The CountingApp, or How to Count Vehicles in 500 Hours of Video

Adrien Lessard
Polytechnique Montréal
adrien.lessard@polymtl.ca

Francois Belisle
WSP Canada inc.
Francois.Belisle@wspgroup.com

Guillaume-Alexandre Bilodeau, Nicolas Saunier
Polytechnique Montréal
{gabilodeau,nicolas.saunier}@polymtl.ca

Abstract

This paper proposes a new method for counting vehicles based on video tracking. The process consists of two main steps: tracking vehicles and processing the output with minimal user input, separating the vehicle positions into sets of trajectories, which correspond to the paths drivers can take. The method allows to rapidly analyze videos from road sections and intersections, and yields detailed results in the form of turning movement counts. A large dataset of five hundred hours of traffic videos was processed using this method and the results are promising as mean absolute percentage error (MAPE) can get as low as 14 % depending on the conditions and the quality of the video capture. This paper also discusses the factors that affect counting performance and how to improve counting accuracy.

1. Introduction

Road user counts are one of the most important inputs for transportation projects. They are required to properly plan transportation facilities, to adapt and operate transportation networks and facilities when behavior and demand evolves over time. Traffic counts are necessary at intersections to select whether traffic signals should be installed or not, to properly configure even the most basic fixed time traffic signals and can be used if available in real-time to adjust the time allocated to the demand on each approach and movement. Construction work often has impacts on transportation networks, for example by reducing their capacity: it is then essential to know where traffic flows to evaluate and mitigate these impacts. Counts are also crucial for road safety diagnosis, as they are the most important factor that determines the number of accidents and they must be accounted for to evaluate the impacts of other factors such as various road designs. Counts are therefore indispensable to build and operate facilities according the road users' needs, which in turn lowers the negative impacts of transportation, such as energy waste and pollution.

As a first step, this paper focuses on motorized vehicles. Vehicles are typically counted manually or automatically using various technologies, based on sensors that are installed in or over the roadway. Sensors have various characteristics, with advantages and disadvantages, depending on the needs and context: the most common type of sensor is the loop detector that is installed in the roadway and therefore requires costly maintenance. Other sensors such as radars are easier to install on the side or over the roadway but more expensive. Among these sensors, video sensors are one of the most flexible as they allow both manual and automated data extraction through computer vision algorithms. They are already installed as traffic cameras at many sites, and are otherwise relatively easy to install as they are not in the roadway and therefore do not impact the traffic flow. They are also one of the few sensors that can provide spatial coverage, while most other sensors are limited to counting at a given point on the roadway. Spatial coverage means that richer information such as trajectories and road user behavior can be extracted from the data.

Given the cost and time to manually count road users in video data, researchers in transportation and computer vision have worked together since at least the early 1990s to automate the process [6]. This is still a very challenging task in the general case because: 1) it requires vehicles to be detected properly which is challenging in adverse lighting and weather conditions or with poor camera setup, 2) at intersections and interchanges, vehicles have to be tracked or re-identified at origins and destinations to count vehicle movements, and 3) the process should be fast as counting vehicles is usually done over long periods of times if not continuously.

There are three main methods to count vehicles in videos: 1) when a vehicle goes across a line, 2) when a vehicle is present in a zone and 3) through trajectories, which are then checked for intersections with a line or zone of interest. The most general solution is to track all vehicles since any kind of count, including all turning movement counts (TMC) at intersections, can be then derived from trajectories.

There is considerable amount of research in general purpose detection and tracking, but relatively little work on the application to large scale transportation tasks such as counting. Jie et al. [2] proposed a system based on the line crossing approach. Background subtraction is first used to detect vehicles, and then each vehicle is tracked using the Kanade-Lucas-Tomasi tracker [10] method. Vehicles are counted every time their bounding boxes cross a user defined line. TMCs are not considered. Messelodi et al. [5] also follows the line crossing approach. However, in their case, a polygon is drawn in the scene and each side of the polygon corresponds to an entering or exiting virtual gate. Tracking is performed in a similar way as for the previous approach. They did not consider complex intersections as each side of the polygon has to be assigned as either existing or entering, not both. Mendes et al. [4] proposed an origin-destination counting system based on tracking with optical flow and a zone-based method. The user of the system draws polygons to define the zones. Counts are computed for all vehicles that are crossing a predefined pair of zones (origin-destination). They do not consider TMCs. Zangenehpour et al. [12] count cyclists using feature-based tracking and zones in a similar way. Zhao et al. [13] automatically determine entering and exiting points in the traffic scene (called sources and sinks), from trajectory clustering using a Gaussian Mixture Model. Counting is performed by determining the number of trajectories that are close to a sink or a source. This method does not consider all possible TMCs as some movements may not result in a detected source or sink. Recent work shows improved performance for TMCs, as well as speed and waiting time measurements, thanks to the integration of tracking, counting and a path reconstruction module to reconnect broken tracks of the same stationary vehicles at intersections [11].

In this paper, we propose a semi-automated method for counting vehicles. It is based on two independent components: a fully automated tracker that outputs vehicle trajectories and a semi-automated counting method based on trajectory filtering from user selected areas. The method is designed to limit the amount of user interactions. It is also generic with respect to the tracker and has already been tested with two existing open source trackers developed in the past, Urban Tracker [3] and Traffic Intelligence (TI) [9, 8]. TI is used in this paper as it is used in several transportation applications, it provides good and transferable results [7] and is relatively fast. For the semi-automated counting, we propose a new method based on trajectory filtering. For a path of interest, where counts are needed, trajectories are iteratively removed until only the trajectories corresponding to that path of interest remain. A path is defined as a possible movement of a vehicle from an entrance zone to an exit zone for a traffic scene in the field of view of the camera. Note that road user classification is not part of this study.

The paper is organized as follows. First, the methodology is presented. The results of the proposed method over 500 h of videos are then presented and compared with ground-truth counts. The paper then finishes with the conclusion.

2. Methodology

2.1. Detecting and Tracking Vehicles

The proposed method relies on a tracker to provide trajectories for counting. Obviously, the tracking performance will greatly influence the counting performance. The open source Traffic Intelligence (TI) tracker [9, 8] is chosen to develop and demonstrate the counting method as it is in active use in other projects and its performance on transportation applications has been demonstrated in several projects.

TI is a feature-based tracker, meaning that it relies on the detection of feature points in images and their tracking from one image to the next using the Kanade-Lucas-Tomasi (KLT) tracker [10]. The next step is to group feature trajectories into unique road user trajectories, based on common motion constraints. For that purpose, the method in [1] for vehicle tracking and traffic surveillance on highways was adapted in [9] to all traffic facilities by avoiding to manually specify entrance and exit regions. The main parameters governing feature grouping are the connection and segmentation parameters d_{con} and d_{seg}. Two features i and j are considered to correspond to the same moving road user if their distance $d_{i,j}(t)$ at time t satisfies the following conditions:

$$d_{i,j}(t_0) \leq d_{con} \qquad (1)$$

$$max_{t_0 \leq t \leq t_f} d_{i,j}(t) - min_{t_0 \leq t \leq t_f} d_{i,j}(t) \leq d_{seg} \qquad (2)$$

where t_0 and t_f is the first and last instants of simultaneous tracking of the two features. The challenge is to find the right balance between over-grouping and over-segmentation when d_{con} and d_{seg} are set respectively too large or too small. Even if correctly chosen, two road users within d_{con} and maintaining their relative motion within d_{seg} may be considered as the same road user. One large road user, larger than d_{con}, may not be covered by features that are dense enough for all its tracked features to be grouped together. Recent work has therefore focused on optimization to find systematically better parameters than can be done manually [7].

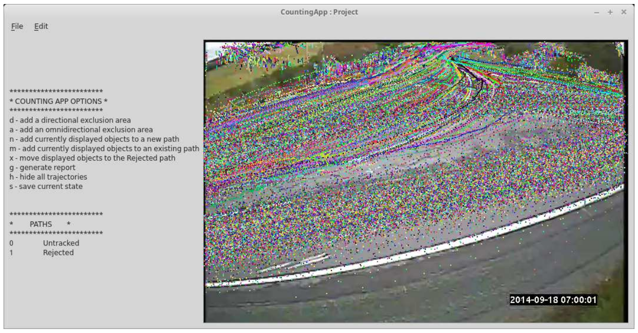
2.2. Counting

The counting part of the video analysis is done through what is named the CountingApp[1]. The required input is a set of vehicle trajectory positions, denoted

$$S = [t, i, x, y] \tag{3}$$

where, for each frame t, (x, y) are the coordinates of the centroid of vehicle i in world coordinates if a homography was used to project the vehicle positions from image space to real world coordinates.

The CountingApp outputs a report with a count for every user-defined path and for every time interval (of user-defined duration). Most of the analysis is achieved through a graphical user interface, presented here in Figure 1. The application window is divided in two parts. The left part is a standard command list, allowing for textual inputs and outputs. The right part consists of a single video frame for positional inputs and outputs. All vehicle positions are displayed on the video frame at start-up. All positions corresponding to the same vehicle i are displayed with the same color, making a complete trajectory T_i.



```
************************
* COUNTING APP OPTIONS *
************************
d - add a directional exclusion area
a - add an omnidirectional exclusion area
n - add currently displayed objects to a new path
m - add currently displayed objects to an existing path
x - move displayed objects to the Rejected path
g - generate report
h - hide all trajectories
s - save current state


************************
*       PATHS      *
************************
0          Untracked
1          Rejected
```

Figure 1. The CountingApp start-up screen showing all trajectories on a video frame background

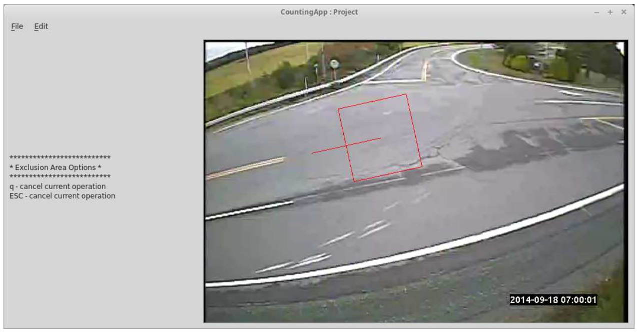
Paths for which vehicles will be counted are defined as a set of vehicle trajectories, starting with all trajectories and excluding the ones that are not part of the current path of interest. During the analysis, the user's work-flow is as follows (initializing $j = 0$):

1. Exclude trajectories T_i until all remaining trajectories belong to the same path P_j

2. Name the path P_j and assign the displayed trajectories to it

3. Display all remaining trajectories, i.e. excluding the trajectories already assigned to any of the paths P_0 to P_j, increment j and go back to 1.

Excluding trajectories is achieved by either an exclusion box or a directional exclusion box, defined by the user on the screen. Any number of exclusion boxes can be used. All trajectories crossing an exclusion box will be removed from the display. A directional exclusion box (see Figure 2) is similar to an exclusion box, but with an added directional vector $\vec{d}$ of unit length. All trajectories crossing the box are removed from display if $\vec{t} \cdot \vec{d} > \cos\theta$, where $\vec{t}$ is a unit vector obtained through a linear regression of n points of the trajectory in the area of the box, and θ is the maximum angle accepted between $\vec{t}$ and $\vec{d}$.



```
*************************
* Exclusion Area Options *
*************************
q - cancel current operation
ESC - cancel current operation
```

Figure 2. An exclusion box defined by the user

At some point, the user might find trajectories that correspond to an already defined path. The CountingApp provides a feature for merging displayed trajectories with an existing path. Finally, some trajectories may be tracking errors or be meaningless to the counting process. They can be excluded permanently from the application.

3. Experiments

3.1. Experimental Methodology

To validate our vehicle counting method, we used ten sites with ten different camera views totaling about 100 Gb of video data or 539 h of videos representing five intersections and five road sections (see Table 1 for details). Videos are taken in the province of Québec, Canada, in both rural and urban settings. This represents a small subset of a larger database representing 2000 h of traffic videos to be used as a benchmark for tracking and counting problems. The original videos were not intended to be part of a scientific study but were made by an engineering firm for particular clients in need of specific traffic counts. An important consequence of the nature of the videos is that respecting installation constraints was more important to the original user than having a camera view that would yield better results by the tracker and the counting application. This can easily be changed in later tests and can have a significant impact on TMC results. Also, the videos have been shot with multiple cameras with varying but generally low image quality.

In traffic studies at road sections, counts are generally

[1] Available under an open source license at `https://bitbucket.org/Alpheratzz/countingapp`.

ID	Principal Street	Secondary Street	Town	Type	# Mvts	# Hours	# Vehicles
004	R-108	Bretelle 55 S	Ste-Catherine de Hatley	Intersection	4	12	4944
026	Campus entry	12e Avenue N	Sherbrooke	Intersection	6	12	10378
027	R-216	Duplessis	Sherbrooke	Intersection	6	12	5997
028	R-116	Gouin	Richmond	Intersection	12	12	7276
029	A-720	St-Laurent	Montréal	Section	2	126	284604
030	Bretelle A-20-E	Angrignon	Montréal	Section	1	168	64617
032	rue Notre-Dame O	Entrée A-20 O	Montréal	Section	1	168	4904
054	Chauveau	Bretelles A573 N	Québec	Section	3	13	26199
057	des Métis	Bretelles A-73	Québec	Section	2	4	3637
060	du Lac	Léo-T.-Julien	Québec	Intersection	6	12	18032

Table 1. Description of the ten video data collection sites (# means number and # Mvts is the number of vehicle movements to count)

done for each direction of traffic, amounting to two paths. At intersections, paths of interests are usually all possible movements between all origins and destinations of the intersection (within the field of view), including U-turns. A four-legged intersection with traffic in both directions thus has 16 possible movements, from each origin going straight (or "thru"), left, right or doing a U-turn.

Two metrics are used to evaluate the quality of the counting results for each site and movement, the Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE) as defined below,

$$RMSE = \sqrt{\frac{1}{n_{meas}} \sum_{k=1}^{n_{meas}} (GT_k - N_k)^2} \tag{4}$$

$$MAPE = \frac{1}{n_{meas}} \sum_{k=1}^{n_{meas}} \left| \frac{GT_k - N_k}{GT_k} \right| \tag{5}$$

$$\tag{6}$$

where n_{meas} is the number of 15-min intervals for a given site and movement, N_k and GT_k are respectively the number of vehicles counted automatically and the ground-truth number of vehicles for time interval k. At the site level, both RMSE and MAPE are averaged over all movements. A Weighted Absolute Percentage Error (WAPE) is computed over all sites as the average site MAPE weighted by the ground-truth number of vehicles at each site. For ground truth, we used the results of a commercial provider of traffic counts that guarantees a 95 % accuracy rate. Random checks were made to ensure the reliability of this claim.

3.2. Results

Results are presented in the Table 2. They represent RMSE (and MAPE) for vehicle counts for every 15 min interval for all possible movements at each site. As mentioned earlier, each site has different movements and video data was collected for various amounts of time, varying from 4 to 168 hours. An average for each site is presented in the last column of the Table.

The overall average RMSE and MAPE are respectively 66 vehicles and 39 % per 15 min measurement period, whereas the global WAPE is 56 %. However, three sites stand out with particularly low results: 026, 029 and 032. As can be seen in figures 8, 5 and 9, these three sites exhibit highly problematic camera angles and low video quality weighing down significantly on the results. Without these three sites, the WAPE is 23 %. Generally speaking, MAPE varies for different movements and sites, from 7 % (004 Eastbound Right) to 181 % (026 Eastbound Left). This wide variation of errors testifies of a complex relationship between errors, general camera setup and calibration, vehicle tracking and trajectory grouping by the CountingApp. Error sources will be discussed in the next section.

As such, there seems to be no correlation between errors and movements in sites. In fact, principal movements (thru movements) or turning movements (left and right movements) do not seem to show widely diverging errors. Error is closely related to camera viewpoints and characteristics. Errors are concentrated on movements where the tracker cannot properly follow vehicles or on trajectories that cannot be easily separated by the CountingApp. However, the movements with the lowest errors are thru movements close to the camera focus. Moreover, there does not seem to be widely diverging errors between sections and intersections where vehicle operate differently: on sections, vehicles can be more easily tracked, whereas vehicles are expected to stop in intersections and tracking is lost. Once again, camera characteristics prevail over tracking and counting parameters.

Finally, the absolute percentage errors for all 15-min intervals are plotted as a function of traffic flow to evaluate if a higher number of vehicles yielded a higher error (see figure 3 for a section and figure 4 for an intersection). This does not seem to be the case. In figure 3, the error seems to be an inverse function of flow for all movements, which

	Southbound			Westbound			Northbound			Eastbound			AVG
ID	Right	Thru	Left	Right	Thru	Left	Right	Thru	Left	Right	Thru	Left	
004					6 (16 %)	1 (16 %)				1 (7 %)	13 (17 %)		7 (14 %)
026	3 (5 %)		5 (25 %)	24 (58 %)	8 (18 %)						40 (85 %)	52 (181 %)	26 (62 %)
027					6 (14 %)	4 (34 %)	9 (49 %)		2 (11 %)	7 (34 %)	28 (68 %)		13 (36 %)
028	9 (100 %)	1 (23 %)	2 (35 %)	5 (84 %)	16 (35 %)	1 (13 %)	3 (35 %)	3 (75 %)	8 (38 %)	4 (20 %)	12 (27 %)	3 (64 %)	6 (46 %)
029											505 (71 %)		505 (71 %)
030											17 (20 %)		17 (20 %)
032				6 (69 %)									6 (69 %)
054							14 (14 %)		8 (8 %)	24 (56 %)			17 (26 %)
057		54 (30 %)						11 (17 %)					39 (23 %)
060	5 (9 %)		3 (9 %)	169 (34 %)	50 (24 %)						36 (26 %)	3 (13 %)	64 (19 %)
All													66 (39 %)

Table 2. RMSE (MAPE) for each movement of the ten video data collection sites. AVG is the average of all movements (except U-turns) for a site. The "Al" row gives the average over all sites (and all movements)

may be expected by definition of MAPE (small errors will be a larger proportion of small traffic flows). However, this relationship does not hold for the principal Westbound thru (in blue) or eastbound thru (in grey) movements where the trend curve is flat across the range of flow values. It should be noted that both movements can clearly be identified from the camera without distortion or occlusion as can be seen in the example frame on the left of figure 3. Additionally, traffic conditions are under-saturated and vehicles can flow freely without congestion throughout the analysis period.

In figure 4, the same overall inverse relationship applies between absolute percentage error and traffic flow. This relationship is emphasized by the trend line of the Eastbound thru movement (in green) that is clearly descending. However, error increases with flow for the Westbound thru movement. In the video frame on the left of figure 4, the Eastbound vehicles go from the right to the top left of the image whereas Westbound vehicles go from the left to the right. Although the Westbound is in clear view of the camera, Eastbound vehicles are caught by the tracker when they have crossed their stop bar, after having regained speed if they have stopped at the traffic light. It should be noted that this intersection exhibits saturated and under-saturated regimes, covering all ranges of possible flow values.

Finally, saturated multi-lane highways have not been tested and should, in all probability, exhibit higher error rates than one lane sections because of repeated stop-and-go motion at different positions for different lanes.

3.3. Discussion

This section provides an overview of the three main error sources: the video data input, the trajectories and the methodology used in the CountingApp.

3.3.1 Video Data Issues

The data used for the experiments was not optimized for computer vision-based counting. For example, three main issues limit the tracker's ability to follow vehicles :

1. **Video Quality.** The video quality is often low because of low resolution and poor lighting conditions often including back lighting;

2. **Camera Calibration.** The cameras are not perfectly calibrated and some images are distorted leading to trajectories that are not as straight as they should be;

3. **Orientation.** The cameras were not necessarily oriented to properly observe the vehicle paths.

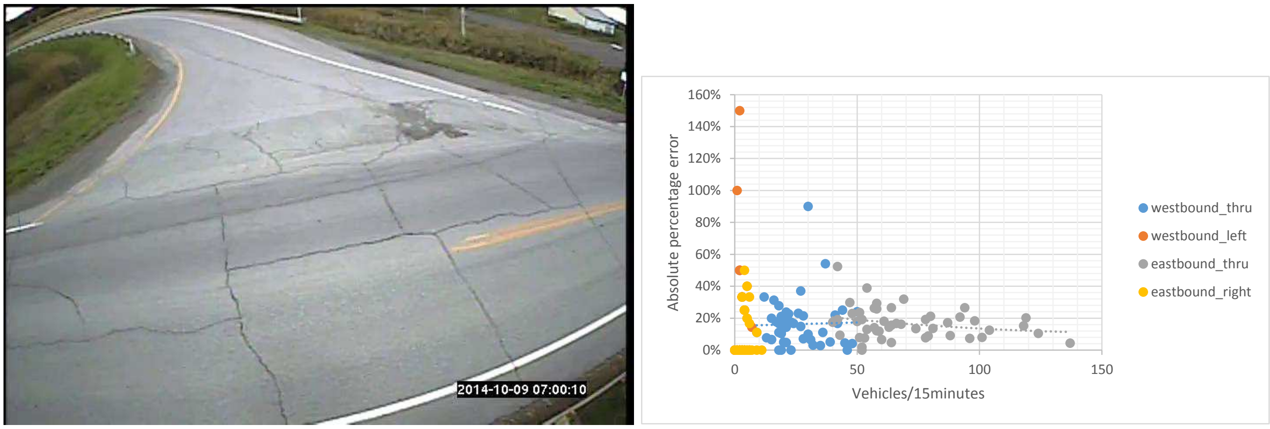


Figure 3. Absolute percentage error for each movement and 15-min interval as a function of flow for site 004 (section), shown on the left

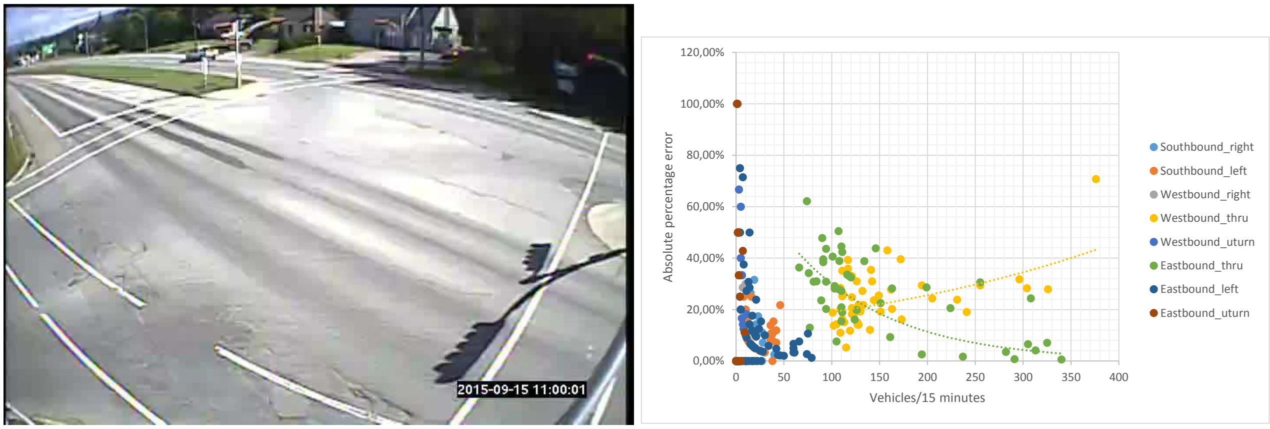


Figure 4. Absolute percentage error for each movement and 15-min interval as a function of flow for site 060 (intersection), shown on the left

Figure 5 is an example of a problematic site. There is back lighting at this particular time and the zone of interest for counting is to the right of the concrete divide, which is invisible in this image.

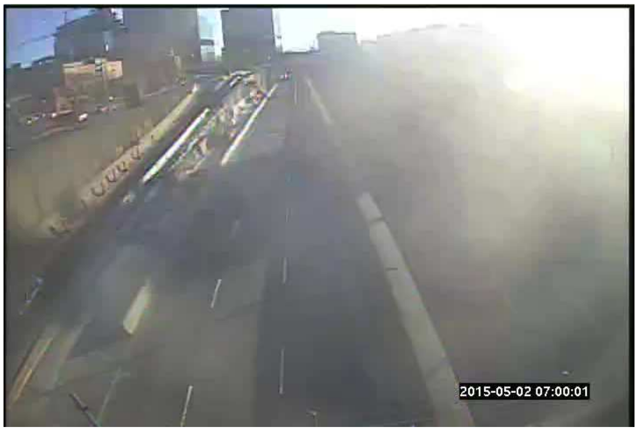


Figure 5. Back lighting effect that limits the counting performance (site 029)

3.3.2 Tracking Issues

There is still no tracker that can operate fully automatically in all traffic conditions [3]. From a counting perspective, there are three main sources of error:

1. **Vehicle classification.** Although no classification has been made in this study, vehicles such as trucks are often considered as multiple vehicles, leading to overcounting.

2. **Occlusion.** Fragmented vehicle trajectories multiply the number of trajectories for the same vehicle, once again yielding higher counts for the same vehicle. Occlusion can be caused by vehicles passing in front of the camera view or by elements in the scene, such as traffic signal posts or other street furniture (see figure 6).

3. **Stopped vehicles.** With TI, trajectories are terminated when a vehicle stops and a new trajectory is created when the stopped vehicle starts to move again (see figure 7). This trajectory fragmentation also yields higher

counts for the same vehicle. In an intersection, this fact can be compensated in the CountingApp by using an elimination strategy, keeping only the trajectories that are directly inside of the intersection (specifically: in movement between the stop bars). That being said, because of certain lane configuration and site-specific realities, certain movements are only tracked when leaving the intersection and are always fragmented.

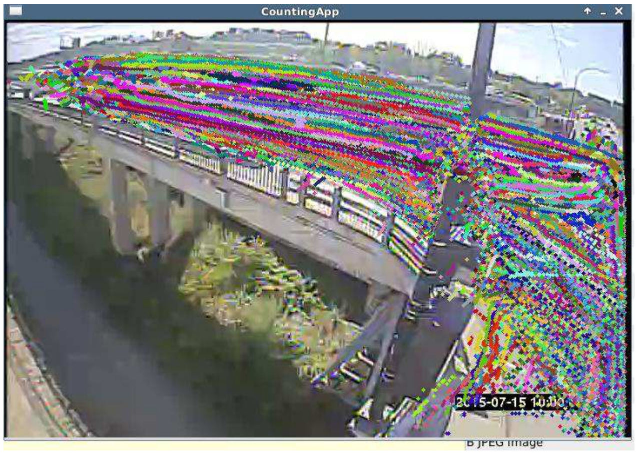


Figure 6. Trajectory fragmentation because of street furniture (site 030)

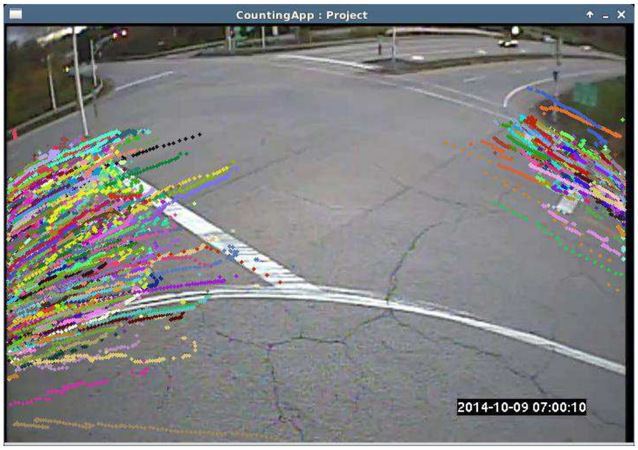


Figure 7. Trajectory fragmentation because of stop-and-go motion of vehicles at the approach of a stop sign (site 028)

3.3.3 CountingApp issues

The CountingApp also faces challenges that have an impact on counting errors, such as:

1. **Noise.** Noise (trajectories of objects that are not vehicles such as trees, shadows, etc.) cannot always be easily separated from vehicle trajectories, leading to over-counting.

2. **Trajectory overlap.** Because of lanes shared by different movements and limited camera viewpoints, trajectories of different movements cannot be isolated and are assigned to the wrong path, yielding higher and lower counts in shared lane counts. An example of this phenomenon is shown in figure 8 where two left-turn lanes are shared with a one lane thru movement. At this stage of the exclusion and assignment procedure, it seems as though the remaining trajectories correspond to the thru movement. However, this is not the case as many fragmented trajectories of stopped left-turning vehicles remain.

3. **Trajectory projection.** Because of camera viewpoints and perspective geometry, some vehicle trajectories can be projected in other lanes than those in which the vehicle actually was. Without seeing the vehicle itself, the application user can easily and wrongly assign a trajectory to a given movement. An example of this misclassification can be viewed in figure 9. In the video of site 032, the benchmark was to count the westbound highway on-ramp entry lane; however, differentiating all westbound trajectories is at best difficult and generally not accurate because of the projection of the on-ramp trajectories projected onto the highway behind it. After having eliminated all Eastbound vehicles, all remaining vehicles are Westbound. To differentiate between the two local Westbound lanes from the Westbound on-ramp and the two Westbound lanes, one must guess where the trajectories of the vehicles fall; without prior knowledge of the kind of vehicles using the ramp (tall-standing trucks or low lying passenger cars), this is at best an inaccurate process.

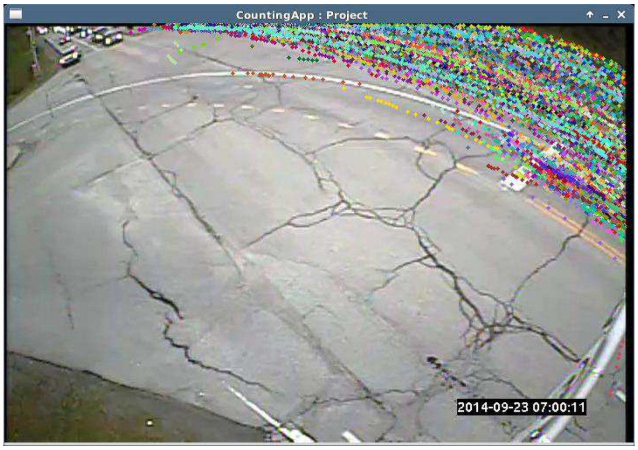


Figure 8. Trajectory overlap between overlapping movements in shared lanes (site 026)

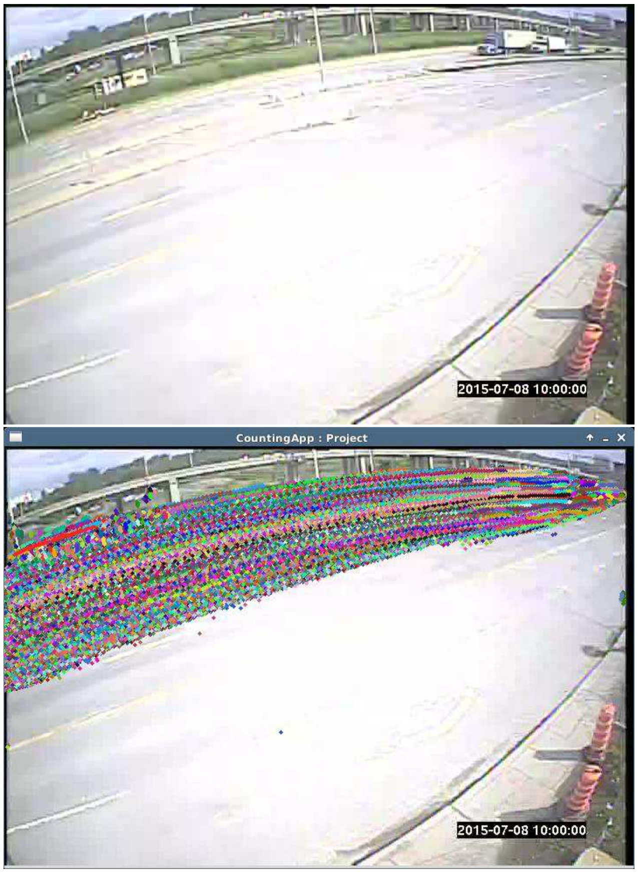


Figure 9. Top: Views of site 032 with, in order from bottom to top, two eastbound lanes, two westbound lanes, one westbound highway entry lane and two westbound highway lanes. Bottom: Westbound highway and entry lanes trajectories.

4. Conclusion

This paper proposed a new method for counting vehicles based on video tracking. A large dataset of five hundred hours of video was processed and the results are promising as the mean absolute percentage error (MAPE) can get as low as 14 % at an intersection depending on the conditions and the quality of the video capture. Since the original videos were not intended to be part of a scientific study, respecting installation constraints was more important to the original user than having a camera view that would yield accurate results by the tracker and the counting application.

Different sources of error were identified in the video data input, the tracker and the counting application in itself. Trajectory fragmentation is a major error factor in the counting application, caused by phenomena such as occlusion or vehicles stopping at stop-lines. A solution to this issue would be to use a different tracker based on background subtraction that ensures vehicle persistence in video frames, such as Urban Tracker [3]. Also, using video data that is collected specifically for automated tracking and counting would help to evaluate the contribution of the tracker and the counting application to the errors.

The next steps in the development of the counting application is the inclusion of vehicle classification and the possibility to automatically group trajectories using clustering algorithms with trajectory similarity measures such as the longest common sub-sequence, respecting physical characteristics such as lane width.

Acknowledgments

The authors would like to thank Polytechnique Montréal for the undergraduate research initiation scholarship UPIR that funded a part of this work. They also wish to thank the anonymous reviewers for their valuable comments.

References

[1] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C: Emerging Technologies*, 6(4):271–288, 1998.

[2] K.-S. Jie and M. Liu. Computer vision based real-time information acquisition for transport traffic. In *Information Acquisition, 2005 IEEE International Conference on*, pages 6 pp.–, June 2005.

[3] J.-P. Jodoin, G.-A. Bilodeau, and N. Saunier. Tracking people, bikes and vehicles at traffic intersections in urban setting. *IEEE Transactions on Intelligent Transportation Systems*, 2016. In Press.

[4] J. C. Mendes, A. G. C. Bianchi, and Á. R. P. Júnior. Vehicle tracking and origin-destination counting system for urban environment. In *VISAPP 2015 - Proceedings of the 10th International Conference on Computer Vision Theory and Applications, Volume 3, Berlin, Germany, 11-14 March, 2015.*, pages 600–607, 2015.

[5] S. Messelodi, C. M. Modena, and M. Zanin. A computer vision system for the detection and classification of vehicles at urban road intersections. *Pattern Analysis and Applications*, 8(1):17–31, 2005.

[6] P. G. Michalopoulos. Vehicle detection video through image processing: the autoscope system. *IEEE Transactions on Vehicular Technology*, 40(1):21–29, 1991.

[7] P. Morse, P. St-Aubin, L. F. Miranda-Moreno, and N. Saunier. Transferability study of video tracking optimization for traffic data collection and analysis. In *Transportation Research Board Annual Meeting Compendium of Papers*, 2016. 16-6810.

[8] N. Saunier. Traffic Intelligence. Online, 2011-2016. Software under the open source MIT License.

[9] N. Saunier and T. Sayed. A feature-based tracking algorithm for vehicles in intersections. In *International Conference on Computer and Robot Vision*, Québec, June 2006. IEEE.

[10] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600. IEEE, 1994.

[11] M. S. Shirazi and B. T. Morris. Vision-based turning movement monitoring:count, speed & waiting time estimation. *IEEE Intelligent Transportation Systems Magazine*, 8(1):23–34, 2016.

[12] S. Zangenehpour, T. Romancyshyn, L. F. Miranda-Moreno, and N. Saunier. Video-based automatic counting for short-term bicycle data collection in a variety of environments. In *Transportation Research Board Annual Meeting Compendium of Papers*, 2015. 15-4888.

[13] R. Zhao and X. Wang. Counting vehicles from semantic regions. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):1016–1022, June 2013.