 4(a) shows the result of the road detection algorithm. Here it can be seen that the road is not detected as a single element; but as multiple elements that are added together to give the totality of the road. These interruptions in the continuity are due to external elements that prevent its detection and with which the route calculation algorithm must deal. The proposed overflight path calculation algorithm starts by identifying the endpoints of each element, which can be seen in red in Fig. 4(b). The function in charge of calculating the endpoints by morphological processing is called regionprops (ImageBina,'Extreme') [26]. Then the image is divided into four equal quadrants, and it is calculated which of these extreme points for each quadrant is higher, lower, left, and right than the others; which determines the orientation and direction of the road. Once these points have been calculated, the start and end point of the route is calculated as the midpoint between these points, see Algorithm 2.

Algorithm 2. Navigation route calculation

```
Start Variables;
for a=1: up to the number of items detected
calculations of the extreme points of each element
 if (the element is the highest)
                save highest_item
 else if (is the lowest)
              save lowest_item
 else if (is the most to the left)
              save left_item
 else if (it is the most to the right)
              save right_item
 end if
end for
%%% horizontal or vertical direction calculation

if (highest_item < left_item)
 fI=fminI+(fmaxI-fminI)/2  % top point
else
 fD=fminD+(fmaxD-fminD)/2 % left point
end if
if (lowest_item > right_item)
 cU=cminU+(cmaxU-cminU)/2 % down point
else
 cD=cminD+(cmaxD-cminD)/2 % right point
end if
paint line on the detected points
```

IV. Results

The algorithm was implemented on a computer with Windows 10 Pro 64bits operating system, Intel(R) Core (TM) i3-3110M CPU 2.40GHz (4 CPUs), RAM 12288MB, and MATLAB 2020b Update 5 under license from Universidad Distrital Francisco José de Caldas. The implemented system operates pixel by pixel on the input matrix. Table I shows the time report; here it can be seen that the time that contributes most to the system depends on the number of elements that were detected; since the algorithm for calculating extreme points and calculating the route must be applied individually to these elements.

The automatic road detection and overflight route calculation system was tested with different images taken by fixed-wing drone overflight. These images were taken with different cameras at different altitudes and different times of the day to demonstrate the robustness of the algorithm.

Fig. 5 shows the response of the developed system; it is observed that the system can detect and calculate the overflight path for roads going vertically, horizontally, or diagonally; demonstrates the effectiveness of the algorithm for images with different illuminations; double-lane roads, and cars. Finally, demonstrate what happens to the algorithm if it encounters intersections, elevated bridges, round-point, and other elements that are usually present on the road.

As the algorithm is designed to run on a mini pc on board the aircraft, the validation of operation was performed using different processing units. Fig. 6 shows the response time of the algorithm on different processors; it should be noted that these times were measured using the same image in which five elements were obtained after binarization.

TABLE I. SYSTEM PROCESSING TIME MEASUREMENTS

Time Report [seconds]		
Image size in pixels	**Number of Detected Objects**	**Time in Seconds**
33.000	3	0.2488
50.246	1	0.1113
50.246	2	0.3273
50.325	3	0.2526
50.400	1	0.0977
50.400	3	0.4041

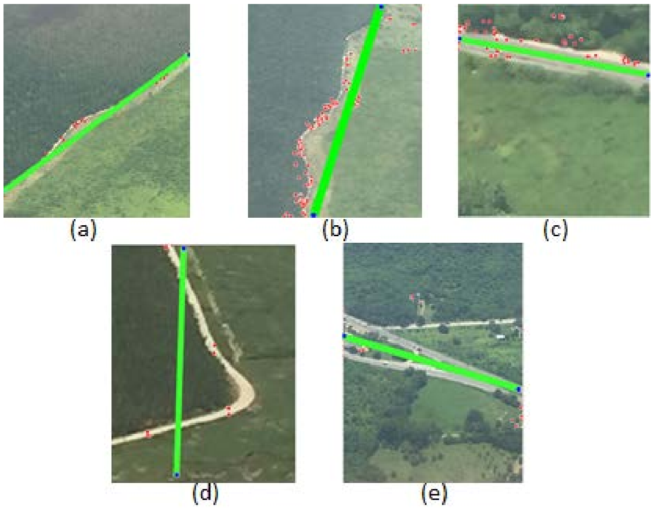

Fig. 5. Tests of the Algorithm with Photos taken from Different Environments and Altitudes.

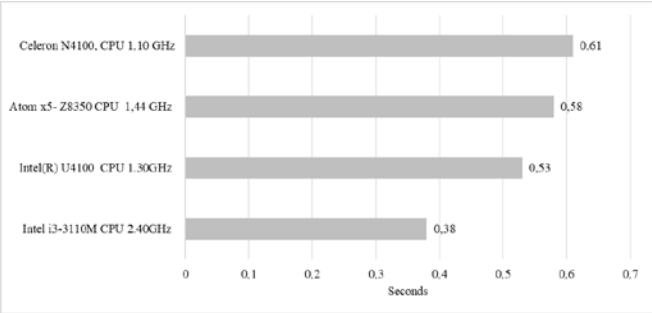

Fig. 6. Algorithm Response Time with Different Processors.

V. Conclusions and Future Work

As can be seen in Fig. 5, the performance of the system is high since it can detect the road in photos taken with different devices at different heights and with different degrees of illumination. It is important to emphasize that the overflight path calculation algorithm is a novel proposal of the researchers, this calculation is performed through the endpoints of the detected road.

Although the decorrelation stretching algorithm has been implemented to highlight the color difference in an image; its use to detect roads in aerial images is a proposal introduced in this research. This proposal demonstrates its robustness since it can separate the colors of the road and the landscape in different aerial images.

Finally, as can be seen in Fig. 5(e) the system can make a coherent decision when encountering intersections, elevated bridges, round-point, and other elements that are usually present on the road. Thus, demonstrating the high degree of stability and applicability of the proposed solution; ensuring that the road detection system can be applied in various environments since it can be functionally adapted to various road environments.

Although the algorithm was tested in a specific task, this implementation offers the possibility that it can be used in various fields of the productive sector. Since the proposed solution allows for the identification clearly and punctually of artificial long objects in a natural landscape. The next step is to complement the system with algorithms that can automatically detect anomalies (pipeline monitoring, fire outbreaks, traffic jams, suspicious vehicles, etc.) that may occur during the overflight of the drone, and thus seek to complete an autonomous aerial tracking and surveillance system.

Acknowledgment

This work was supported by the Universidad Distrital Francisco José de Caldas, in part through CIDC, and partly by the Technological Faculty. The views expressed in this paper are not necessarily endorsed by the university. The authors thank the research group ARMOS for supporting the development of the code and its implementation.

References

[1] A. M. Orozco, " Informe Transporte en cifras Estadisticas 2020", in Ministerio de transporte de colombia, centro de estudios regionales, Versión 1.0, 2021.

[2] M. Al-Smadi, K. Abdulrahim, R. Abdul, "Traffic Surveillance: A Review of Vision Based Vehicle Detection, Recognition and Tracking”, *International Journal of Applied Engineering Research*, vol. 11, no. 1, pp 713-726, 2016.

[3] T.M. Amir-Ul-Haque, M. Das, S. Reza, "Computer Vision Based Traffic Monitoring and Analyzing From On-Road Videos”, *Global Journal of Computer Science and Technology*, vol. 19, no. 2, pp. 1-7, 2019.

[4] P. Vanitchatchavan, "Low-Cost Smart Surveillance and Reconnaissance Using VTOL Fixed Wing UAV", *IEEE Aerospace Conference*, 2020.

[5] S. A. Hassnain Mohsan, M. Asghar Khan, F. Noor, I Ullah, M. H. Alsharif, "Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive ReviewGait", *Drones*, vol. 6, no. 147, pp. 1-27, 2022.

[6] A Berrondo Urruzola, "Detección de carreteras en imágenes de reconocimiento remoto mediante deep", Grado en Ingeniería Informática Computación, Univeridad del pais vasco, Facultad de informatica, 2020.

[7] A. Yasin, A. Kocatepe, "Automatic road detection from orthophoto images", *mersin photogrametri journal*, vol. 2, no. 1, pp. 10-17, 2020.

[8] E. Brewer , J. Lin, P. Kemper, J. Hennin, D. Runfola, “Predicting road quality using high-resolution satellite imagery: A transfer learning approach”, *PLoS ONE*, vol. 16, no.7, pp. 1-18, 2021.

[9] E. Garilli,N. Bruno, F Autelitano, R Roncella, F Giuliani, “Automatic detection of stone pavement's pattern based on UAV photogrammetry”, *Automation in Construction, vol 122*, pp. 1-14, 2021.

[10] R. Nanmaran, S. Hari Priya, “Design and Development of Decorrelation Stretch Technique for Enhancing the Quality of Satellite Images with Improved MSE and UIQI in Comparison with Wiener Filter”, *ECS Transactions*, vol. 107, no. 1, 2022.

[11] P. Sinha, B. Horgan, R. Ewing, E. Rampe, M. Lapotre, M. Nachon, M. Thorpe, A. Rudolph, C. Bedford, K. Maso, E. Champion, P. Gray, E. Reid, M. Faragalli, “Decorrelation stretches(DCS) of visible images as a tool for sedimentary provenance investigation on earth and mars”, *NTRS - NASA Technical Reports Server*, March 16, 2020.

[12] S. Boopathiraja, P. Kalavathi, M. Geethalakshmi, “Performance Analysis of Multispectral Color Composite Image Enhancement Technique using Decorrelation Stretching and Histogram Equalization Methods,” *International Journal of Computer Sciences and Engineering*, vol. 6, no. 4, pp.319-323, 2018.

[13] P. Sudharshan & M. Deepa, “Color detection in RGB-modeled images using MATLAB”, International Journal of Engineering and Technology, vol. 7, no. 2, pp. 29-33, 2018.

[14] A. Hasan Ali, M. Rasheed, S. Shihab, T. Rashid, A. AbdulJabbar, S. Hussein Abed Hamad, “An Effective Color Image Detecting Method for Colorful and Physical Images”, *Journal of Al Qadisiyah for Computer Science and Mathematics*, vol. 13, no. 1, pp 88 – 98, 2021.

[15] A. Raghunandan, Mohana, P. Raghav and H. V. R. Aradhya, "Object Detection Algorithms for Video Surveillance Applications”, *International Conference on Communication and Signal Processing (ICCSP)*, pp. 0563-0568, 2018.

[16] R. Parekh, “Fundamentals of image, audio, and video processing using matlab® with applications to pattern recognition”, Edition1st Edition, First Published202, eBook Published15 April 2021.

[17] Y. Chen, X zhou, “research and implementation of robot path planning based on computer image recognition technology”, *Journal of Physics: Conference Series*, 1744 022097, 2021.

[18] Shutterstock. Image repository. https://www.shutterstock.com/es/image-photo/aerial-view-on-mountain-road-drone-2024801798. 2022.

[19] A. Kumar Shakya, A. Ramola, A. Vidyarthi, K. Sawant, “Satellite Image Enhancement for Small Particle Observation using Decorrelation Stretcher”, *International Conference on Advances in Computing, Communication & Materials (ICACCM)*, 2020.

[20] The MathWorks Inc, “Image Processing Toolbox For Use with MATLAB®”, decorrstretch function, Version 3, User's Guide, https://www.mathworks.com/help/images/ref/decorrstretch.html.

[21] D. Hema, S. Kannan. “Interactive Color Image Segmentation using HSV Color Space”, *Science and Technology Journal*, vol. 7, no. 1, pp. 37-41, 2020.

[22] V. Tiwari, V. Sharma, Development of Algorithm for Object Detection & Tracking Using RGB Model International Journal of Computer Trends and Technology ·, vol. 8, no. 2, pp. 37-44, 2020.

[23] B. Desai, U. Kushwaha, S. Jha," Image Filtering- Techniques, Algorithm and Applications", *GIS Science Journal*, vol. 7, no. 11, pp. 970-975, 2020.

[24] A. R. Kuroswiski, N. M. F. de Oliveira and É. H. Shiguemori, "Autonomous long-range navigation in GNSS-denied environment with low-cost UAV platform", Annual IEEE International Systems Conference (SysCon), 2018.

[25] S. Mufti, V. Roberge, and M. Tarbouchi, "A GPU Accelerated Path Planner for Multiple Unmanned Aerial Vehicles", *IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, 2019.

[26] J. Ananthanarasimhan, P. Leelesh, M .S. Anand, R. Lakshminarayana, "Validation of projected length of the rotating gliding arc plasma using 'regionprops' function", *Plasma Res. Express*, vol. 2, no. 3, 2020.