ALINA: Advanced Line Identification and Notation Algorithm

Mohammed Abdul Hafeez Khan
Florida Institute of Technology
Melbourne, Florida
mkhan@my.fit.edu

Parth Ganeriwala
Florida Institute of Technology
Melbourne, Florida
pganeriwala2022@my.fit.edu

Siddhartha Bhattacharyya
Florida Institute of Technology
Melbourne, Florida
sbhattacharyya@fit.edu

Natasha Neogi
NASA Langley Research Center
Hampton, Virginia
natasha.a.neogi@nasa.gov

Raja Muthalagu
BITS Pilani Dubai Campus
Dubai, UAE
raja.m@dubai.bits-pilani.ac.in

Figure 1. (a) Illustrates the aircraft used for this research, (b) and (c) displays the setup of cameras at different perspectives inside the aircraft, (d) shows the manually defined region of interest on a frame, (e) showcases the annotation label created by ALINA on a frame.

Abstract

Labels are the cornerstone of supervised machine learning algorithms. Most visual recognition methods are fully supervised, using bounding boxes or pixel-wise segmentations for object localization. Traditional labeling methods, such as crowd-sourcing, are prohibitive due to cost, data privacy, amount of time, and potential errors on large datasets. To address these issues, we propose a novel annotation framework, Advanced Line Identification and Notation Algorithm (ALINA), which can be used for labeling taxiway datasets that consist of different camera perspectives and variable weather attributes (sunny and cloudy). Additionally, the CIRCular threshoLd pixEl Discovery And Traversal (CIRCLEDAT) algorithm has been proposed, which is an integral step in determining the pixels corresponding to taxiway line markings. Once the pixels are identified, ALINA generates corresponding pixel coordinate annotations on the frame. Using this approach, 60,249 frames from the taxiway dataset, AssistTaxi have been labeled. To evaluate the performance, a context-based edge map (CBEM) set was generated manually based on edge features and connectivity. The detection rate after testing the annotated labels with the CBEM set was recorded as 98.45%, attesting its dependability and effectiveness.

1. Introduction

With the advancement of sensors, hardware and intelligent software, autonomous vehicles have become the center of attention for computer vision and robotics research. Among the several components of autonomous vehicles, the camera-based lane detection plays a crucial role in perceiving the lines and then assisting autonomous vehicular systems in making decisions to follow the correct path/lanes [11].

It is envisioned, as we gain confidence in the correctness of the design of autonomous systems, it will pave the way for the integration of learning based technologies in assisting safety critical systems such as, autonomous aircraft for

urban air mobility or aerial delivery [6, 28]. As a result, this has increased the need for an advanced perception system, equipped with precise line identification capability. In future, it can assist pilots in the aircraft, especially when navigating the airport taxiway. As stated by the Aviation Safety Network [16], approximately, 33.33% of the aircraft accidents, between the years 2015 and 2022, happened during the taxiing phase. The common causes of taxiway accidents include poor weather conditions and high traffic on the taxiway. In order to address these issues, some researchers have formulated solutions for assisting pilots during the taxiing phase, including: light-based guidance systems, which have been designed by the Federal Aviation Administration and Honeywell [4, 36], computer vision based taxiway guidance techniques [5], Airport Moving Maps [22], and sensors, such as LIDAR and cameras mounted on an aircraft [26, 38]. This research builds on the work of Ganeriwala et. al [18], particularly with the contour-based detection and line extraction method (CDLEM). They introduced AssistTaxi dataset, which incorporates more than 300,000 frames of taxiway and runway instances. The data was collected from Melbourne (MLB) and Grant-Valkaria (X59) general aviation airports.

In this research, we introduce an Advanced Line Identification and Notation Algorithm (ALINA), which is an annotation framework developed to detect and label taxiway line markings from video frames (Fig. 1). ALINA establishes a uniform trapezoidal region of interest (ROI) by utilizing the initial frame, that is consistently applied across all subsequent frames of the video. The ROI is then geometrically modified and the color space is transformed to produce a binary pixel map. ALINA pinpoints pixels representing taxiway markings through the novel CIRCular threshoLd pixEl Discovery And Traversal (CIRCLEDAT) algorithm, leading to frame annotations and coordinate data files. A context-based edge map (CBEM) set was generated for comparison to ensure accuracy in marking detection. ALINA was tested on a subset of AssistTaxi dataset - 60,249 frames extracted from three distinct videos with unique camera angles. The focus of this research had been on 60,249 frames out of a vast 300,000-frames AssistTaxi dataset, as the motivation was to lay down a rigorous yet tractable foundation for the empirical validation of ALINA's efficacy, paving the way for its scalability to larger datasets in future. Through this research, we primarily aim to reduce the intensive, expensive and error-prone manual labeling [17], and provide labeled data to enhance taxiway navigation safety.

Our contribution include the following four main aspects:

- To reduce the intensive, expensive, and error-prone manual labeling process, thus contributing to the efficient combination of automated and manual creation of labelled datasets.
- Development of the Advanced Line Identification and Notation Algorithm (ALINA) providing a robust framework for precise detection and labelling of continuous video datasets, particularly focusing on taxiway line markings.
- Introduction of a novel algorithm, CIRCLEDAT, tailored to pinpoint pixels representing taxiway/road markings with high accuracy, ensuring precise labeling across continuous video frames.
- Establishing a systematic approach based on change in perspective of scenario to justify the sample size of ground truth which is a subset from the AssistTaxi dataset [18].

The remainder of the paper is organized as follows: In Sec. 2, related research works have been discussed. Sec. 3 entails the detailed methodology of ALINA, Sec. 4 presents the experimental results. Finally, Sec. 5 provides the conclusion and outlines the directions for future work.

2. Literature Review

While the research in classification of taxiway line markings is still in its preliminary stages, the domain of car lane detection has seen substantial advancements in road labeling techniques. The knowledge derived from detecting car lanes not only provides fundamental perspectives that guide our comprehension of detecting taxiway markings but also highlights research gaps in car lane labeling methods which can be avoided and guide innovative approaches for taxiway marking detection. Therefore, this section begins by presenting notable works from the car lane detection domain and highlights their research gaps.

Some researchers have used classification techniques for lane detection, where images are segmented into grids for row-based lane location identification [10, 32, 42]. However, these methods lack precision and might miss some lanes. To ensure consistent lane detection, techniques such as parametric curve modeling [15, 25, 31, 37, 39] and keypoint association [33, 40, 41] have been used. Even though these methods acquire high results, they can struggle in adverse weather conditions. Andrei *et al*. [3] have utilized probabilistic Hough transform and dynamic parallelogram region of interest (ROI) to develop a lane detection system. It was implemented on video sequences with manual ROI definition but encountered challenges with capturing curved line endpoints, which are crucial for complete lane boundary delineation in real-world scenarios. Chen *et al*. [8] detected road markings using machine learning with binarized normed gradient detection and principal component analysis (PCA) classification, achieving an accuracy of 96.8%. However, the dependency on PCA limited the adaptability for dynamic scenarios and challenging environments. Similarly, the method proposed by Ding *et al*. [13], which combined PCA and support vector machine (SVM), achieved a

detection accuracy of 94.77%. However, it struggled to detect road markings due to the failure of the ROI extraction, in the case of reflection, occlusion and shadow.

Gupta *et al*. [20] introduced a real-time framework for camera-based road and lane markings, using techniques such as spatio-temporal incremental clustering, curve fitting, and Grassmann manifold learning. The real-time nature of this approach brings challenges in processing efficiency and speed, especially with increasing dataset complexities. Jiao *et al*. [23] proposed an adaptive lane identification system using the scan line method, lane-voted vanishing point, and multi-lane tracking Kalman filtering, achieving a 93.4% F1-Score. This approach lacked the adaptability to diverse conditions across different datasets and real-world scenarios.

Kiske [24] developed an autonomous labeling system for identifying highway lane markings using Velodyne lidar, HDR video streams, and high-precision GPS. However, the reliance on multiple data sources makes it less cost-effective and more complex for wide-scale implementations. Muthalagu *et al*. [27] proposed a vision-based algorithm for lane detection in self-driving cars, which used polynomial regression, histogram analysis, and a sliding window mechanism for detecting both straight and curved lines. The algorithm achieved notable accuracy, but its high computational demand poses scalability issues, and its limitation in detecting steep foreground curves presents challenges in dynamic terrains.

Contour-based detection and line extraction method (CDLEM) was initially used for labeling the taxiway line markings from AssistTaxi dataset [18]. The Canny edge detection identified the line marking's edges. Subsequently, the Hough transform, Ramer-Douglas-Peucker, and Bresenham's algorithms were utilized to identify and label both straight and curved taxiway line markings. The limitation this approach posed was the requirement to outline an ROI around taxiway line marking for every scenario shift.

While labeling car lanes has provided a guiding example for an end to end lane detection system [2, 29, 43], a direct comparison of lane detection to labeling taxiway line markings is not being made in this research. The distinct semantics and attributes associated with the car lane datasets and airport taxiways dataset pose different set of challenges. For example, the airport taxiway dataset has different layouts, diverse markings, and presence of aircraft on airport taxiways. Transfer learning can be one of the approaches to address the differences and identify the commonalities within the two datasets [19], however it requires the taxiways to be extensively labeled. Therefore, our research emphasizes creating and evaluating algorithms for labeling specific to airport taxiways, rather than contrasting them with car lane datasets.

3. Methodology

In this section, we provide a detailed overview of the ALINA framework using Fig. 2 (each subsection elaborates elements from the figure), and we discuss its steps referencing image instances from Fig. 3 and Fig. 4.

3.1. Frame Representation

While reading a frame, ALINA stores each x, y coordinate and its red, green, blue (RGB) channel values from the frame into a multi-dimensional array, as represented in Eq. (1).

$$A[i, j, k] = I[i, j, k] \tag{1}$$

where, i, j, and k denote the row, column, and channel indices of the frame, respectively. This formula copies the pixel values of the frame at location (i, j) in the k^{th} color channel into the corresponding location in the array.

3.2. Interactive ROI Definition on the Initial Frame

The user provides source points to ALINA for creating a trapezoidal region of interest (ROI) on the initial frame of the video, as shown in Fig. 3 (a), (e), (f). It is drawn in the vicinity of where the camera expects to see the taxiway's line markings. ALINA treats the ROI as a constraint to limit the search area for taxiway's line markings, resulting in reduced computational complexity and improved detection accuracy [21]. The source points, which correspond to the vertices of ROI, are denoted as follows: $[x_1, y_1]$, $[x_2, y_2]$, $[x_3, y_3]$, $[x_4, y_4]$.

3.3. Perspective Transformation of the ROI

The next step in ALINA is warping the perspective. The trapezoidal ROI is warped into a bird's eye view. Fig. 3 (b), (f), (j) illustrates this transformation. The destination points, which correspond to the vertices of a rectangle defining the bird's eye view, are represented as: $[x'_1, y'_1]$, $[x'_2, y'_2]$, $[x'_3, y'_3]$, $[x'_4, y'_4]$.

It enables a more comprehensive and consistent view of the taxiway and its features, allowing for improved detection of line markings irrespective of their orientation or curvature [1].

Given the pairs of source and destination points, a matrix M is derived by solving a system of linear equations formed from the pixel correspondences. The matrix essentially maps any pixel in the trapezoidal ROI to its corresponding pixel position in the bird's eye view.

The matrix M is represented in Eq. (2):

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix} \tag{2}$$

with each m_{ij} computed based on the defined system of equations.

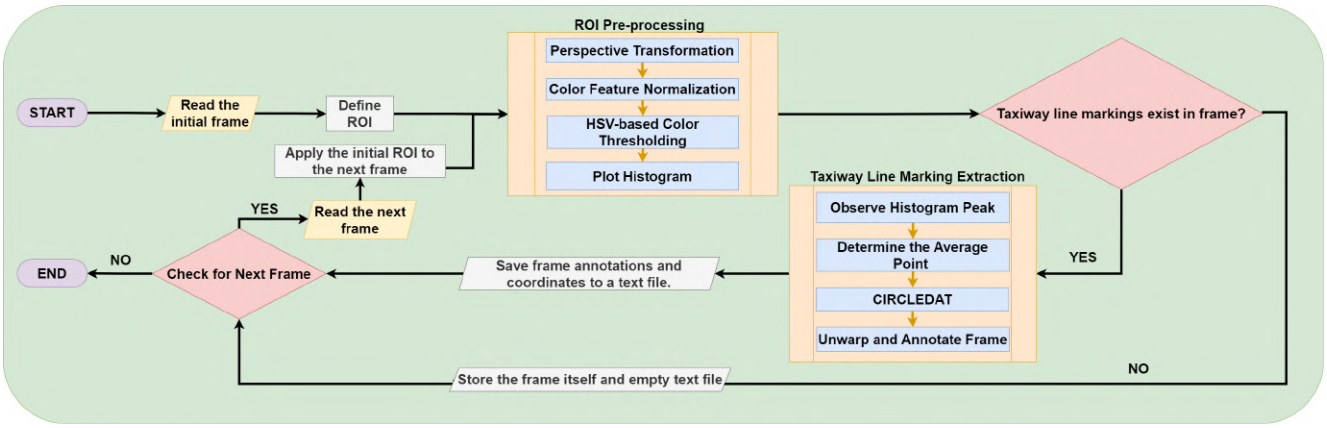


Figure 2. Framework of ALINA

Once M is computed, it's applied to each pixel in the trapezoidal ROI to achieve its position in the bird's eye view. This transformation is attained using Eq. (3):

$$\mathrm{t}(x,y) = s\left(\frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}}\right) \tag{3}$$

Where $t(x,y)$ and $s(x,y)$ denote pixel coordinates in the bird's eye view and trapezoidal ROI respectively.

3.4. Color Feature Normalization

To accurately distinguish line markings within the ROI, ALINA performs normalization of color characteristics. The normalization phase consists of the following key steps:

1. RGB **to** HSV **Conversion:** The RGB color space of ROI is transformed to the hue, saturation, and value (HSV) color space. Given a pixel's RGB values, denoted by (R,G,B), the transformation into the HSV space is depicted as $(R,G,B) \rightarrow (H,S,V)$. This transformation is performed because HSV factors in variations induced by lighting conditions and intrinsic color properties, providing a robust representation of color in an image [35].
2. **Component Decomposition:** The HSV is decomposed into its individual channels: H,S,V.
3. **Normalization:** Every component X, where $X \in \{H,S,V\}$, undergoes 8-bit min-max normalization as shown in Eq. (4):

$$X_{\text{norm}} = \frac{X - \min(X)}{\max(X) - \min(X)} \times 255 \tag{4}$$

where $\min(X)$ and $\max(X)$ denote the minimum and maximum values of the component X, respectively. This ensures the values are scaled to fit within the $[0,255]$ range.

4. **Component Recombination:** The normalized components, denoted as $H_{\text{norm}}, S_{\text{norm}}, V_{\text{norm}}$, are recombined to form the normalized HSV ROI, represented as HSV_{norm}.

Following this normalization process, the color features of the frame are standardized, as illustrated in Fig. 3 (c), (g), (k). This ensures uniform contribution from each feature to the final representation, facilitating precise color thresholding for taxiway line marking detection in the subsequent step.

3.5. HSV-based Color Thresholding

Color thresholding is an essential technique in image segmentation, leveraging color information to partition image pixels into meaningful regions [9]. In the context of ALINA, this technique finds its application in isolating the taxiway line markings from the rest of the regions within the ROI, using the HSV color space. Hue captures the wavelength of color, while saturation measures the intensity, and value quantifies the brightness.

Determining the precise range for HSV that corresponds to the taxiway line markings necessitated a series of empirical tests. Multiple frame samples were analyzed and a frequency distribution of HSV values for taxiway line marking regions were plotted. Peaks in this distribution, indicative of dominant HSV values for taxiway line markings, helped in ascertaining the lower and upper bounds of H,S,V, i.e. (0, 70, 170) and (255, 255, 255), respectively.

For a pixel p with HSV values denoted by $H(p)$, $S(p)$, $V(p)$, the color thresholding function $\Theta(p)$is defined in Eq. (5):

$$\Theta(p) = \begin{cases} 255 & \text{if } 0 \le H(p) \le 255 \\ & \text{and } 70 \le S(p) \le 255 \\ & \text{and } 170 \le V(p) \le 255 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Here, $\Theta(p)$ produces a binary outcome for each pixel in the ROI: a value of 255 (white) indicates a pixel belonging

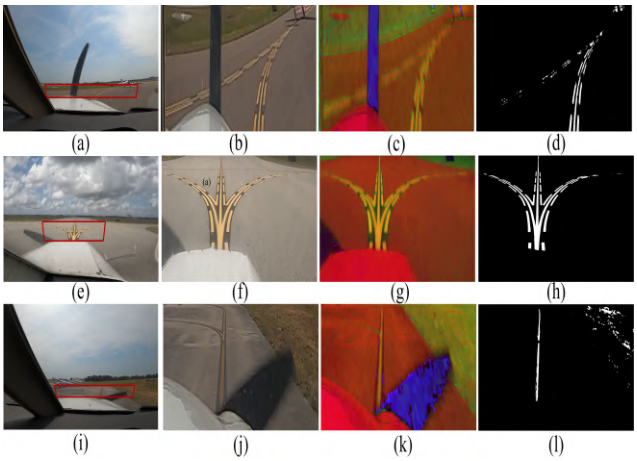


Figure 3. (a),(e),(i): ROI Definition, (b),(f),(j): Perspective Transformation, (c),(g),(k): Color Feature Normalization, (d),(h),(l): HSV-based Color Thresholding

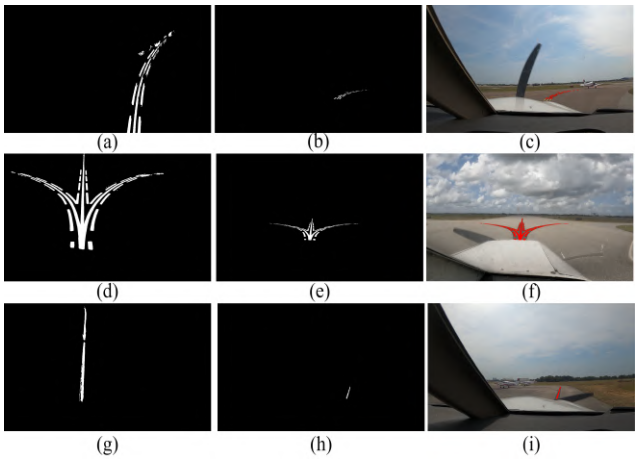


Figure 4. (a),(d),(g): CIRCLEDAT, (b),(e),(h): Frame Unwarping, (c),(f),(i): Taxiway Line Marking Annotation

to the taxiway line marking, while 0 (black) denotes a pixel that does not. However, after applying HSV-based color thresholding, some non-taxiway line marking pixels within the ROI still matched the specified range, as shown in Fig. 3 (d), (h), (l). ALINA addresses this in the subsequent step.

3.6. Histogram-based Analysis of the Thresholded ROI

The histogram analysis focuses on the vertical projection of white pixels in the thresholded ROI, analyzing the spatial distribution and density of the taxiway line markings.

For a thresholded ROI with dimensions $W \times H$, where W represents the width (number of columns) and H represents the height (number of rows), the binary representation of a pixel at position i, j is defined in the Eq. (6), where i is the column index and j is the row index:

$$B(i,j) = \begin{cases} 1 & \text{if pixel at } (i,j) \text{ is white} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The vertical histogram, denoted as $H_{\text{vert}}(i)$, quantifies the column-wise distribution of white pixels and is calculated using Eq. (7):

$$H_{\text{vert}}(i) = \sum_{j=0}^{H} B(i,j) \tag{7}$$

For each column i, this equation aggregates the presence of white pixels across all rows j, offering a count of white pixels for that column. When plotted against the column index i, the histogram produced accentuates the density of white pixels along the y-axis. Peaks in this histogram, denote the presence and spatial positioning of taxiway line markings within the ROI.

3.7. Identifying Line Markings and Mitigating False Detections

The histogram peak value, represented by H_p, indicates the highest density of white pixels in a particular column of the histogram. It was important to identify whether a peak in the histogram truly represents a taxiway line marking or in contrast, is a result of noise or other disturbances. We identified the presence of a taxiway line marking in the ROI using a threshold value.

3.8. CIRCLEDAT: Circular Threshold Pixel Discovery and Traversal Algorithm

The CIRCLEDAT algorithm assists in isolating the pixels that correspond to the taxiway line markings, irrespective of their dimension or curvature, and eliminate all other pixels from the ROI. For an ROI $\mathcal{I}$ with dimensions $W \times H$, an initial coordinate pair (x, y), a radius θ, a set $\mathcal{V}$ for recording visited pixels, and an array $\mathcal{L}$ for collecting taxiway line marking pixels, the algorithm executes as follows:

1. **Initialization:** The starting point (x, y) is pushed onto a stack S and recorded in $\mathcal{V}$. A set $\mathcal{R}$ is created containing all pixel coordinates within the circular distance.
2. **Exploration:** While S is not empty, the top coordinate (x, y) is popped. If $\mathcal{I}[y][x] = 255$, which indicates a white pixel, (x, y) is added to $\mathcal{L}$. For each offset (i, j) in $\mathcal{R}$:
   - Calculate new coordinates $(x_{\text{new}}, y_{\text{new}}) = (x+i, y+j)$.
   - If $(x_{\text{new}}, y_{\text{new}}) \in \mathcal{V}$ or $(x_{\text{new}}, y_{\text{new}})$ is outside $0 \leq x_{\text{new}} < W$ and $0 \leq y_{\text{new}} < H$, continue to the next offset.
   - Otherwise, push $(x_{\text{new}}, y_{\text{new}})$ onto S and record in $\mathcal{V}$.
3. **Result:** Once S is exhausted, the algorithm returns $\mathcal{L}$, representing all the white pixels, which correspond to the taxiway line markings.

The Fig. 4 (a), (d), (g) represent the output generated after applying the CIRCLEDAT algorithm (1) within the ROI. Through a depth-first search approach, CIRCLEDAT ensures a comprehensive exploration of pixels pertaining to the taxiway line markings. The $\mathcal{V}$ set aids in efficient traversal by preventing revisits, and the use of $\mathcal{R}$ ensures a circular-based neighborhood exploration.

Algorithm 1 Circular Threshold Pixel Discovery and Traversal

function CIRCLEDAT($image$)
  $stack \leftarrow [(x, y)]$
  $visited.add((x, y))$
  $circular\_range \leftarrow list(product(range(-\theta, \theta + 1), repeat \leftarrow 2))$
  $circular\_range.remove((0, 0))$
  while $stack$ is not empty **do**
    $x, y \leftarrow stack.pop()$
    if $image[y][x] = 255$ **then**
      $add\ (x, y)\ to\ line\_marking\_pixels$
      for $i, j \leftarrow circular\_range$ **do**
        $x\_new \leftarrow x + i$
        $y\_new \leftarrow y + j$
        if $(x\_new, y\_new) = visited$ **then**
          continue
        end if
        if $x\_new, y\_new < 0\ or\ x\_new, y\_new \geq image.width, image.height$ **then**
          continue
        end if
        $add\ (x\_new, y\_new)\ to\ stack$
        $visited.add((x\_new, y\_new))$
      end for
    end if
  end while
  $return\ line\_marking\_pixels$
end function

3.9. Frame Unwarping and Annotating the Taxiway Line Marking Pixels

The final stage of labeling the frame involves an inverse perspective transformation that returns the warped ROI to its original view, as shown in Fig. 4 (b) ,(e), (h). After obtaining the unwarped ROI, white pixels are located and mapped onto the original frame using color red to clearly identify the taxiway line markings, as illustrated in Fig. 4 (c), (f), (i). This representation provides a precise and clear depiction of the taxiway line markings in the original frame. In addition, a text file is generated containing the x, y coordinates of all pixels corresponding to the taxiway line markings.

Having completed the labeling of the video's initial frame, ALINA proceeds to label the subsequent frames using the same process and the ROI established for the first frame, as illustrated in Fig. 2.

4. Experimental Results

4.1. ALINA Performance: Specifications and Scenarios

The detailed processing time breakdown for ALINA when labeling a frame is presented in Tab. 1. On average, ALINA requires 50.9 milliseconds (ms) to label a single frame, yielding a rate of approximately 19.65 frames per second (fps). In the course of our systematic labeling process, ALINA effectively labeled 60,249 frames on a Linux system equipped with an Intel Core i7-9700K CPU clocked at 3.60GHz and 15GB RAM, operating on Ubuntu 22.04.1 LTS. The algorithm was developed in the PyCharm IDE with Python 3.8.15, harnessing essential libraries such as OpenCV, NumPy, MatPlotLib, and statistics.

Table 1. Processing time analysis of ALINA (in Milliseconds).

Process	**Time (ms)**
Perspective Transformation	4.41
Color Feature Normalization	5.91
HSV-based Color Thresholding	1.05
Histogram Analysis	28.71
CIRCLEDAT	3.33
Projection Remapping	6.68
Total	**50.09**

As mentioned earlier, we applied ALINA on a subset of the AssistTaxi dataset, consisting of 60,249 frames extracted from three distinct videos, each with unique camera angles. Fig. 3 and Fig. 4 demonstrates ALINA's consistent labeling performance across these frames, unaffected by differing camera angles or weather conditions. Specifically, in Fig. 3 (a) from the first video, ALINA labeled the taxiway line marking between two aircrafts. In Fig. 3 (e) from the second video, it labeled three directional taxiway line markings: left, straight, and right. Lastly, in Fig. 3 (i) from the third video, ALINA labeled the taxiway line marking in front of the aircraft, which is situated between stationary aircraft on the left and grassy area on the right.

4.2. Ablation Study on Threshold Value

Initially, the decision was kept binary, in which there was no set threshold, as shown in Eq. (8). This approach considered even a slight peak in the histogram as a taxiway line marking, leading to a high percentage of false positives, as shown in Tab. 2.

$$\Delta(p) = \begin{cases} 1 & \text{if } H_p > 0 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Here $\Delta(p)$ indicates the presence (1) or absence (0) of the taxiway line marking.

Through empirical testing and histogram analyses, we identified that true taxiway line markings consistently yielded peak values significantly above sporadic noise. As shown in Tab. 2, setting a threshold at 75 led to a substantial fall in the percentage of false positives, but by raising the threshold to 150, the false positives were dropped to 0. Hence, the $T_{optimal}$ was established at 150, using the equations 9 and 10. This was primarily because true taxiway line markings consistently resulted in columns extending well over 150 white pixels, while disturbances never reached this level.

Table 2. Threshold Value Comparison

Threshold Value	False Positives (%)
0	83.33
75	22.22
150	0

$$T_{\text{optimal}} = \underset{T}{\operatorname{argmin}} \left(F_{\text{false positive}}(T) - F_{\text{true positive}}(T) \right) \quad (9)$$

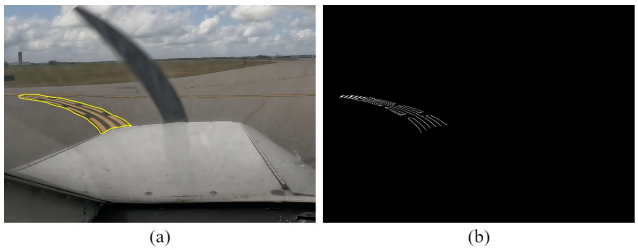
$$\Delta(p) = \begin{cases} 1 & \text{if } H_p >= T_{optimal} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Once the taxiway line marking is detected using $\Delta_{150}(p)$, ALINA extracts its centroid coordinate, initializing the subsequent CIRCLEDAT algorithm. In contrast, if there is no taxiway line marking in the ROI, ALINA simply stores the frame along with an empty text file, signifying that there is no taxiway line marking present in the frame, as shown in Fig. 2.

4.3. Generating a Context-based Edge Map Set

To assess ALINA and CDLEM's effectiveness, we manually created a set of context-based edge maps (CBEM), which emphasize the edge pixel presence, edge corner localization, thick edge occurrence, and edge connectivity [7]. We prioritized the detection of the edges of a taxiway line marking as our primary validation metric, because this alone provides precise information into the line marking's position within the frame. Our approach for developing the CBEM was as follows:

1. **Outlining the Contour Region:** The taxiway line markings in the frames were outlined manually to create an accurate reference without including any other edge details from the frame.



(a) (b)

Figure 5. (a) Manually Outlined Contour Region (b) CBEM

2. **Pre-processing:** The contour region was transformed to grayscale and then subjected to Gaussian blur, which helped in reducing the visual noise and improved the clarity of edges. Consequently, the gradient amplitude and direction were calculated and non-maximum suppression was applied to eliminate the non-edge pixels.
3. **Edge Detection with the Canny Algorithm:** We used the Canny algorithm for precise edge extraction of taxiway line markings. We also employed an automated method for selecting the upper and lower thresholds [34]. This method computes the median pixel intensity v of an image and subsequently determines the thresholds using equations 11 and 12:

$$\text{lower} = \max(0, (1.0 - \sigma) \times v) \quad (11)$$

$$\text{upper} = \min(255, (1.0 + \sigma) \times v) \quad (12)$$

where σ is a coefficient, defaulting to 0.33, for refining the thresholds.

The process of manually outlining the contour region around the taxiway line marking and generating CBEM for a frame is illustrated in Fig. 5 (a) and (b) respectively. The Fig. 1 (e) shows the output of ALINA on the same frame.

From the 60,249 frames, we selected a set of 120 frames to construct the CBEM's. The 60,249 frames spanned from three videos with durations of 11.47, 1.34, and 3.23 minutes. Our objective was to identify the frames representing scenario shifts. Instead of conducting a granular frame-by-frame analysis, we reviewed the videos comprehensively and marked the specific frames that captured the scenario shifts, amounting to a total of 120 images.

Our selection is underpinned by the Law of Large Numbers (LLN), as illustrated in Eq. (13):

$$\bar{X}_n \to \mu \quad \text{as} \quad n \to \infty \quad (13)$$

where $\bar{X}_n$ represents the sample mean and μ is the expected population mean [14]. This indicates that our subset, if representatively selected, provides a reliable approximation of the comprehensive dataset's attributes.

Furthermore, the Central Limit Theorem (CLT) [14] also reinfornces our approach, as expressed in Eq. (14):

$$\frac{S_n - n\mu}{\sigma\sqrt{n}} \to N(0, 1) \quad (14)$$

where $S_n = X_1 + X_2 + \ldots + X_n$ and X_i are independent, identically distributed random variables, n is the sample size, μ is the population mean, and σ is the standard deviation. The CLT's cornerstone assertion, relevant in our context, is that with a sufficiently extensive sample size, the sample mean's distribution gravitates towards a normal distribution. This holds irrespective of the originating population's distribution. Therefore, the mean distribution extrapolated from all possible 120-frame subsets is poised to achieve normality. Leveraging the Central Limit Theorem, our diverse 120-frame sample is statistically representative of the entire 60,249-frame dataset. With the backing of both LLN and CLT, our method ensures a robust evaluation of ALINA and CDLEM using CBEM set.

4.4. Sliding Window Vs CIRCLEDAT

In the study by Muthalagu et al. [27], a sliding window (SW) search algorithm detected lane line marking pixels with a time complexity of $O(m \times n)$, where m and n represent the height and width of the frame, respectively. In contrast, our work introduces the innovative CIRCLEDAT algorithm, which pinpoints line marking pixels with a significantly reduced time complexity of $O(k)$, where k is number of pixels corresponding to the line marking in a given frame. Prior to introducing CIRCLEDAT, we used the SW search algorithm to detect taxiway line marking pixels. Tab. 3 compares the performance of both algorithms when tested on the taxiway dataset frames.

Table 3. Comparative Analysis of Pixel Detection Algorithms

Algorithm	**Time Complexity**	**Processing Time (ms)**
SW Search	$O(m \times n)$	10.90
CIRCLEDAT	$O(k)$	3.33

4.5. Performance Evaluations

We evaluated ALINA's performance against CDLEM using CBEM set. By comparing x and y coordinates from both the CBEM and ALINA or CDLEM, we calculated true positives (TP) and false negatives (FN). TP indicates accurate identification of taxiway line marking pixels, while FN denotes missed pixels that should have been identified as part of the taxiway line marking.

The recall or detection rate, essential for evaluating object detection algorithms, gauges an algorithm's accuracy in identifying the particular objects in a frame [30] [12]. In airport taxiway line marking detection, missing a marking can pose safety risks, underscoring the importance of comprehensive identification. The detection rate is calculated as the ratio of TP values to the sum of TP and FN values, as shown in Eq. (15).

$$Detection\ Rate\ (Recall) = \frac{TP}{TP + FN} \tag{15}$$

ALINA and CDLEM achieved detection rates of 98.45% and 91.14%, respectively, as shown in Tab. 4. Additionally, in terms of processing time, ALINA processed a frame in 50.09 ms, corresponding to approximately 19.65 fps, whereas CDLEM took 120.35 ms per frame, translating to roughly 8.30 fps.

ALINA's superior performance is attributed to several key features. Its perspective transformation provides a bird's-eye view of taxiway, eliminating distortions and offering clarity in distinguishing line markings from anomalies—a challenge for CDLEM due to varying distances and angles. Unlike CDLEM, which may miss essential pixels using the Hough Transform and curve fitting, ALINA's shift to the HSV color space, prioritizing H, S, and V components, enhances taxiway line marking detection even under varying weather. The CIRCLEDAT algorithm within ALINA swiftly captures all crucial pixels of taxiway markings, regardless of their shape or fragmentation. A notable limitation of CDLEM is its need to define a new ROI for each scenario shift, resulting in 120 scenarios for 60,249 frames, demanding extensive video pre-viewing. In contrast, ALINA only requires an ROI for the initial frame of a video, applied consistently to all following frames, leading to just 3 ROIs for the same number of frames.

Table 4. Comparing ALINA Vs CDLEM

Algorithm	**Detection Rate (%)**	**Processing Time (ms)**
ALINA	98.45	50.09
CDLEM	91.14	120.35

5. Conclusion

In this work, we propose ALINA, a novel annotation framework, primarily designed for labeling pixel coordinates in taxiway datasets. This approach streamlines the annotation process, significantly reducing cost and manual labor needed for precise labeling in these contexts. We also propose a traversal algorithm CIRCLEDAT, which determines the pixels corresponding to the taxiway line markings. We provide a comparative analysis with the sliding window search algorithm and evaluate the performance of the framework on a subset of the AssistTaxi dataset. We have tested ALINA with labels generated for 60,249 frames, and evaluated it with a context-based edge map (CBEM) set which was generated manually. We also provide theoretical analysis and a comparative study for ALINA to Contour-Based Detection and Line Extraction Method (CDLEM). In the future, we aim to evaluate ALINA for annotating car lane datasets, with the CIRCLEDAT algorithm being utilized to identify pixel coordinates of road lane markings.

References

[1] Md Abdullah Al Noman, Li Zhai, Firas Husham Almukhtar, Md Faishal Rahaman, Batyrkhan Omarov, Samrat Ray, Shahajan Miah, and Chengping Wang. A computer vision-based lane detection technique using gradient threshold and hue-lightness-saturation value for an autonomous vehicle. *International Journal of Electrical and Computer Engineering*, 13(1):347, 2023. 3

[2] Heba Aly, Anas Basalamah, and Moustafa Youssef. Lanequest: An accurate and energy-efficient lane detection system. In *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 163–171, 2015. 3

[3] Mihail-Alexandru Andrei, Costin-Anton Boiangiu, Nicolae Tarbă, and Mihai-Lucian Voncilă. Robust lane detection and tracking algorithm for steering assist systems. *Machines*, 10 (1):10, 2022. 2

[4] Anonym. Evaluation of a prototype advanced taxiway guidance system (atgs), 2017. 2

[5] Aman Batra and Jason Gauci. Aerodrome taxiway line detection and cross-track error estimation using computer vision techniques. In *2020 IEEE Aerospace Conference*, pages 1–9, 2020. 2

[6] Siddhartha Bhattacharyya, Jennifer Davis, Anubhav Gupta, Nandith Narayan, and Michael Matessa. Assuring increasingly autonomous systems in human-machine teams: An urban air mobility case study. *Electronic Proceedings in Theoretical Computer Science*, 348:150–166, 2021. 2

[7] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 7

[8] Tairui Chen, Zhilu Chen, Quan Shi, and Xinming Huang. Road marking detection and classification using machine learning algorithms. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 617–621, 2015. 2

[9] Heng-Da Cheng, X_ H_ Jiang, Ying Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern recognition*, 34(12):2259–2281, 2001. 4

[10] Shriyash Chougule, Nora Koznek, Asad Ismail, Ganesh Adam, Vikram Narayan, and Matthias Schulze. Reliable multilane detection and classification by utilizing cnn as a regression network. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 2

[11] Luca Cultrera, Lorenzo Seidenari, Federico Becattini, Pietro Pala, and Alberto Del Bimbo. Explaining autonomous driving by learning end-to-end visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020. 1

[12] Francis X Diebold and Robert S Mariano. Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1):134–144, 2002. 8

[13] Ling Ding, Huyin Zhang, Jinsheng Xiao, Bijun Li, Shejie Lu, Reinhard Klette, Mohammad Norouzifard, and Fang Xu. A comprehensive approach for road marking detection and recognition. *Multimedia Tools and Applications*, 79(23-24): 17193–17210, 2020. 2

[14] William Feller. *An introduction to probability theory and its applications, Volume 2*. John Wiley & Sons, 1991. 7

[15] Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17062–17070, 2022. 2

[16] Flight Safety Foundation. Asn aviation safety database: https://aviation-safety.net/database/, 2023. 2

[17] Teodor Fredriksson, David Issa Mattos, Jan Bosch, and Helena Holmström Olsson. Data labeling: An empirical investigation into industrial challenges and mitigation strategies. In *International Conference on Product-Focused Software Process Improvement*, pages 202–216. Springer, 2020. 2

[18] Parth Ganeriwala, Siddhartha Bhattacharyya, Sean Gunther, Brian Kish, Mohammed Abdul Hafeez Khan, Ankur Dhadoti, and Natasha Neogi. Assisttaxi: A comprehensive dataset for taxiway analysis and autonomous operations. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 1094–1099, 2023. 2, 3

[19] Parth Ganeriwala, Siddhartha Bhattacharyya, and Raja Muthalagu. Cross dataset analysis and network architecture repair for autonomous car lane detection*. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6, 2023. 3

[20] Any Gupta and Ayesha Choudhary. A framework for camera-based real-time lane and road surface marking detection and recognition. *IEEE Transactions on Intelligent Vehicles*, 3(4):476–485, 2018. 3

[21] Toan Minh Hoang, Se Hyun Nam, and Kang Ryoung Park. Enhanced detection and recognition of road markings based on adaptive region of interest and deep learning. *IEEE Access*, 7:109817–109832, 2019. 3

[22] JEPPESEN. Jeppesen airport moving maps: ww2.jeppesen.com/navigation-solutions/airport-moving-maps/, 2021. 2

[23] Xinyu Jiao, Diange Yang, Kun Jiang, Chunlei Yu, Tuopu Wen, and Ruidong Yan. Real-time lane detection and tracking for autonomous vehicle applications. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 233(9):2301–2311, 2019. 3

[24] Jeffrey Kiske. Automatic labeling of lane markings for autonomous vehicles. 3

[25] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3694–3702, 2021. 2

[26] Claire Meymandi-Nejad, Esteban Perrotin, Ariane Herbulot, and Michel Devy. Aircraft navigation on taxiways: Evaluation of line detection algorithms proposed for automotive applications. In *Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control*, New York, NY, USA, 2021. Association for Computing Machinery. 2

[27] Raja Muthalagu, Anudeepsekhar Bolimera, and V Kalaichelvi. Lane detection technique based on perspective transformation and histogram analysis for self-driving cars. *Computers & Electrical Engineering*, 85:106653, 2020. 3, 8

[28] Nandith Narayan, Parth Ganeriwala, Randolph M. Jones, Michael Matessa, Siddhartha Bhattacharyya, Jennifer Davis, Hemant Purohit, and Simone Fulvio Rollini. Assuring learning-enabled increasingly autonomous systems*. In *2023 IEEE International Systems Conference (SysCon)*, pages 1–7, 2023. 2

[29] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 286–291, 2018. 3

[30] Rafael Padilla, Wesley L Passos, Thadeu LB Dias, Sergio L Netto, and Eduardo AB Da Silva. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics*, 10(3):279, 2021. 8

[31] Marc Proesmans Proesmans, Bert De Brabandere Brabandere, Davy Neven Neven, Luc Van Gool Gool, and Stamatios Georgoulis Georgoulis. Towards end-to-end lane detection: an instance segmentation approach. 2018. 2

[32] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 276–291. Springer, 2020. 2

[33] Zhan Qu, Huan Jin, Yang Zhou, Zhen Yang, and Wei Zhang. Focus on local: Detecting lane marker from bottom up via key point. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14122–14130, 2021. 2

[34] Adrian Rosebrock. Zero-parameter, automatic canny edge detection with python and opencv. 2015. 7

[35] Li Shuhua and Guo Gaizhi. The application of improved hsv color space model in image processing. In *2010 2nd International Conference on Future Computer and Communication*, pages V2–10. IEEE, 2010. 4

[36] P Stern. Advanced surface movement guidance system: Installing honeywell's a- smgcs system has helped reduce ground delays at incheon international airport (iia), 2001. 2

[37] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6150–6156. IEEE, 2021. 2

[38] Kevin Theuma, David Zammit-Mangion, Jason Gauci, Kenneth Chircop, and Nicolas Riviere. A particle filter for ground obstacle tracking in the airfield. In *AIAA Scitech 2019 Forum*, page 1265, 2019. 2

[39] Bingke Wang, Zilei Wang, and Yixin Zhang. Polynomial regression network for variable-number lane detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 719–734. Springer, 2020. 2

[40] Jinsheng Wang, Yinchao Ma, Shaofei Huang, Tianrui Hui, Fei Wang, Chen Qian, and Tianzhu Zhang. A keypoint-based global association network for lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1392–1401, 2022. 2

[41] Shenghua Xu, Xinyue Cai, Bin Zhao, Li Zhang, Hang Xu, Yanwei Fu, and Xiangyang Xue. Rclane: Relay chain prediction for lane detection. In *European Conference on Computer Vision*, pages 461–477. Springer, 2022. 2

[42] Seungwoo Yoo, Hee Seok Lee, Heesoo Myeong, Sungrack Yun, Hyoungwoo Park, Janghoon Cho, and Duck Hoon Kim. End-to-end lane marker detection via row-wise classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 1006–1007, 2020. 2

[43] Wenhui Zhang and Tejas Mahale. End to end video segmentation for driving : Lane detection for autonomous car. *CoRR*, abs/1812.05914, 2018. 3