THE IMPACT OF DIFFERENT VIDEO RESOLUTIONS IN A FEATURE-BASED VEHICLE DETECTION ALGORITHM

Leandro Arab Marcomini
André Luiz Cunha
University of São Paulo
São Carlos School of Engineering

ABSTRACT
Cameras with higher resolutions are more commonly being used in surveillance systems to detect and track vehicles. Higher resolutions cause an increase in processing times on feature-based tracking algorithms. The objective of this paper is to evaluate the impact of video resolution during the detection process on feature-based tracking algorithms. 32 videos were originally recorded in 1080p, totaling 394 recording minutes, and were converted into 5 different resolutions. To evaluate the results, manual counts were compared with automatic counts generated by an algorithm implemented by the author. Results show that detection rate errors are higher on lower resolutions, such as 428 x 240. On the other hand, higher resolution videos used more processing time to complete. In conclusion, videos with an intermediate resolution, such as 704 x 480, are recommended for feature-based tracking algorithms.

RESUMO
Câmeras com resoluções maiores vêm sendo cada vez mais usadas em sistemas de monitoramento de tráfego para detectar e rastrear veículos. A maior resolução causa o aumento nos tempos de processamento dos algoritmos de detecção baseados em *features*. O objetivo deste trabalho é avaliar o impacto da resolução do vídeo durante o processo de detecção de veículos em algoritmos de rastreamento por *Features*. Os 32 vídeos originalmente capturados a 1080p, totalizando 394 minutos de gravação, foram convertidos em outras cinco resoluções avaliadas. A avaliação do algoritmo em cada resolução de vídeo foi feita através da comparação da contagem veicular manual com as saídas automáticas do programa implementado pelo autor. Os resultados mostraram que os erros de detecção foram maiores nos vídeos com baixa resolução, como 428 x 240. Por outro lado, nos vídeos com as maiores resoluções, a maior diferença foi no tempo de processamento. Pode-se concluir que vídeos com resoluções intermediárias (704 x 480) são indicados para algoritmos baseados em rastreamento por *Features*.

1. INTRODUCTION

Contemporary solutions to solve the problems of traffic congestion and issues on traffic networks are based on the availability of data. Better information on what is happening on the network enables for better decision-making by administrators.

The information collected from the network is getting increasingly reliant on traffic surveillance, involving cameras distributed in key points, in a movement to distance itself from manual data collection. Because of that, better vehicle detection and tracking systems are all on demand.

Coupled with the increasing need for surveillance cameras, there was an improvement on the quality of the images created by those devices. Cameras went from a 100 x 100 pixel resolution, on the first digital camera created [Sasson, 2007], to almost ubiquitous smartphone cameras that can record videos at 4K resolutions (3840 x 2169) at 60 frames per second.

The evolution of camera resolutions intensified an already existing problem for algorithms trying to extract information from images: processing time. Although computers have become more powerful, algorithms involving detection of objects are demanding. They usually use sliding windows to find an object, as in the case of SVMs (Support-Vector Machines) [Cortes and Vapnik, 1995] or Neural Networks [Rumelhart et al., 1985], or process images searching for specific combination of pixels, as in the case of Feature Tracking [Shi and Tomasi, 1994].

On this paper, we propose a method to identify whether resolution changes affect performance on a Feature-based Tracking algorithm for the detection of vehicles.

2. FEATURE TRACKING

In a feature-based tracking algorithm, features can be corners of an object, borders, points of interest, or any characteristic that makes the object distinct from the background. In the particular case of tracking vehicles, changes on weather and vehicle overtakes creates cases where occlusion and luminosity changes are common. Because feature detectors abandon the idea of tracking the whole vehicle, the negative effect on the detection on those situations is decreased. [Saunier and Sayed, 2006].

Feature detection, although important, is just the first step that a tracking algorithm must complete. After the detection, the features must be grouped together in what is called a vehicle hypothesis. Grouping is considered a critical point of feature-based tracking [Cavallaro et al., 2005].

Several authors suggest improvements and solutions for the grouping problem, but no technique was able to maintain a good performance under all circumstances. Beymer et al. [1997] were able to achieve a detection rate of 75.2% on North American highways. Coifman et al. [1998], on different traffic conditions and luminosity, were able to detect between 75% and 97% of the vehicles. Collins et al. [2005] had to initialize feature tracking manually, selecting important points of each vehicle, achieving good results.

More recently, Jazayeri et al. [2011] proposed a method using Harris Corner Detection, which attributes different weights to each region of the image based on their intensity changes, to detect features on a vehicle. This technique, coupled together with a line detector and a light intensity peak detector, achieved a detection rate of 86.6% on their tests on different videos. Lin et al. [2012] suggest the use of two distinct classes of features. The first class of features are selected based on a training database, with common features for a vehicle. The second class of features are all border related, detecting features on the edge of vehicles. The algorithm was able to detect 90% of the vehicles on the tested videos. Do and Woo [2016] proposed a method to track vehicles based on the Shi-Tomasi algorithm [Shi and Tomasi, 1994] and on [Lowe, 2004]. The combination of those two methods was successful, but only the tracking of features was analyzed – vehicle tracking was not measured. Shih and Zhong [2017] also used Shi-Tomasi to find borders and corners. Feature grouping was done by using the relative distance between detected features and their algorithm achieved 90% of detection rate.

On this work, we used a similar approach with the one suggested by Shih and Zhong [2017], with Shi-Tomasi being responsible to detect features and the Euclidian distance to group features.

3. PROPOSED METHOD

The hypothesis proposed in this paper is that there is no impact using different video resolutions on a feature-based tracking algorithm to vehicle detection. The method consists in converting the original footage, recorded in 1920 x 1080 pixels, to several different common resolutions. The videos were then used as the input in a feature-based tracking system and all the vehicles were counted for each resolution. The feature-based tracking method on this paper uses a combination of corner detectors [Shi and Tomasi, 1994] and optical flow [Lucas et al., 1981] to detect and

track vehicle features across the frame, implemented and proposed by Marcomini [2018].

The detected features are then grouped by their Euclidian distances and the vehicle is counted as it reaches the end of the region of interest. The result is a comparison of all the automatic counts with the manual count. A flowchart of the method can be seen in Figure 1.

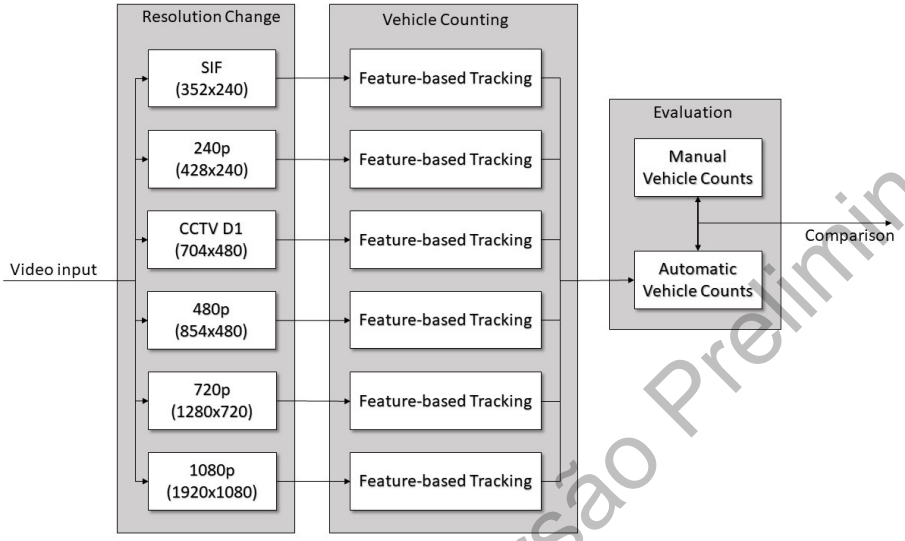

Figure 1: Flowchart of the proposed method.

For a better understanding of the traking algorithm used to create the automatic count of vehicles, a flowchart of the method proposed by Marcomini [2018] can be seen in Figure 2. All videos have their perspective changed, to eliminate the influence of different camera angles. The resulting frame has it's background removed, using an adaptive Gaussian mixture model known as MOG2 [Zivkovic, 2004], and features are detected. All detected features are tracked and grouped in vehicle hypothesis, which is then extracted as data files from the system.

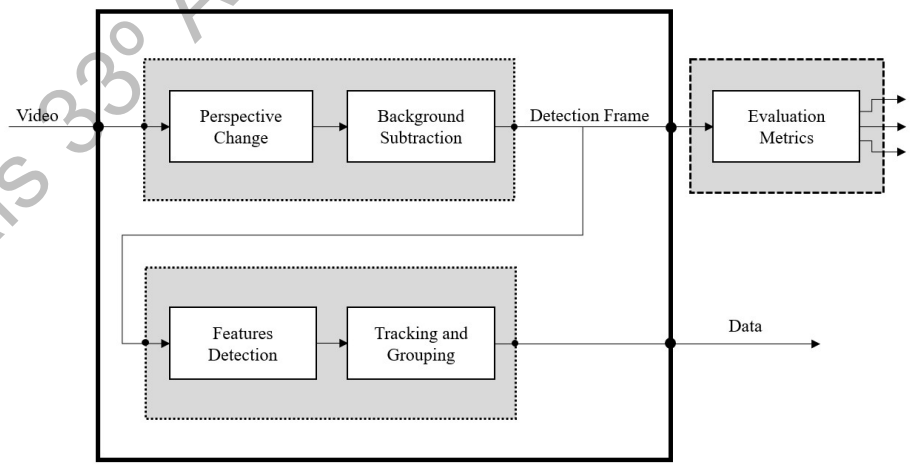

Figure 2: Flowchart of the detection algorithm based on Marcomini [2018].

All functions on the used algorithm are implemented in Python 2.7, using the OpenCV 2.4 library [OpenCV, 2019].

4. DATA

In total, we analyzed 32 videos for each resolution presented in Figure 1, totaling 394 minutes of footage recorded on the same day. The cameras were placed on top of a footbridge, over an avenue, with no intervention on the local traffic and centered on the line dividing lanes. All original videos have a resolution of 1920x1080 pixels, recorded at 30 frames per second (FPS).

Half the videos (16) were recorded with the camera pointing to the same direction of the traffic flow. That way, vehicles appear on the bottom of the frame and vanish on the top. The other 16 videos were recorded against traffic flow, at the top of the same footbridge, at the same time. Vehicles enter the frame on the top and leave on the bottom. Examples of angle and direction of the videos can be seen in Figure 3.

Figure 3: Videos recorded both in the same direction and against the traffic flow.

It was important to keep conditions constant through all recorded videos, so we would be able to isolate the possible effects only of the resolution changes.

5. RESOLUTION CHANGE

Given that the original videos were recorded at 1920x1080 pixels, it was necessary to create other instances of the same files, but with different resolutions. For that purpose, we implemented a method in Python using the functions available in the OpenCV library. More specifically, we used the function cv2.resize(), which takes as parameters the frame to be converted and the new desired size, with the option to choose from a range of interpolations. The default conversion algorithm uses a geometric transformation based on bilinear interpolation OpenCV [2016]. On this conversion, the intensity of the pixel to be scaled up or down, pixel (x,y) on Figure 4, is determined based on four diagonal closest neighbors on a 2 by 2 window, pixels (x1,y1), (x2,y2), (x3,y3), (x4,y4) [CambridgeInColour, 2019].

The hypothesis of this paper is to analyze the effects on vehicle detection when the resolution

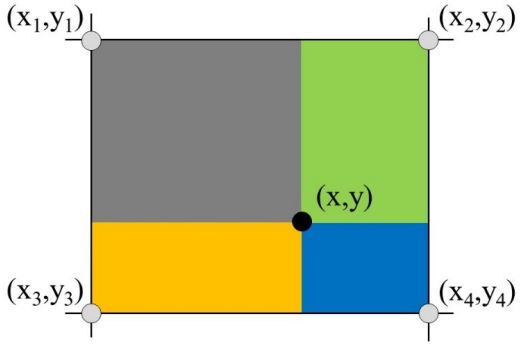

Figure 4: Undetermined intensity pixel (x,y) and closest neighbors.

changes. Therefore, it's important to isolate the changes from other common variables on traffic images, such as lighting and vehicle sizes. By using the same videos, but converted to several different resolutions, we aim to reduce the effects of grouping errors that different vehicles sizes have on feature-based tracking algorithms. Since the only variable that changes between the videos is the resolution, the difference in vehicle counting can be associated with the resolution change.

6. VEHICLE COUNTING

In order to compare the performance of several different resolutions, we manually counted vehicles in all 32 videos and registered the number of vehicles at every minute mark. An example of the procedure can be seen in Table 1.

Table 1: Manual vehicle counts grouped by minute.

	Video 1		Video 2		Video 3	
Time (min)	Frame no.	Vehicles	Frame no.	Vehicles	Frame no.	Vehicles
1	1793	11	1800	13	1774	16
2	3630	12	3640	14	3602	11
3	5395	2	5494	11	5406	17
4	7313	13	7296	13	7247	12
5	8979	11	9127	13	9065	12
6	10837	11	10843	12	10792	15
7	12607	11	12664	12	12573	10
8	14401	10	14410	15	14528	18
9	16205	4	16158	13	16400	16
10	17986	11	18013	17	18036	15
11	19878	13	19802	19	19788	9
12	21581	10	21596	12	21580	11
13	23788	19	23653	10	24193	9
		138		174		171

By using the values for each minute on our manual count to compare with the automatic count for each resolution, we were able to extract an absolute average detection error (using Equation 1) for each video. This absolute average detection error is calculated based only on positive numbers, so if the error observed on a specific minute is negative (automatic count smaller then

manual count), the absolute value will be used. For example, in Table 2, it is possible to notice that all detection errors are positive numbers. Thus, by calculating the average of all values, the total detection error on that video will be 6.5%. By using this method instead of counting the total number of vehicles on the entire video and comparing the total values, we aim to diminish the effect of accumulated errors. For instance, if the algorithm counts one less vehicle at a minute and then counts one extra vehicle later, the total detection error would be 0%. By separating each minute, we compute this localized error in the total.

$$AbsError = \frac{\sum_1^m \frac{n_{man}}{n_{aut}}}{m} \quad (1)$$

where: m = Video duration, in minutes;
n_{man} = Manual vehicle count;
n_{aut} = Automatic vehicle count.

Table 2: Detection errors on one 352 x 240 resolution video.

Time (min)	Frame no.	Manual Count	Automatic 352x240	Detection Error
1	1793	11	11	0.0%
2	3630	12	12	0.0%
3	5395	2	2	0.0%
4	7313	13	14	7.7%
5	8979	11	12	9.1%
6	10837	11	10	9.1%
7	12607	11	12	9.1%
8	14401	10	9	10.0%
9	16205	4	4	0.0%
10	17986	11	10	9.1%
11	19878	13	13	0.0%
12	21581	10	12	20.0%
13	23788	19	17	10.5%
				6.5%

7. RESULTS

In order to easily observe the distribution of detection error of all 32 videos, grouped by resolution, histogram plots can be seen in Figure 5.

Based only on the histograms, it's possible to notice that detection errors tend to concentrate more on smaller values on intermediate and high resolutions, from 704 x 480 to 1920 x 1080, resulting in less detection errors overall.

The descriptive statistics of all detection errors can be seen on the boxplot graphic in Figure 6. Each boxplot represents data from all videos of one resolution. To calculate the values, such as first quartile, average resolution error, and third quartile, data from all 32 videos of each resolution was used, i.e., to extract the average resolution error value for the resolution 352 x 240, we summed all detection errors of all videos on that resolution and divided the value by the number of videos. The equation used can be seen in Equation 2. All different vehicles categories - motorcycles, cars, trucks, busses - are grouped together on the counts, since the objective of

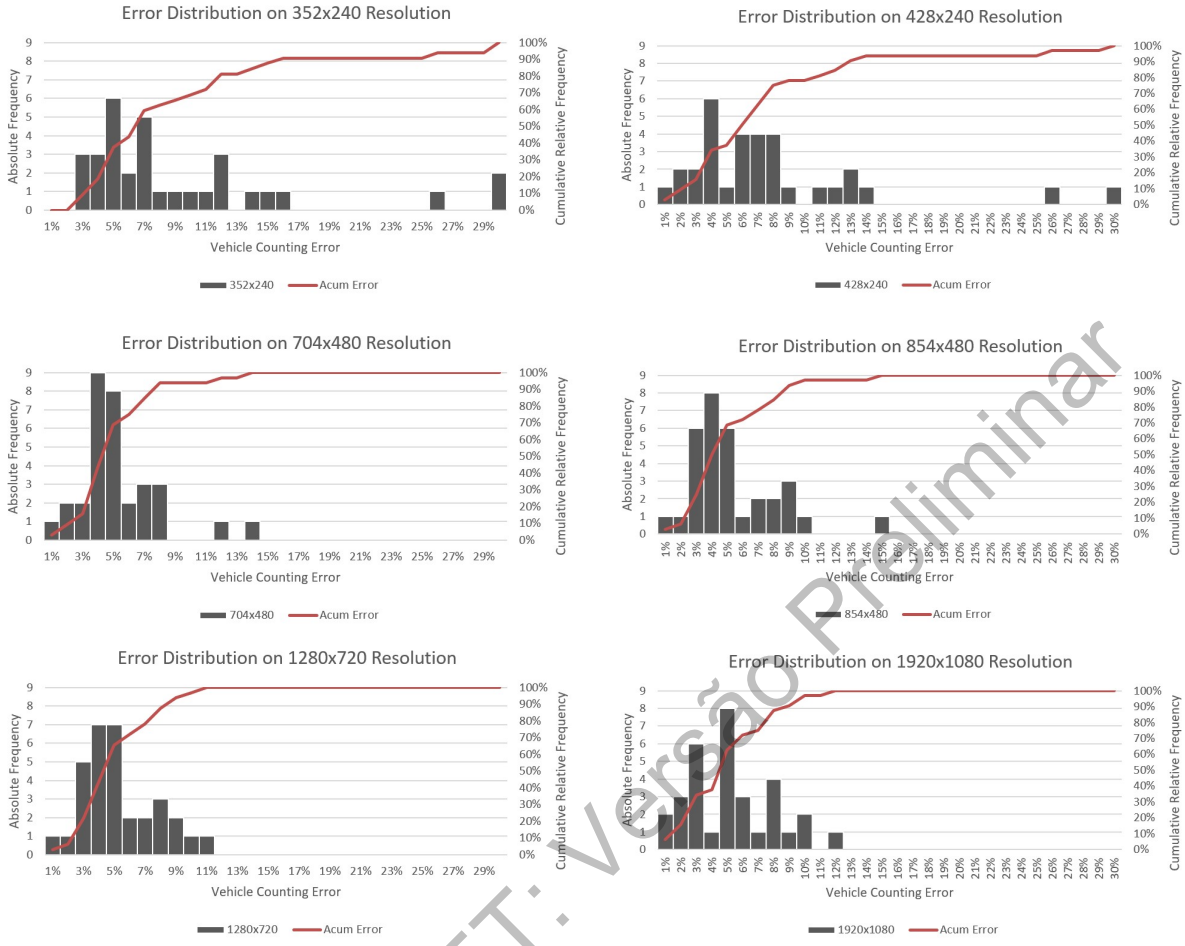

Figure 5: Histogram plots of vehicle counting errors for each resolution.

this paper is to analyze the effect of resolution changes on the overall detection rate, although the variation on vehicle's sizes present a challenge for grouping features, as demonstrated by Marcomini [2018] and others.

$$AvgResErr = \frac{\sum_1^{n_{vid}} AbsError}{n_{vid}} \tag{2}$$

where: $AvgResErr$ = Average detection error of a specific resolution;
n_{vid} = number of videos (32);
$AbsError$ = Absolute average error, from Equation 1.

It is possible to notice that at the lowest evaluated resolution, 352 x 240, 75% (3º quartile) of the encountered detection errors happened below the 11% margin. In other words, in 75% of all evaluated videos, the feature-based tracking algorithm incurred errors that were lower or equal to 11% in the counting process. Although that is not a bad tracking and counting performance, this result is still the worst observed in all evaluated resolutions.

The best result can be seen on the 704 x 480 resolution, where 75% of all detection errors were below 6%. Two other resolutions also presented a satisfying performance. At 854 x 480 and

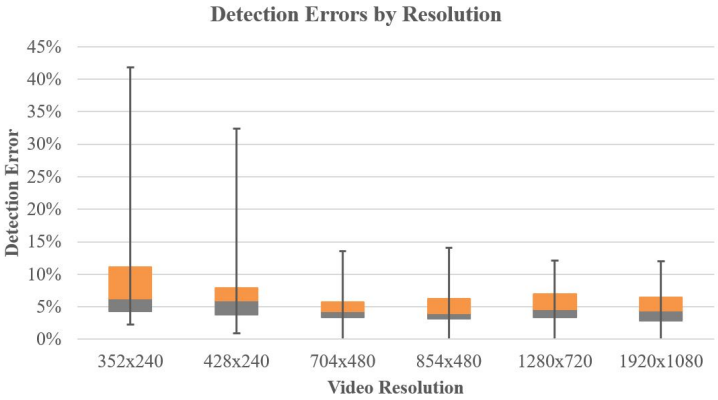

Figure 6: Boxplot of the detection errors grouped by resolution.

1920 x 1080, detection errors were equivalent between each other.

The results indicate that low resolutions may incur in greater detection errors. Since vehicles are represented by a small number of pixels, the feature-based tracking algorithm suffers to encounter relevant points to track. However, the opposite is not true. High resolutions, in our tests, did not outperform other lower resolutions.

In order to validate the null hypothesis that resolutions do not present a difference between each other, a Kolmogorov-Smirnov test was applied in the sample. Since 32 videos have been tested, our sample size is 32. For that sample size, the critical value to $\alpha = 5\%$ is $D_{critical} = 0.24$. If the critical value is lower than 0.24, it's not possible to reject the null hypothesis that the error distribution between the two videos are different. The results of the tests can be seen in Table 3

Table 3: Comparison for Kolmogorov-Smirnov results. A green check denotes equal distributions while a red check denotes different distributions.

Kolmogorov-Smirnov Test ($D_{critical} = 0.24$)						
	352x240	428x240	704x480	854x480	1280x720	1920x1080
352x240	-	✓	✗	✗	✗	✗
428x240	-	-	✗	✗	✗	✗
704x480	-	-	-	✓	✓	✓
854x480	-	-	-	-	✓	✓
1280x720	-	-	-	-	-	✓

For lower resolutions (352 x 240 and 428 x 240), the Kolmogorov-Smirnov test passed, so it's not possible to reject the hypothesis that the error distributions are different. But, when compared to higher resolutions, the test failed, indicating that there is a difference between higher resolutions and lower ones.

When comparing the average processing times for each resolution, the two higher resolutions tested, 1280 x 720 and 1920 x 1080, took more time to process all video files but did not had a proportional gain in vehicle detection. At our highest resolution, 1920 x 1080, the algorithm took, on average, 422 minutes to process 394 minutes of footage, exceeding the recording time in 7%, as can be seen in Figure 7.

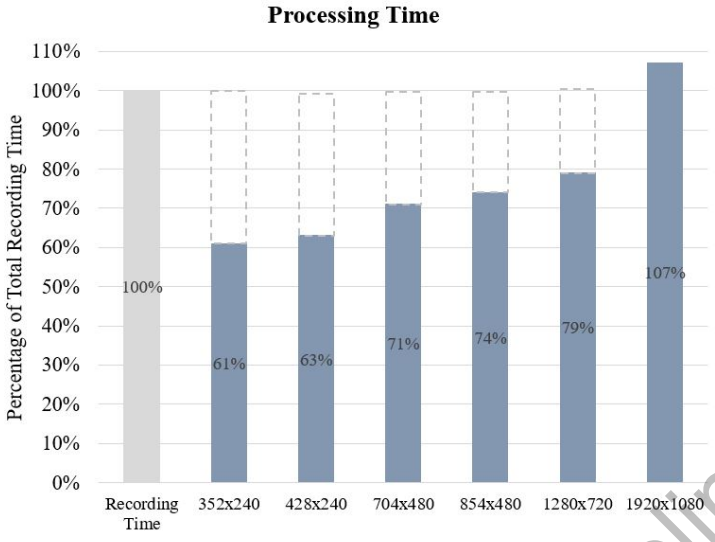

Figure 7: Total average processing time of different resolutions compared to the total time of evaluated footage.

In conclusion, our tests suggest that low resolutions, such as 352 x 240 or 428 x 240, are not recommended for a feature-based tracking algorithm, although its processing time is significantly lower than other resolutions. Not enough pixels are used to represent vehicles and, therefore, feature detection and tracking are impaired. High-definition resolutions do not suffer from this issue. However, greater amount of pixels do not represent a significant detection gain over intermediate resolutions, such as 704 x 480 or 854 x 480. Furthermore, HD resolutions incur in greater processing times. Intermediate resolutions, on the other hand, showed an equivalent detection rate to high resolutions, with a smaller processing time. Based on these results, it is recommended to use intermediate resolutions, such as 704 x 480 or equivalent in other aspect ratios, to decrease processing times while keeping the detection error rate at similar levels with higher resolutions.

8. CONCLUSION

The hypothesis proposed in this paper was that there is no impact on vehicle detection of a feature-based tracking algorithm using different video resolutions. This hypothesis was rejected for videos with smaller resolutions, but accepted for higher resolutions. We analyzed 32 videos, comprising of 394 minutes of footage, comparing the results of automatic vehicle counts to manual vehicle counts. As a result, our tests suggest that the use of low resolutions have a bad impact on feature-based tracking algorithms, incurring in greater detection errors. Since less pixels are used to draw vehicles, farther vehicles are grouped together in one single object. We also concluded that high resolutions, such as HD or full HD, do not represent a gain in vehicle detection and have a negative impact on processing times. Consequently, intermediate resolutions, such as 480p (704 x 480 or 854 x 480), are recommended for feature-based tracking algorithms and present the best results in our tests, both on detection rate and processing time.

Acknowledgements
The authors would like to thank São Carlos School of Engineering, for all the support. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

REFERENCES

Beymer, D., P. McLauchlan, B. Coifman, and J. Malik (1997). A real-time computer vision system for measuring traffic parameters. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pp. 495–501. IEEE.

CambridgeInColour (2019). *Understanding Digital Image Interpolation*. Available at `http://www.cambridgeincolour.com/tutorials/image-interpolation.htm`.

Cavallaro, A., O. Steiger, and T. Ebrahimi (2005). Tracking video objects in cluttered background. *IEEE transactions on circuits and systems for video technology 15*(4), 575–584.

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

Collins, R. T., Y. Liu, and M. Leordeanu (2005). Online selection of discriminative tracking features. *IEEE transactions on pattern analysis and machine intelligence 27*(10), 1631–1643.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine learning 20*(3), 273–297.

Do, V. D. and D.-M. Woo (2016). Multi-resolution estimation of optical flow for vehicle tracking. *Contemporary Engineering Sciences 9*, 843–851.

Jazayeri, A., H. Cai, J. Y. Zheng, and M. Tuceryan (2011). Vehicle detection and tracking in car video based on motion model. *IEEE Transactions on Intelligent Transportation Systems 12*(2), 583–595.

Lin, B.-F., Y.-M. Chan, L.-C. Fu, P.-Y. Hsiao, L.-A. Chuang, S.-S. Huang, and M.-F. Lo (2012). Integrating appearance and edge features for sedan vehicle detection in the blind-spot area. *IEEE Transactions on Intelligent Transportation Systems 13*(2), 737–747.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60*(2), 91–110.

Lucas, B. D., T. Kanade, et al. (1981). An iterative image registration technique with an application to stereo vision.

Marcomini, L. A. (2018). *Automatic Identification of Traffic Behavior Using Video Images*. Ph. D. thesis, Universidade de São Paulo.

OpenCV (2016). *Geometric Image Transformations*. Available at `https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html#resize`.

OpenCV (2019). *Geometric Image Transformations*. Available at `https://docs.opencv.org/2.4/`.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Sasson, S. (2007). *We Had No Idea*. Available at `https://web.archive.org/web/20130121220935/http://pluggedin.kodak.com/pluggedin/post/?id=687843`.

Saunier, N. and T. Sayed (2006). A feature-based tracking algorithm for vehicles in intersections. In *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, pp. 59–59. IEEE.

Shi, J. and C. Tomasi (1994). Good features to track. *Computer Vision and Pattern Recognition*, 593–600.

Shih, F. Y. and X. Zhong (2017). Automated counting and tracking of vehicles. *International Journal of Pattern Recognition and Artificial Intelligence 31*(12), 1750038.

Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Volume 2, pp. 28–31. IEEE.

Leandro Arab Marcomini (leandro.marcomini@usp.br)
André Luiz Cunha (alcunha@usp.br)
Department of Transport Engineering, São Carlos School of Engineering, University of São Paulo
1465 Dr. Carlos Botelho Avenue – São Carlos, SP, Brasil