ISSN 2007-9737

Improving the Automatic Counting in a Region of Interest in Videos of Urban Traffic Taken by Drones Using Video Stabilization

Guillermo A. Jiménez-Frías[1], Francisco J. Hernandez-Lopez[2], Thomas Batard[1,*], Victor Muñiz-Sánchez[1], Jean-Bernard Hayet[1], Silvana Forti-Sosa[3], Eduardo Pérez-Pech[3]

[1] Centro de Investigación en Matemáticas A.C., Mexico

[2] SECIHTI - Centro de Investigación en Matemáticas A.C., Mexico

[3] Universidad Modelo, Mexico

{guillermo.jimenez, fcoj23, thomas.batard, victor_m, jbhayet}@cimat.mx, {silvana.forti, eduardo.perez.pech}@gmail.com

Abstract. Due to the rapid progress of video analytics, traffic monitoring has emerged as a vital method for collecting information about traffic conditions. This paper presents an improved, automated system for counting vehicles and pedestrians from drone footage. The system allows a user to define any counting line within the video scene. Our four-module pipeline consists of: (1) a robust video stabilization module that compensates possible movements of the drone, (2) a deep learning-based object detection model trained on a custom dataset recorded in Yucatán, Mexico to identify eight distinct classes (car, truck, bus, van, motorcycle, bike, moto-taxi, and pedestrian), (3) an object tracking algorithm that analyzes detection bounding boxes, and (4) a counting module that tallies objects crossing the user-defined segment. The proposed counting system extends our previous work that did not use a stabilization module [6]. On experiments with the YucaMex-MOCS dataset, the proposed counting system reaches average precisions of 72.52% and 75.38% without and with the stabilization module, respectively, which shows the relevance of adding a stabilization module.

Keywords. Automatic counting, video stabilization, object detection, object tracking, deep learning

1 Introduction

Recent advances in video analysis and deep learning have positioned drones as powerful tools for traffic monitoring and surveillance. Compared to traditional surveillance cameras, drones provide clearer views and cover a wider range of areas with reduced occurrences of object occlusions and shadow interferences, while also providing flexibility in complex environments without requiring costly infrastructure. Combined with state-of-the-art deep learning object detection and tracking models, drone-based systems enable accurate, robust, real-time traffic analysis [2, 7].

1.1 Related work

Several approaches have been proposed for vehicle detection and counting in traffic monitoring systems. In [11], a system based on optical flow and block matching estimates movement patterns; however, it requires user intervention to manually define the entry and exit zones for vehicles at intersections. Accuracy ranges from 14.29% to 100%. Similarly, the method in [14] relies on background subtraction and shadow removal, combined with a region of interest (ROI) defined by the user. This approach tracks moving objects within the ROI; however, it does not differentiate between different types of vehicles. In [16], the ViBe background subtraction method is combined with Kernelized Correlation Filters (KCF)

for object detection and tracking, respectively. This system requires defining a virtual line across the scene, where objects are counted once they cross it. It achieves 90% accuracy in vehicle counting on static-background videos and 85% on moving-background videos.

More advanced approaches have leveraged deep learning techniques: for instance, [17] proposes an automatic vehicle counting algorithm using R-CNN for detecting and classifying cars, vans, trucks, and buses. They apply Intersection over Union (IoU) with the Hungarian algorithm for robust tracking across frames. The model reaches 34.66%, 27.72%, 19.27%, and 28.10% in average precisions (APs) for cars, vans, trucks, and buses, respectively, while the counting method yields a mean absolute error between 2.47 and 13.86. Expanding the scope, [15] introduces an automatic counting system tailored for motorcycle-dominated intersections in Vietnam, with 15 vehicle categories. While effective in many scenarios, the approach struggles to detect small vehicles at a distance and partially occluded motorcycles, due to the camera placement. The counting model achieves over 90% accuracy compared to manual counting. Similarly, [9] proposes a drone-based object detection framework using YOLOv4 with a convolutional block attention module, focused on five vehicle categories (motorcycles, personal cars, taxis, trucks, and buses) from a nadir (straight-down) camera view in Iraq. This system achieves a mean AP (mAP) of 88.25%.

In [10], the authors propose a system based on YOLOv4 and DeepSORT that incorporates a trajectory association module to mitigate ID switches caused by camera inclination and the resulting vehicle occlusions in Vietnam cities. The counting module treats all bounding boxes sharing the same tracking ID as a single trajectory. Their detection approach achieves a mAP of 97.8% for motorcycles, cars, buses, and trucks. In [1], a vehicle counting framework based on automatic ROI selection, YOLOv8, and DeepSORT is proposed, enabling bidirectional counting of people and vehicles at park entrances along virtual lines. In [18], a counting system based on YOLOv5 and OCSORT is proposed, in which both traffic motion directions are considered. This method

Fig. 1. Sample of objects from our dataset of interest, grouped by class. (A) Motorcycle, (B) Pedestrian, (C) Bike, (D) Truck, (E) Bus, (F) Car, (G) Van, (H) Moto-taxi.

reaches a mAP of 69.5% in the detection task, and a counting accuracy of 94.4%. In the context of drone-based traffic monitoring in Mexico, [12] have developed a dataset and an automatic counting system using videos of urban traffic captured by drones in the region of Yucatán, Mexico. The counting system comprises sequential modules for detecting, tracking, and counting eight object categories: car, truck, bus, van, motorcycle, bike, moto-taxi, and pedestrian (see Fig. 1). Using the dataset created in [12], and considering the same objects classes, [6] describes an automatic counting system designed to count vehicles and pedestrians crossing a segment specified by the user (see Fig. 2). The counting system reports a weighted average accuracy of 80.38%. The experiments have showed very promising results but have also revealed some limitations of the method. Among them, the instability of the video due to the drone motion can lower the accuracy of the counting, as described in the sect. 1.2.

1.2 Contribution of this work

Open-air video capture is subject to weather conditions, which can affect subsequent processing and analysis tasks. For instance, the presence of clouds can create variations in the lighting of the video frames, which can affect object detection and tracking. Moreover, wind currents can compromise the stability of the drone, resulting in variations in the scale, rotation, and translation of the captured video frames, altogether with motion blur.

Fig. 2. First frame of the different videos of the dataset used in this paper, with some regions of interest, represented by colored segments

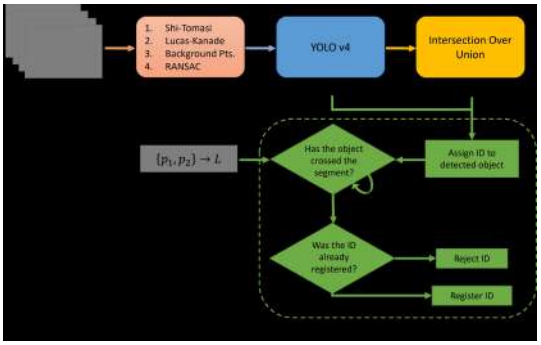

Fig. 3. The proposed counting system modules: Stabilization, Object Detection, Object Tracking, Counting

In particular, areas of interest may not be captured for significant periods of time. While many drones incorporate a mechanical stabilizer (e.g., a Gimbal) or flight modes that return them to a preset position, there is still a moment during which the drone may not have captured some areas of interest. Drone instability, and consequently video frame instability, can drastically affect the results of any counting system.

Video stabilization is the process of registering video frames to eliminate undesirable camera movements such as shakes, tremors, and yaws, ensuring that the video appears stable. In this paper, we improve the results obtained in [6] by inserting a video stabilization module in the counting system, at the beginning of the pipeline (see Fig. 3). Video stabilization is obtained by estimating homographies that approximate the transformations between video frames.

We test different approaches based on the works reported in [5, 3]. The method developed in [5] detects characteristic points (CP) in each frame of the video and estimates a homography matching the characteristic points between consecutive frames using the RANSAC method [4]. In [3], the estimation of the homography relies on a new similarity measure between two frames called Enhanced Correlation Coefficient (ECC).

Experiments show that the approach in [5] achieves better results in terms of execution time and quality of the stabilization, from which we include the method [5] in our counting system.

Then, we compare the performance of the proposed counting system with that in [6] (i.e., without video stabilization), using precision and weighted precision metrics. For these experiments, we have used the same dataset[1] as the one used in [6], and described in sect. 4.2.1. The results show an average improvement of 2.86% in precision and 0.9% in weighted precision, confirming the importance of including a stabilization module in the counting system.

The outline of the paper is the following. In sect. 2, we review the modules forming the counting system [6]. In sect. 3, we give details about the two aforementioned video stabilization methods. Experiments are described in sect. 4 and sect. 5 presents conclusions and future works.

2 Previous Work

In this section, we describe the counting system from [6], to which we refer for more details. The counting system contains three of the four modules of the proposed system: Object Detection, Object Tracking, and Counting (see Fig. 3).

Fig. 4. Objection detection with YOLOv4 at two different frames of the video

Fig. 5. Object tracking based on the IoU metric. The red segment indicates the region of interest provided by the user where crossing objects are counted. The tracked object is depicted with a yellow ellipse. From top to bottom: different steps towards the counting of an object

Fig. 6. Automatic counting system at a given frame of the video 0362

2.1 Object Detection

The object detection module of the counting system is based on the work in [12], where different deep learning-based models have been tested on a similar dataset and with the same classes of objects considered in this work. Among all the methods tested, namely Faster-RCNN, and several versions of YOLO (YOLOv3, YOLOv4, YOLOv5, YOLOv7, and YOLOv8), the authors have shown that the best results are obtained with YOLOv4.

Hence, the model used in the object detection module in [6] is also YOLOv4. The output of the object detection module are bounding boxes delimiting the detected objects in each video frame (see Fig. 4), where the classes are distinguished according to the color of the bounding boxes.

2.2 Object Tracking

Once the objects have been detected in each frame of the input video, the corresponding detections have to be associated into consistent tracks with unique IDs. These IDs allow to distinguish the different objects of the scene and to reconstruct their trajectories. In particular, any time one of these trajectories crosses a segment of interest implies a counter increment. Note that, for this tracking task, performing the video acquisition with UAVs reduces significantly the risk of target occlusion, which is frequent with more traditional monitoring setups. This has also motivated us in basing the tracking algorithm on the intersection over union (IoU) metric between bounding boxes.

Fig. 5 illustrates the object tracking module of the counting system. First, the object is detected (top image), then, if its movement exceeds a certain threshold, it is assigned an ID. Finally, it possibly crosses the region of interest (bottom image).

2.3 Counting

The automatic counting system increases the number of occurrences of an object class each time the bounding box of a detected moving object in this class intersects the region of interest. Fig. 6 shows the result of the automatic counting at a given frame of the video, where the counter located in the top-left corner indicates the number of objects per class that have crossed the region of interest up to the current frame.

[1] https://github.com/MemoJmz/YucaMex-MOCS

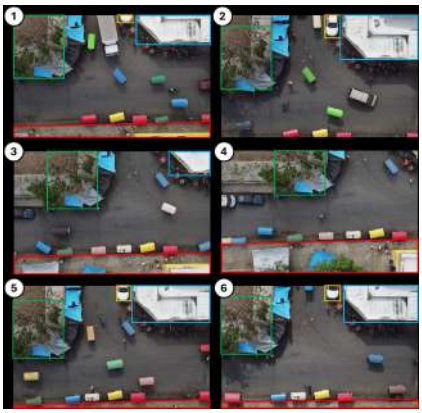

Fig. 7. Instability of the video capture due to the drone motion. The positions of the objects can vary significantly among the different frames of the video

3 Video Stabilization Module

3.1 On the Instability of the Videos

As mentioned in the Introduction, weather conditions can strongly affect the drone hovering flight and the video capture. To illustrate this phenomenon, we consider the video 0362, which is the one providing the greatest instability among all the videos of the dataset. We select four static objects of the scene: a tree, a car, a building, and a sidewalk delimited by green, yellow, blue, and red bounding boxes, respectively. Fig. 7 depicts six different frames of the video and the bounding boxes delimiting the four objects in each frame.

We can observe that the positions of the bounding boxes vary *significantly* between the frames, and even disappear in some of them.

3.2 Stabilization based on Characteristic Points

This method was originally designed for the purpose of video augmentation, in the context of videos taken from unstable cameras [5]. Denoting the first frame of the video by f^0, this method stabilizes the subsequent frames $f^t, t > 0$, with respect to f^0, considered as the reference frame. More precisely, this is a six-stage iterative procedure that stabilizes the different frames according to their order of appearance in the video, that is, f^{t-1} is stabilized before f^t. We detail the different stages of the method in what follows.

1. Detection of corner points in the reference frame. The corner points in f^0 are determined by the Shi-Tomasi method [13]. From the set of corner points, a subset of M points is extracted, denoted by P^0. As in [5], we consider $M = 1000$ points.

2. Tracking of the corner points of the reference frame. Tracking the corner points from the reference frame throughout the video generates a set P^t for each frame f^t. At any $t > 0$, we get P^t by applying the pyramidal version of the Lucas-Kanade optical flow method [8] to P^{t-1}. Note that P^t is refined later in stages 4 and 5.

3. Homography estimation. Assuming that the transformation between any pair of frames is well-approximated by a homography (which holds well in the case of aerial views), the homography $H_{0,t}$ between f^0 and $f^t, t > 0$, is determined as follows. First, the homography $H_{t-1,t}$ between f^{t-1} and f^t is estimated, then $H_{0,t}$ is determined recursively by:

$$H_{0,t} = H_{t-1,t} H_{0,t-1}. \tag{1}$$

With $H_{0,0}$ being the Identity. A two-stage procedure is applied to determine $H_{t-1,t}$: (i) A first estimation of $H_{t-1,t}$ is obtained using the RANSAC method [4] between the elements of the set P^{t-1} and their corresponding elements in P^t (see stage 2); (ii) the estimate $H_{t-1,t}$ is refined using the Levenberg-Marquardt method.

4. Validation of the corner points. To increase the accuracy of $H_{0,t+1}$ in the next iteration, the set P^t constructed in stage 2 is refined in stages 4 and 5. In stage 4, a subset Q^t of P^t is constructed, with the aim of removing the points in P^t that are outliers with respect to the homography $H_{t-1,t}$. In other words, the set Q^t is defined as

$$Q^t = \left\{ x \in P^t, \min_{x' \in P^{t-1}} \| x - \mathcal{C}\left(H_{t-1,t} \hat{x}'\right) \|^2 < d_1 \right\}. \tag{2}$$

Where $\mathcal{C}(\cdot)$ is the operator that transforms a point from homogeneous coordinates to cartesian coordinates and $\hat{x}'$ are the homogeneous coordinates of the point x'. As in [5], the value of the threshold d_1 we take here is 2.

5. Update of the corner points. Let us consider a set R^t of $M - \#Q^t$ corner points in f^t determined by the Shi-Tomasi method and such that:
(i) $x \in R^t \Longrightarrow x \notin P^t$.
(ii) $x \in R^t$ is far enough from any point in Q^t, i.e.

$$x \in R^t \Longrightarrow \min_{x' \in Q^t} \|x - x'\|^2 > d_2. \tag{3}$$

We take here $d_2 = 5$ and P^t is updated using

$$P^t := Q^t \cup R^t. \tag{4}$$

The set P^t contains M elements.

6. Iterative step. Apply stages 2,3,4,5 with $t := t + 1$ to determine $H_{0,t+1}$, from which the stabilization of the next frame f^{t+1} is derived. Repeat until the last frame of the video.

3.3 Stabilization based on Enhanced Correlation Coefficient (ECC)

3.3.1 The Original Method

Given two images I_r and I_d of K pixels whose coordinates are $x_1, \cdots, x_K$, the method aims to construct a homography that maps as much as possible I_d onto I_r. The method relies on the minimization of a pixel-based objective function.

Given a deformation $T(.;p)$ of parameters p, and $y_k(p) = T(x_k;p), k = 1, \cdots, K$, the reference vector i_r and the deformed vector i_d are

$$i_r = \begin{pmatrix} I_r(x_1) \\ I_r(x_2) \\ \vdots \\ I_r(x_K) \end{pmatrix} \qquad i_d(p) = \begin{pmatrix} I_d(y_1(p)) \\ I_d(y_2(p)) \\ \vdots \\ I_d(y_K(p)) \end{pmatrix}. \tag{5}$$

Then, from i_r and i_d we derive the centered vectors $\overline{i_r}$ and $\overline{i_d}$, through

$$\overline{i_r} = i_r - \frac{1}{K} \sum_{j=1}^{K} I_r(x_j) \tag{6}$$

and

$$\overline{i_d} = i_d - \frac{1}{K} \sum_{j=1}^{K} I_d(x_j), \tag{7}$$

that is we subtract from i_r and i_d their means.

Algorithm 1 ECC Version 1

Require: f^0 y f^t ▷ Initial and current frames
Ensure: H_{Total} ▷ Global homography
while f^t **do**
$H_{total} = FindTransformECC(f^0, f^t)$
⋮
end while

The accuracy of the mapping $T(.;p)$ is given by

$$ECC(p) = \left\| \frac{\overline{i_r}}{\|\overline{i_r}\|} - \frac{\overline{i_d}}{\|\overline{i_d}\|} \right\|^2. \tag{8}$$

Note that ECC is invariant to changes of:
- contrast, as the means of $\overline{i_r}, \overline{i_d}$ is 0,
- luminance, as $\overline{i_r}, \overline{i_d}$ have been normalized.

In practice, the minimum of (8) is numerically approximated by means of an iterative descent numerical scheme. We use the implementation of the algorithm available in the OpenCV 3 library named *FindTransformECC* [3].

3.3.2 Application to Video Stabilization

We propose two strategies to stabilize a video based on ECC, which consist of determining the homography that best maps the current frame onto the initial frame. On the one hand, ECC (Version 1) directly calculates the homography between the current and the initial frames (see Algorithm 1). As the CP method described in sect. 3.2, ECC (Version 2) first estimates the homography between two consecutive frames, from which the homography between the current and the initial frames is derived by recursivity (see Algorithm 2).

4 Experiments

4.1 Video Stabilization Methods

In this section, we compare the video stabilization methods CP described in sect. 3.2 and ECC described in sect. 3.3 from both quantitative and qualitative viewpoints, by comparing their execution times and visual results.

Algorithm 2 ECC Version 2

Require: f^t y f^{t+1} ▷ Consecutive frames
Ensure: H_{Total} ▷ Global homography
$H_{Total} \leftarrow Id$
$H_{temp} \leftarrow Id$ ▷ Homography between f^t y f^{t+1}
while f^{t+1} **do**
$H_{total} = H_{temp} * H_{total}$
$H_{temp} = FindTransformECC(f^t, f^{t+1})$
⋮
end while

Table 1. Statistics of the execution times of the different stabilization methods tested on the video 0362 to construct an homography between consecutive frames. The total number of homographies calculated was 9613.

	CP	ECC V1	ECC V2
Mean	0.3056	1.7127	2.1237
Std	0.0873	0.5776	0.4205
Min	0.0899	1.3240	1.3802
25%	0.3117	1.3694	1.7944
50%	0.3194	1.4031	1.9112
75%	0.3627	1.9173	2.5071
Max	0.3889	5.4331	3.3213

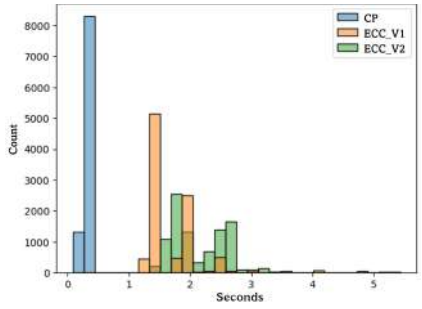

Fig. 8. Histograms of the time (in seconds) required by each stabilization method.

4.1.1 Execution Times

Fig. 8 shows the histograms of the times (in seconds) required by each method to obtain the homography between two consecutive frames. The histogram of CP method is depicted in blue, whereas the one of ECC method is depicted in orange (Version 1) and green (Version 2). Note that, for each method, we have removed the outliers corresponding to values less than the 5% quantile and more than the 95% quantile.

We can observe a significant difference among the histograms of the CP and ECC methods, which tells us that the execution of CP method is faster and also has less variability than ECC method (both versions). These results are validated by the statistics in Table 1. Note that the total execution time of each method for the video tested is:
- *CP*: ≈ 41:35 minutes
- *ECC Version 1*: ≈ 3:30:02 hours.
- *ECC Version 2*: ≈ 4:01:32 hours.

4.1.2 Visual Results

Fig. 9 shows a visual comparison between the two stabilization methods. We observe that both CP and ECC (Version 1) provide acceptable results, the result of CP being slightly better. On the other hand, the method ECC (Version 2) fails in this case.

4.1.3 Interpretation of the Results

From the results obtained in the experiments described in sect. 4.1.1 and sect. 4.1.2, we can deduce that the method CP gives the best results. Indeed, while CP provides slightly better visual results than ECC (Version 1), its execution time is much shorter. The latter may be due to:
(i) ECC requires all the points of the image, whereas CP only requires a subset of them.
(ii) ECC relies on an optimization problem whose solution is obtained from an iterative scheme.

Hence, we have chosen the CP method as the stabilization module of our counting system.

4.2 Automatic Counting System

The aim of the experiment described in this section is to show that the results in [6] can be improved by adding a stabilization module to the automatic counting system.

Fig. 9. Visual result of the two stabilization methods applied to the video 0362. Counterclockwise from top: CP, ECC (Version 1), ECC (Version 2)

4.2.1 Description of the Experiment

Our dataset contains 9 videos of traffic situations on road intersections, in different urban areas of the state of Yucatán, in México. The drone captured the videos in a hovering configuration, at a height ranging from 30 to 70 meters. The videos have a 3840×2160 resolution and their duration range from 3 to 5.5 minutes; they are split in up to three segments (called $\mathbf{S}_1, \mathbf{S}_2$ and $\mathbf{S}_3$), in function of the traffic flow. In the following, the evaluation is done by applying the proposed counting algorithm on one of these segments (see Fig. 2).

To provide a *ground truth*, we asked six people to perform a manual counting. This way, we have gathered three counting values per segment, and we use the corresponding average value as the ground truth. These ground truth values are presented in Table 2. In the following, we provide the results of counting experiments where we apply our counting algorithm to sub-sampled versions of the video sequences. We vary the sub-sampling factor through $j = 1, 2, ..., 5$. Note that the tracking difficulty increases when j gets higher (the motion amplitude of the objects between one frame and the next gets larger).

4.2.2 Evaluation Metrics

As an indicator for the performance of the counting algorithm, we focus on two metrics: *precision* and *weighted precision*.

For a sub-sampling factor j, a class i and a number K of test videos, we define the *precision* of the automatic counting algorithm as:

$$\bar{P}_{ij} = \frac{1}{K} \sum_{k=1}^{K} P_{ij}^k, \tag{9}$$

where the precision for video k is

$$P_{ij}^k = \left(1 - \frac{|C_{ij}^k - \mu_i^k|}{\mu_i^k}\right) * 100, \tag{10}$$

measuring how close is the counting value C_{ij}^k estimated by the counting algorithm to the ground truth value μ_i^k.

Similarly, the *weighted precision* assigns weights to the videos in function of the classes prevalence

$$\widehat{Q}_{ij} = \sum_{k=1}^{K} \lambda_i^k P_{ij}^k, \tag{11}$$

where λ_i^k gives the ratio of occurrences of class i over total number of objects in the video k.

For each j, we consider the *average precision* and the *average weighted precision*

$$\overline{P}_j = \frac{1}{\#Classes} \sum_{i \in Classes} \overline{P}_{ij}, \tag{12}$$

$$\widehat{Q}_j = \frac{1}{\#Classes} \sum_{i \in Classes} \widehat{Q}_{ij}. \tag{13}$$

4.2.3 Results and their Analysis

Table 3 shows the precision and average precision for $j = 1, 2, 3, 4, 5$ of the counting system without (top) and with (bottom) stabilization. The results indicate that the best precision is obtained by the counting system that contains the stabilization module in 6 of the 8 classes (*Car, Truck, Motorcycle, Bike, Moto-taxi, Pedestrian*). In one class (*Van*), adding the stabilization module to the counting system reduces the precision. Note that the system with stabilization also outperforms the one without stabilization when comparing their best average precision. Actually, in both cases, the best score is obtained for $j = 4$ (76.76% with

Table 2. For each segment, the ground truth counting value is the class-wise Average of three Manual Counts

	0027			0089	0362			0597			0614			0674	0861		0883	4040	
Class	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_1$	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_1$	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_1$	$\mathbf{S}_2$	**Total**
Car	36	61	39	5	4	3	7	0	9	15	20	0	16	16	8	0	46	33	318
Truck	4	9	5	4	4	0	5	0	2	3	4	0	9	9	4	0	16	8	86
Bus	2	1	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	3	9
Van	1	2	3	0	1	0	1	0	0	2	1	0	1	0	1	0	7	0	20
Motorcycle	10	7	13	2	9	2	12	0	8	6	15	0	7	22	9	0	35	25	182
Bike	3	1	5	1	5	3	3	0	1	1	0	0	1	0	1	0	2	4	31
Moto-taxi	6	7	6	1	38	21	61	0	0	1	0	0	0	8	15	0	17	2	183
Pedestrian	8	0	0	6	23	24	14	4	0	0	0	7	3	6	30	20	1	3	149

Table 3. Precision and average precision of the counting system without (top) and with (bottom) stabilization.

	Car	**Truck**	**Bus**	**Van**	**Motorcycle**	**Bike**	**Moto-taxi**	**Pedestrian**	$\overline{\mathbf{P}}_\mathbf{j}$
$j=1$	**94.92**	73.59	86.11	85.32	**85.98**	**51.48**	91.78	19.12	73.54
$j=2$	94.23	74.98	86.11	85.32	84.83	30.56	**92.11**	18.88	70.88
$j=3$	92.54	70.39	86.11	85.32	85.91	24.35	**92.11**	19.61	69.54
$j=4$	92.91	79.34	86.11	85.32	83.59	41.39	**92.11**	**34.18**	**74.37**
$j=5$	94.44	**81.84**	86.11	85.32	74.62	40.28	91.78	39.96	74.29
	Car	**Truck**	**Bus**	**Van**	**Motorcycle**	**Bike**	**Moto-taxi**	**Pedestrian**	$\overline{\mathbf{P}}_\mathbf{j}$
$j=1$	94.27	**83.88**	86.11	76.98	85.29	51.67	**98.28**	30.81	75.91
$j=2$	95.33	74.31	86.11	77.78	86.73	49.54	92.73	**45.14**	75.96
$j=3$	94.52	81.26	86.11	76.98	89.85	**60.74**	**98.28**	20.92	76.08
$j=4$	**95.44**	76.29	86.11	76.98	**90.24**	47.04	**98.28**	43.69	**76.76**
$j=5$	92.08	73.33	86.11	**83.33**	76.66	36.20	**98.28**	31.64	72.20

stabilization versus 74.37% without stabilization). Finally, by taking the average of $\bar{P}_j$ over j, we obtain a score of 75.38% for the system with stabilization and 72.52% for the system without stabilization, showing that the stabilization makes the counting system more robust to camera motion.

Table 4 shows the weighted precision and weighted average precision for $j = 1, 2, 3, 4, 5$ of the counting system without (top) and with (bottom) stabilization. The results indicate that the best weighed precision is obtained by the counting system that contains the stabilization module in 4 of the 8 classes (*Car, Motorcycle, Moto-taxi, Pedestrian*).

In one class (*Truck*), adding the stabilization module to the counting system reduces the weighted precision. Note that, as in the case of average precision, the system with stabilization

Table 4. Weighted precision and average weighted precision of the counting system without (top) and with (bottom) stabilization.

	Car	**Truck**	**Bus**	**Van**	**Motorcycle**	**Bike**	**Moto-taxi**	**Pedestrian**	$\widehat{\mathbf{Q}}_\mathbf{j}$
$j=1$	**92.14**	82.56	66.67	80.0	**88.46**	54.84	94.54	**84.56**	**80.47**
$j=2$	91.82	82.56	66.67	80.0	86.26	**58.06**	**95.08**	83.22	80.46
$j=3$	91.51	80.23	66.67	80.0	87.36	41.94	**95.08**	81.21	78.00
$j=4$	91.82	82.56	66.67	80.0	74.73	54.84	**95.08**	83.89	78.70
$j=5$	91.51	**84.88**	66.67	80.0	57.69	51.61	94.54	83.22	75.27
	Car	**Truck**	**Bus**	**Van**	**Motorcycle**	**Bike**	**Moto-taxi**	**Pedestrian**	$\widehat{\mathbf{Q}}_\mathbf{j}$
$j=1$	91.19	**83.72**	66.67	75.0	89.01	51.61	**97.27**	86.58	80.13
$j=2$	**93.08**	82.56	66.67	**80.0**	**90.66**	48.39	96.72	83.22	80.16
$j=3$	92.45	**83.72**	66.67	75.0	90.11	**58.06**	**97.27**	88.59	**81.48**
$j=4$	92.77	82.56	66.67	75.0	81.87	45.16	**97.27**	89.26	78.82
$j=5$	91.51	82.56	66.67	**80.0**	59.34	45.16	**97.27**	**91.95**	76.81

also outperforms the one without stabilization when comparing their best average weighted precision (81.48% with stabilization against 80.47% without stabilization).

Finally, by taking the average of $\widehat{Q}_j$ over j, we obtain a score of 79.48% for the system with stabilization and 78.58% for the system without stabilization, which corroborates the results of Table 3, i.e. the stabilization module helps the counting system to be more robust to changes in the number of consecutive frames processed.

5 Conclusion and Future Work

In this article, we have addressed the problem of automatic counting of vehicles and pedestrians in the context of urban traffic. The proposed counting system is an extension of a previous system of ours, to which we have added a stabilization module to reduce the fluctuations between the frames of the video. We have tested two stabilization methods that rely on the estimation of the homography between consecutive frames. We have showed that the method that estimates the homography from the matching of characteristic points using RANSAC provides better results, which has led us to consider that method as the stabilization module of our counting system. The results of the experiments show that the counting performance benefits from the presence of a stabilization module. Further experiments will consist of validating the robustness of the proposed counting system to object detection and tracking by testing the invariance of the counting to segments located at different places of the same streets. The scores we have obtained for the average precision and average weighted precision show that there is still room for improvement. Future works will be devoted to address some limitations of our counting system, such as its lack of robustness with respect to occlusions.

Acknowledgments

The authors acknowledge the support from “Laboratorio Urbano de la Universidad Modelo” through Grant number 321075 from SECIHTI. The

authors acknowledge support from the Supercomputing Center of CIMAT-Mérida “TOOLOK”. The second author acknowledges the “Investigadoras e Investigadores por Mexico” SECIHTI program.

References

1. **Abdelwahab, M. A., Al-Ariny, Z., Fakhry, M., Hasaneen, E.-S. (2025).** Intelligent roi-based vehicle counting framework for automated traffic monitoring. Signal, Image and Video Processing, Vol. 19, No. 11, pp. 903.
2. **Ebrahimi, S. G., Seifnaraghi, N., Ince, E. A. (2009).** Traffic analysis of avenues and intersections based on video surveillance from fixed video cameras. 2009 IEEE 17th Signal Processing and Communications Applications Conference, IEEE, pp. 848–851.
3. **Evangelidis, G., Psarakis, E. (2008).** Parametric image alignment using enhanced correlation coefficient maximization. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, No. 10, pp. 1858–1865.
4. **Fishler, M. A., Bolles, R. C. (1981).** Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, Vol. 24, No. 6, pp. 381–395.
5. **Hernandez-Lopez, F. J., Rivera, M. (2013).** Avscreen: A real-time video augmentation method. Journal of Real-time Image Processing, Vol. 10, No. 2, pp. 453–465.
6. **Jiménez-Frias, G. A., Hernandez-Lopez, F. J., Batard, T., Forti-Sosa, S., Pérez-Pech, E. (2025).** Automatic counting system in a region of interest from videos taken by drones. Proceedings of Mexican Conference on Pattern Recognition 2025, Springer, Cham, pp. 191–200.
7. **Khan, N. A., Jhanjhi, N., Brohi, S. N., Usmani, R. S. A., Nayyar, A. (2020).** Smart traffic monitoring system using unmanned aerial vehicles (UAVs). Computer Communications, Vol. 157, pp. 434–443.
8. **Lucas, B. D., Kanade, T. (1981).** An iterative image registration technique with an application to stereo vision. Proceedings of the 7th international joint conference on artificial intelligence - volume2, Morgan Kaufmann Publishers Inc, pp. 674–679.
9. **Mustafa, N. M., Alizadeh, F. (2025).** Yolo-based approach for multiple vehicle detection and classification using uavs in the kurdistan region of iraq. International Journal of Intelligent Transportation Systems Research, Vol. 23, No. 2, pp. 747–760.
10. **Nguyen, X.-D., Vu, A.-K. N., Nguyen, T.-D., Phan, N., Dinh, B.-D. D., Nguyen, N.-D., Nguyen, T. V., Nguyen, V.-T., Le, D.-D. (2022).** Adaptive multi-vehicle motion counting. Signal, Image and Video Processing, Vol. 16, No. 8, pp. 2193–2201.
11. **Prommool, P., Auephanwiriyakul, S., Theera-Umpon, N. (2016).** Vision-based automatic vehicle counting system using motion estimation with taylor series approximation. 2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 485–489.
12. **Salcedo-Delgado, J. A. (2024).** Conteo automático de vehículos en movimiento a partir de videos tomados por drones. Master’s thesis, Centro de Investigación en Matemáticas, A.C., Nuevo León, MX.
13. **Shi, J., Tomasi, C. (1994).** Good features to track. 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600.
14. **Srijongkon, K., Duangsoithong, R., Jindapetch, N., Ikura, M., Chumpol, S. (2017).** Sdsoc based development of vehicle counting system using adaptive background method. 2017 IEEE Regional Symposium on Micro and Nanoelectronics (RSM), pp. 235–238.
15. **Vu, T., Thai, H. N., Pham, V. N., Vu, H. T., Luong, A. T., Van Luong, T. (2025).** Counting mixed traffic volumes at motorcycle-dominated intersections by using computer vision. International Journal of

Intelligent Transportation Systems Research, Vol. 23, No. 1, pp. 146–164.

16. **Xiang, X., Zhai, M., Lv, N., El Saddik, A. (2018).** Vehicle counting based on vehicle detection and tracking from aerial videos. Sensors, Vol. 18, No. 8.

17. **Youssef, Y., Elshenawy, M. (2021).** Automatic vehicle counting and tracking in aerial video feeds using cascade region-based convolutional neural networks and feature pyramid networks. Transportation Research Record, Vol. 2675, No. 8, pp. 304–317.

18. **Zou, W., Hu, Y., Wang, X., Li, J. (2025).** Yolov5s-fac: enhanced feature association detector for person-vehicle counting in smart park. Signal, Image and Video Processing, Vol. 19, No. 1, pp. 62.

Article received on 25/08/2025; accepted on 25/11/2025.
**Corresponding author is Thomas Batard.*