Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 32, S. 59-67, Aralık 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 32, pp. 59-67, December 2021

Research Article

Performance Evaluation of Camera-Based Time to Collision Calculation with Different Detectors&Descriptors

Mehmet Özbek[1*], Aysun Taşyapı Çelebi[2]

[1*] Kocaeli University, Faculty of Engineering, Departmant of Electronics and Communication, Kocaeli, Turkey, (ORCID: 0000-0001-8036-4055), mehmettozbek@gmail.com

[2] Kocaeli University, Faculty of Engineering, Departmant of Electronics and Communication, Kocaeli, Turkey, (ORCID: 0000-0003-4047-1547), aysun.tasyapi@kocaeli.edu.tr

(International Conference on Design, Research and Development (RDCONF) 2021 – 15-18 December 2021)

(**DOI:** 10.31590/ejosat.1040524)

ATIF/REFERENCE: Özbek, M., & Çelebi, A.T. (2021). Performance Evaluation of Camera-Based Time to Collision Calculation with Different Detectors&Descriptors. *European Journal of Science and Technology*, (32), 59-67.

Abstract

Nowadays, the demand for producing and using autonomous vehicle is increasing. Due to the latest developments in technology, the capabilities of these vehicles in accident prevention are increasing. As a result of the accuracy of these capabilities, it is very important because it is human life. In today's technology, the collision time calculation called TTC (Time to Collision) can be done in two different ways. The first of these methods is lidar-based calculation. In this paper TTC will be calculated using the camera-based method with different combinations of detectors and descriptors. Pros and cons of these methods will be discussed. The aim of this paper is to expose an exacting performance for related methods, especially its diverse combinations are used matching. In these experiments images are used for 10 images taken from real time traffic scenario of preceding vehicle. This paper includes seven methods for detectors and 6 methods for descriptors. These detectors and descriptors are used in 42 different combinations. The analysis includes four parameters such as total keypoint detection, total matches, total time in ms and performance ratio which is total matches divided by total time.

Keywords: Autonomous Vehicles, Image Processing, Lidar, Detector, Descriptor.

Farklı Dedektörler ve Tanımlayıcılar ile Kamera Tabanlı Çarpışma Süresinin Hesaplanmasının Performans Değerlendirmesi

Öz

Günümüz otonom araç üretme ve kullanma talebi giderek artmaktadır. Teknolojideki gelişmeler nedeniyle bu araçların kaza önleme konusundaki yetenekleri de aynı oranda artmaktadır. Bu yeteneklerin doğruluğunun sonucu olarak insan hayatı söz konusu olduğunda oldukça önemlidir. Günümüz teknolojisinde TTC adı verilen çarpışma süresi hesabı iki farklı şekilde yapılabilmektedir. Bu yöntemden ilki lidar tabanlı hesaplamadır. Bu yazıda TTC, farklı dedektör ve tanımlayıcı kombinasyonları ile kamera tabanlı yöntem kullanılarak hesaplanacaktır. Bu bildirinin amacı, özellikle çeşitli kombinasyonların eşleştirilmesi için kullanılan yöntemler için hızlı. Bu deneylerde, öndeki aracın gerçek zamanlı trafik senaryosundan alınan 10 görüntü kullanılmıştır. Bu bildiri, dedektörler için yedi yöntem ve tanımlıyıcılar için 6 yöntem içermektedir. Bu dedektörler ve tanımlayıcılar 42 farklı kombinasyonda kullanılmaktadır. Analiz toplam anahtar nokta tespiti, toplam eşleşmeler, mili-saniye cinsinden toplam süre ve toplam eşleşmelerin toplam süreye bölünmesiyle elde edilen performans oranı gibi dört parametreyi içerir.

Anahtar Kelimeler: Otonom Araçlar, İmge İşleme, Lidar, Dedektör, Tanımlayıcı.

[*] Corresponding Author: mehmettozbek@gmail.com

1. Introduction

Currently, nearly 50 known companies are working on autonomous vehicles. Among these companies are the leading companies of the automotive world such as Tesla, BMW and Mercedes. In addition, spare parts supply companies that support these companies have directed their production on autonomous vehicles.

In addition to autonomous vehicles, changing lanes, reminding the speed signs, warning the driver against various road warning sign and providing awareness, placing cameras in various parts of the vehicle so that the driver can see the angles that the driver cannot see, increasing the driver's awareness of environmental effects with sensors such as radar and lidar, parking assist systems and blind spot detection systems are available to improve the driver's driving experience. Such systems are called Advanced Driver Assistance Systems (ADAS), and these systems are the precursors of autonomous vehicles as well as features that autonomous vehicles should acquire. In addition to the systems in the market with vehicles that offer autopilot, there are also vehicles that only offer ADAS.

In this paper, prototypes of autonomous vehicles, sensors and the concept of autonomous will also be discussed. For this reason, not all vehicles are fully autonomous. A standard has been established by a community known as the Society of Automotive Engineers (SAE International), called the autonomous level. According to this standard [1], a table was created according to the autonomy levels of autonomous vehicles. These levels of autonomy are shown in the Table 1 below.

Acquire useful datas from sensors for autonomous vehicles are important. Sensors and algorithms developed over the years are used to obtain these real-world data with the least possible loss. These algorithms work in the form of obtaining objects structurally. Image processing has a lot of example in real world. Image matching is a very important process to obtain meaningful information. However, due to the size of the data to be processed, this application is a very difficult situation due to the time in real-time applications. For this reason, in this study time is one of the benchmark parameters and one of the two parameters in performance measurement. To elicit and test these result an image database is taken from UDACITY Sensor Fusion Nanodegree Program. It includes 10 images that taken from real time traffic scenario. This paper consist of the following heading, previous studies about detector-descriptor algorithms are given in section two, in third section engineering logic is explained, fourth and fifth section explained time to collision logic with camera and lidar sensors. Sixth section includes the performance results and seventh section includes results will be discussed. Table 1 shows automation level of autonomous driving.

Table 1. Automation Level of Autonomous Driving

Driving Level and Name	Driving Information
0 – No Automation	There are systems on the vehicles to assist the driver, the drivers still perform the driving task.
1 – Driver Assistance	Level 1 is the lowest level of automation in vehicles. There are automatic systems such as acceleration or steering system. Adaptive cruise control is a good example of this level.
2 – Partial Automation	There must be a driver in the driver's seat and be able to interfere with driving in the event of a mishap.
3 – Conditional Automation	Vehicles at this level have environmental sensing capabilities. The driver must be on the alert.
4 – High Automation	Vehicles can intervene in case of unwanted traffic or system failure. There is not much need for human interaction.
5 – Full Automation	Vehicles of this level do not require human intervention. The driver's driving duty is eliminated.

The sensor set of an example autonomous vehicle is as follows:

- 360 degree Lidar scanner on top of vehicle
- 360 degree coverage radar
- Camera in the upper front of the vehicle
- Camera to the sides and back
- GPS antenna on vehicle roof
- Processing and storage unit

Cameras: Roof cameras can focus at far and near distances. It can monitor braking vehicles, pedestrians, traffic lights and traffic signs. The cameras transmit their image outputs to a central processing computer where the data of other sensors can be processed together. Just like the human eye, the night performance of cameras also decreases. This make cameras less reliable in terms of detection levels and locating accuracy.

Radars: Radars emit radio waves that have the ability to reflect back from objects. The returning waves can be analyzed with their return time and shifted frequency. Another feature is that radars are the only sensors that can measure the speed of objects directly, making the radar distinguishable from camera and lidar in this regard. In addition, the radar is very resistant to adverse weather conditions such as snowfall and fog. Radar, which has been used for many years, gives the best results when identifying large objects with good reflectivity. The performance of the radar is degraded when identifying objects with low reflectivity. Even if the camera and radar work well together, there are cases where both sensors do not work optimally. For these reasons, autonomous vehicle manufacturers add a third sensor in addition to these two sensors.

Lidar: Lidar works similarly to radar. However, unlike radar, it uses infrared light instead of radio waves. The ceiling mounted lidar sensor rotates at high speed. Creates a detailed laser beams 3D model of its surroundings. 128-layer sensors a total of 128 laser beams are used to detect distances up to 300 meters. During a 360-degree rotation, approximately 4 million dots occur per second. Like a camera, a lidar is an optical sensor. Cameras are dependent on ambient light while lidar does not have these dependencies. However, the performance of lidar decreases in adverse weather conditions such as heavy snowfall and fog. In such an environment, the spots formed by the lidar may not be sufficient for detection. For this, it should be supported with sensors besides lidar.

2. Material and Method

2.1. Proposed Studies

In literature, there are too many works related to keypoint detectors and descriptors combinations for comparing their performances. Which combination is more successful is related to which performance measurement parameter the results of these processes are related to. Distinctive features in the images are defined by concepts called detector and descriptor. One of these concepts, the detector finds the important points in the image. The descriptor is a definition that can be matched with each other in common features among different images and is obtained by calculating these features. Image recognition methods are described in the referenced studies below. In the one of the proposed study numbered [2-5], the results revealed the results of BRISK, FREAK, SURF, SIFT descriptors. The best performance seeks the best match between the targeted detection accuracy, speed and the objects it detects in the targeted study. This method is too sensitive to deterioration and robustness cannot be fully ensured. Addition, the linear forward and backward movements of the camera system of the study are too limited. In [6], it is a study investigating detector and descriptor methods for studies on photogrammetrics. It compares five keypoint detector in terms of correctly detected corners, their positions, the density of detected points. But the five methods are few for comparison and the performance analysis does not give a very accurate result. Reference study [7] aims to find the best combination in parallel with our study. For this purpose, it analyzes different combinations with 7 detector and 2 descriptors. A dataset contains 60 images was used. In previous references, in this reference also and in our study, different criteria were used to evaluate performance results. Finally, another study [8] is a study on occlusions and was carried out using a moving camera. Four descriptors were used in the study. These are the SIFT, FREAK, SURF and BRISK descriptors.

The difference of our study from these proposed papers is the difference in the number of detector and and descriptors used and their combinations. The performance measurement parameter used later is a less complex formula. However, this formula is quite sufficient as the area of use is rear image of the preceding vehicle. The above-mentioned studies have inspired our work on different subjects.

2.1. Engineering a Collision Detection System

Collision avoidance system (CAS) is a safety feature that alert drivers and triggers the brake in case of a sudden collision while driving. If there is a vehicle ahead (preceding vehicle), CAS estimates the sustained collision time [9-10] (TTC). When the TTC drops below a predetermined threshold, the CAS may decide to apply the vehicle brakes autonomously. Fig.1 shows the real time traffic scenario

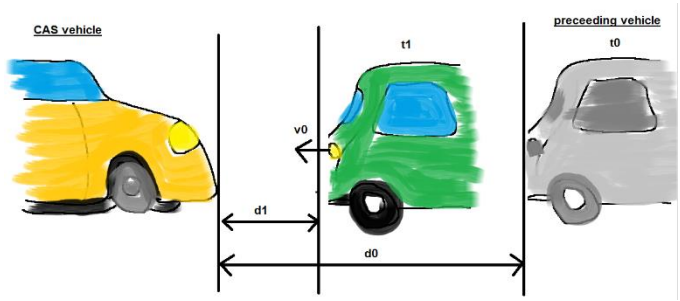

Figure 1 – Real time traffic scenario.

In the traffic scenario shown in the figure above, the green vehicle starts to decrease its speed at the moment *t0* when the yellow vehicle equipped with the collision sensor receives the distance measurement d_0. After a time t (t_1) the green vehicle comes very close and a second measurement of d1 is made. The purpose here is to calculate the TTC. Thus, the TTC can be calculated and the driver of the yellow vehicle can be warned. Even the brakes can be triggered autonomously. However, before this process can be done, a way to describe the movement of vehicles with a mathematical model must be found. The parameters used in the equations below are the relative speeds of the vehicle speeds, the vehicle carrying the sensor, and the vehicle scanned by the sensor. To calculate TTC, the physical behavior of the preceding vehicle must be modeled. One assumption in this regard may be that the relative velocity between the yellow and green vehicle in the above figure is constant. This will result in the constant velocity model (CVM) [11] represented by Equation (1) in the formula below. V represents the velocity and d the distance and should not be confused with the derivative operator.

Constant Velocity

$$d(t + \Delta t) = d(t) - v_0.\Delta t \quad (1)$$

Constant Acceleration (2)

$$d(t + \Delta t) = d(t) - v_0.\Delta t - \frac{1}{2}a_0.\Delta t^2$$

$$v(t + \Delta t) = v(t) - a_0.\Delta t$$

The distance to the vehicle at time instant t + Δt is smaller than at time t because it subtract the product of a constant relative velocity v_0 and time Δt. From an engineering perspective, a sensor is needed because of the capable of measuring the distance to the preceding vehicle on a precise times basis with a constant Δt between measurements. This one achievable quite well with a lidar sensor. Especially in dynamic traffic situations where a vehicle is braking hard, the CVM is not accurate enough, however, as the relative velocity between both vehicles changes between measurements. In the following figure, the approaching vehicle is shown at three-time instants with increasing velocity. Fig. 2 shows the preceding vehicle increasing velocity.

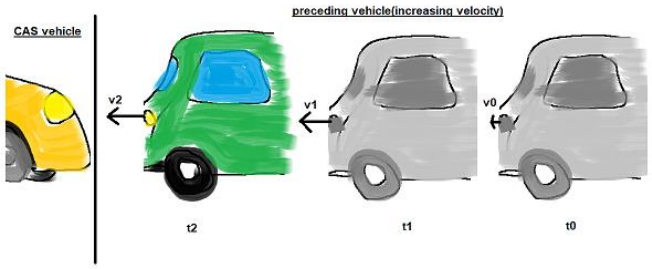

Figure 2 – Preceding vehicle increasing velocity.

Thus CVM can be expand by assuming velocity to be a function of time and subtract the second term in Equation (2) which is the product constant acceleration and the squared time *Δt* between both measurements. Equation (2) displays velocity as a function of time, which is also dependent on the constant acceleration model (CAM) and it is commonly used in commercially available collision detection systems. On a side note, if a radar sensor used instead of a lidar, a direct measurement on velocity could be taken by exploiting a frequency shift in the returning electromagnetic wave due to the Doppler effect [12]. This is a significant advantage over sensors such as Lidar, where velocity can only be computed based on (noisy) distance measurements. In this paper CVM will be used instead of the CAM as it is much simpler to handle with regard to the math involved and with regard to the complexity of the programming task. For small instances of Δt will assumed that the CVM model is accurate enough and that it will give a decent estimate of the TTC. As a conclusion, there are the following types of models possible.

1. Constant Velocity Model (CVM): In this paper, that will consider being working on.
2. Constant Acceleration Model (CAM): An ideal case, but still complex as compared to the CVM model.
3. Changing Acceleration: Real-life scenarios, most often too complex to handle in practice

2.3. Estimating TTC with Lidar

In the following assuming that CAS equipped vehicle using a LIDAR sensor to take distance measurements on preceding vehicles. The sensor in this scenario will be given the distance to the closest 3D point in the path of driving. In the figure below, the closest point is indicated by a red line emanating from a lidar sensor on top of the CAS vehicle. Fig. 3 shows the math behind TTC.

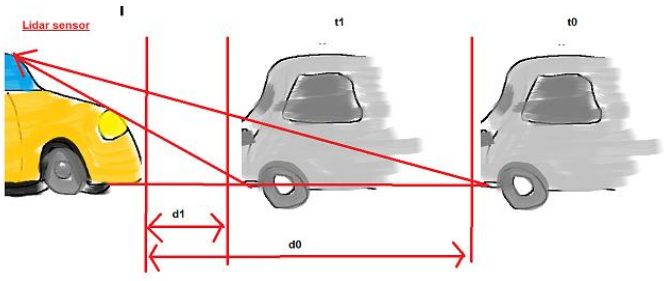

Figure 3 – The math behind TTC.

Based on the model of a constant velocity it is discussed, the velocity v_0 can be computed from two successive lidar measurements as follows:

$$d(t + \Delta t) = d(t) - v_0 . \Delta t \quad (3)$$

$$v_0 = \frac{d(t) - d(t + \Delta t)}{\Delta t} = \frac{d_0 - d_1}{\Delta t} \quad (4)$$

$$TTC = \frac{d_1}{v_0} = \frac{d_1 . \Delta t}{d_0 - d_1} \quad (5)$$

Once the relative velocity v_0 is known, the time to collision can easily be computed by dividing the remaining distance between both vehicles by v0. So given a lidar sensor which is able to take precise distance measurements, a system for TTC estimation [13-14] can be developed based on a CVM and on the set of equations shown above. Note however that a radar sensor would be the superior solution for TTC computation as it can directly measure the relative speed, whereas with the lidar sensor needs to be computed v0 from two (noisy) distance measurement. The following image shows a lidar point cloud as an overlay a camera image taken in a highway scenario with a preceding vehicle directly in the path of driving. Distance to the sensor is color-coded (green is far away, red is close). On the left side, a bird-eyed view perspective of the lidar points is shown as well. Fig. 4 shows the highway scenario with a preceding vehicle.

Figure 4 – Highway scenario with a preceding vehicle.

The lidar sensor provides measurements on the vehicle as well as on the road surface. Also, some 3D points in the camera image do not seem accurate when compared to their surrounding neighbors. Especially the points near the roof of the preceding vehicle differ in color from the points on the tailgate. As measurement accuracy is correlated to the amount of light reflected from an object, it makes sense to consider the reflectiveness r of each lidar point which can be accessed. In addition to the x, y and z coordinates. The image below highlights high reflectiveness with green, whereas regions with low reflectiveness are shown as red. An analysis of the associated reflectivity of the point cloud shows that such deviations often occur in regions with reduced reflectiveness. Fig. 5 shows the reflectiveness with preceding vehicle.

Figure 5 – Reflectiveness with preceding vehicle.

In order to derive a stable TTC measurement from the given point cloud, two main steps have to be performed :

1- Remove measurements on the road surface
2- Remove measurements with low reflectivity

In the figure below, Lidar points are shown in a top-view perspective and as an image overlay after applying the filtering. After removing lidar points in this manner, it is now much easier to derive the distance *d(t)* to the preceding vehicle. Fig. 6 shows the Lidar points perspective.

Figure 6 – Lidar points perspective.

2.3. Estimating TTC with Camera

Monocular cameras are not able to measure metric distances. They are passive sensors that rely on the ambient light which reflects off of objects into the camera lens. It is thus not possible to measure the runtime of light as with lidar technology. To measure distance, a second camera would be needed. Given two images taken by two carefully aligned cameras (also called a stereo setup) at the same time instant, one would have to locate common points of interest in both images (e.g. the tail lights of the preceding vehicle) and then triangulate their distance using camera geometry and perspective projection. For many years, automotive researchers have developed stereo cameras for the use in ADAS products and some of those have made it to market. With more advanced ADAS products and with autonomous vehicles however, stereo cameras have started to disappear from the market due to their package size, the high price and the high computational load for finding corresponding features. Despite those limitations of the mono camera, there is a way to compute TTC without the need to measure distance. Consider the constant velocity motion model that introduced and think about a way to replace the metric distance d with something the camera can measure reliably, such as pixel distances directly on the image plane. In the following figure, it can be seen how the height H of the preceding vehicle can be mapped onto the image plane using perspective projection.

It can be seen that the same height H maps to different heights h_0 and h_1 in the image plane, depending on the distance d_0 and d_1 of the vehicle. It is obvious that there is a geometric relation between *h*, *H*, *d* and focal length *f* of the pinhole camera and this is what needed to exploit in the following. Fig. 7 shows the height of the preceding vehicle effect distance.

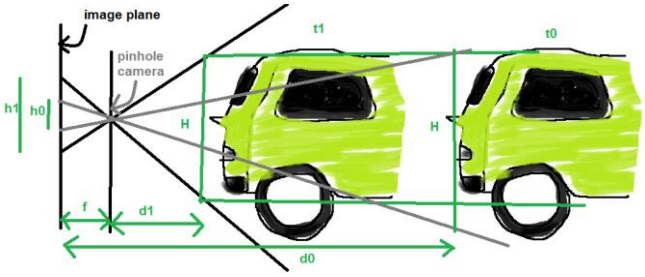

Figure 7 – Height of the preceding vehicle effect distance.

Looking at the following set of equations:

Project object into camera (6)

$$h_0 = \frac{f.H}{d_0};\ h_1 = \frac{f.H}{d_1}$$

Relate projection and distance (7)

$$\frac{h_1}{h_0} = \frac{\frac{f.H}{d_1}}{\frac{f.H}{d_0}} = \frac{d_0}{d_1} \rightarrow\ d_0 = d_1.\frac{h_1}{h_0}$$

Substitute in constant velocity model (8)

$$d_1 = d_0 - v_0.\Delta t = d_1.\frac{h_1}{h_0} - v_0.\Delta t \rightarrow\ d_1 = \frac{-v_0.\Delta t}{\left(1 - \frac{h_1}{h_0}\right)}$$

Compute time to contact / collision (9)

$$TTC = \frac{d_1}{v_0} = \frac{-\Delta t}{\left(1 - \frac{h_1}{h_0}\right)}$$

In Equation (6) the focal length of the camera used and a distance measurement d_0 performed at time t_0 to project the height *H* of the vehicle onto the image plane and thus to a height h_0 in pixels. The same is done at time t_1, leading to a projected height h1. In Equation (7) the ratio of the relative heights h_0 and h_1 are computed. As both *H* and *f* are canceled out, a direct relation can be observed between relative height *h* and absolute metric distance *d*. The distance to the vehicle d_0 can be expressed as the product of d_1 and the ratio of relative heights on the image plane. In Equation (9), d_0 in the equation for constant velocity substituted and solve for d1 which is now dependent on the constant relative velocity v1, on the time between measuring d_0 and d_1 and on the ratio of relative heights on the image plane. Also in Equation (9) the TTC is computed as the ratio of remaining distance to impact, which is d_1 and the constant velocity v_0. As it can easily seen, the TTC now only consists of Δt, h_0 and h_1. Thus it is possible to measure the time to collision by observing relative height change on the image sensor. Distance measurements are not neeeded and it can thus use a mono camera to estimate the time to collision bu observing changes in relative height (also called chang) directly in the image. In the figure below, a neural network has been used to locate vehicles in successive image of a monocular camera.

For each vehicle, the network returns a bounding box, whose width and/or height could in principle be used to compute the height ratio in the TTC equation derived in the last section. When observed closely however, it can be seen that the bounding boxes do not always reflect the true vehicle dimensions and the aspect ratio differs between images. Using bounding box height or width for TTC computation would thus lead to significant errors.

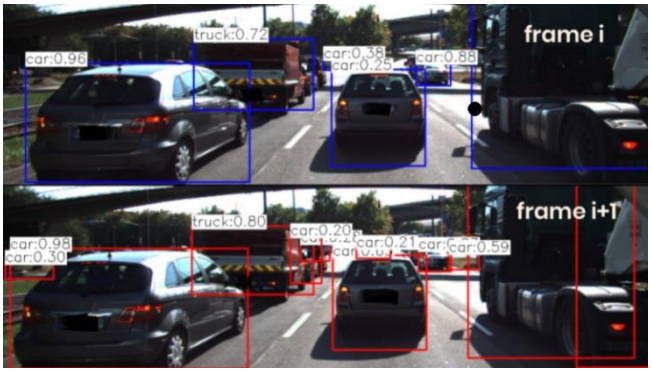

Figure 8 – Locate vehicle in successive images of a monocular camera.

In most engineering tasks, relying on a single measurement or property is not reliable enough. This holds especially true for safety related products. Therefore, it needs to be consider whether there are further properties of vehicles and objects it can be observed in an image. Instead of relying on the detection of the vehicle as a whole now needs to be analyze its structure on a smaller scale. If it were possible to locate uniquely identifiable keypoints that could be tracked from one frame to the next, it could use the distance between all keypoints on the vehicle relative to each other to compute a robust estimate of the height ratio in our TTC equation. The following figure illustrates the concept. Fig. 9 shows the keypoints of a car perspective.

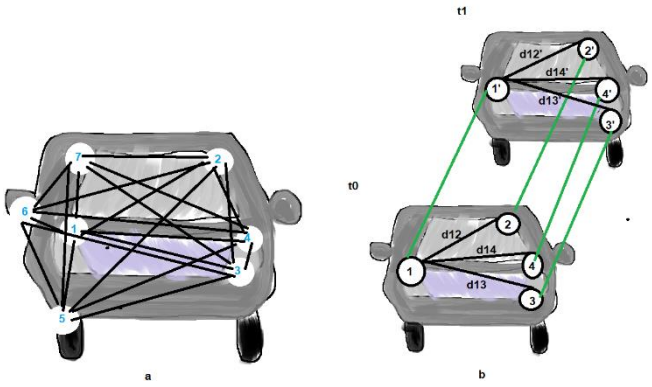

Figure 9 – Keypoints of a car perspective a) relative distances between keypoints in an image b) selected keypoint distances between successive frames.

In (a), a set of keypoints has been detected and the relative distances between keypoints 1-7 have been computed. In (b), four keypoints have been matched between successive images (with keypoint three being a mismatch) using a higher-dimensional similarity measure called descriptor. The ratio of all relative distances between each other can be used to compute a reliable TTC estimate by replacing the height ratio h1/h0 with the mean or median of all distance ratios dk / d'k. Studies on keypoint detectors have increased recently and many algorithms have been developed in recent years for these reasons. Keypoint detection applications include object recognition in image processing, robotic mapping, 3D modeling etc. These detectors are comparable in terms of performance and speed. In recent years, a number of faster detector have been developed which aim at real-time applications on smartphones and other portable devices. Fig. 10 shows the relative distance between keypoints.

Figure 10 – Relative distance between keypoints.

In the literature, a large variety of similarity measures (called descriptors) have been proposed and in many cases, authors have published both a new method for keypoint detection as well as a similarity measure which has been optimized for their type of keypoints. A keypoint detector is an algorithm that selects points from an image based on the local maximum of a function. A descriptor is a vector describing the image patch around a keypoint. It has many describing the image patch around a keypoint. It has many techniques and these include from simple to complex techniques such as comparing raw pixel values, histograms of gradient orientations. Descriptors help to assign similar keypoints in different images to each other. As shown in the figure below, a set of keypoints in one frame is assigned keypoints in another frame such that the similarity of their respective descriptors is maximized and the keypoints represent the same object in the image. In addition to maximizing similarity, a good descriptor should also be able to minimize the number of mismatches, i.e. avoid assigning keypoints to each other that do not correspond to the same object.

Most common methods are ORB, BRISK, SURF, SIFT, SHITOMASI, HARRIS. These methods generally using for this type of study. This study may refer to these steps; keypoint descriptor as given in [3], orientation assignment, keypoint localization and scale-space representation. To put it another way, the last three of the above steps are the detector, while the first is expressed as the descriptor. However, among these methods, there are those that can be used both as detectors and descriptors. Some are just detectors or only descriptor ones. In [15-16], FAST is a detector method. In [17], the BRIEF method is a descriptive method. If we talk about SIFT method, the SIFT method applies a set of DoG filters for multiscale. With this filter, we can obtain a filtered and downsampled version of the original image. The way the SIFT descriptor is created is from a histogram with a gradient size of 4x4. As another example, the basis for the formation of SURF is the sum of two-dimensional Haar wavelets using integral images and approximating the Gaussian derivatives in [3]. The SURF detector approximates the determinant of the Hessian matrix, which gives a local maximum as a result. Unlike the detector, the SURF descriptor consists of a 64 dimensional vector, which is calculated as a result of the sum of the Haar wavelet coefficients over a 4x4 pixel. As described in [15] and [16] the application area of FAST

is to detect the corners. It uses a 16 pixel circle around the corner pixels to understand that the point of interest is the corner. It then classifies these 16 pixels by comparing their brightness, and together with a threshold, it is understood whether the relevant pixel is a corner or not. As for BRIEF, BRIEF is a binary descriptor and based on density comparison. The detector side of BRISK, as summarized in reference [4], calculates the pixel maximum. This maximum is also called FAST score calculation. The BRISK descriptor is combined with a binary array and includes a gloss results test. Fig. 11 shows the keypoint-descriptor relation.

Figure 11 – Keypoint-descriptor relation.

3. Results and Discussion

The experiments to be tested in this paper are the success of the methods used to calculate the keypoints of the approaching vehicle, thus its distance and TTC time. For this, the 10 images below contain the preceding of a vehicle that slows down during traffic to the vehicle in use. The distance between this vehicle and the vehicle used in gradually decreasing, and therefore the TTC is getting closer and closer.

Table 2 contains different detectors and descriptors used in 10 images. It contains all the possible combination of detector and descriptor pairs. Then the total number of keypoints found by these combinations is indicated, the number of keypoints that match with the detected keypoints was found thanks to the descriptors. Since the number of match keypoints are not the only parameter, how long it takes for the relevant combination to find this match number is in another column. In the last column, the relevant value was obtained as a result of the ratio of the number of match keypoints, which is considered as a performance parameter called ratio can be found. Table 3 contains the 3 methods with the highest ratior were selected from the results in the Table 2. The distance calculation result with the camera was the recommended for obtaining TTC.

Figure 12 – Preceding vehicle test images.

Table 2. Benchmark of Detector + Descriptor Combinations with the Related Parameters

No	Combinations	KP's	Match	Time(ms)	Ratio	No	Comb.	KP's	Match	Time(ms)	Ratio
1	SHI+BRISK	11875	1994	124,54	16,01	22	BRISK +FREAK	26144	4828	2117,78	2,28
2	SHI+BRIEF	11875	2861	83,66	34,20	23	BRISK +AKAZE	-	-	-	-
3	SHI+ORB	11875	2526	100,74	25,07	24	BRISK +SIFT	26144	6685	2150,10	3,11
4	SHI+FREAK	11875	2299	318,98	7,21	25	ORB +BRISK	4895	1349	144,82	9,32
5	SHI+AKAZE	-	-	-	-	26	ORB+ BRIEF	4895	1373	120,78	11,37
6	SHI+SIFT	11875	2842	143,58	19,79	27	ORB +ORB	4895	1435	151,04	9,48
7	HARRIS+BRISK	756	227	87,86	2,58	28	ORB +FREAK	4895	613	353,70	1,73
8	HARRIS+BRIEF	756	266	91,27	2,91	29	ORB +AKAZE	4895	-	-	-
9	HARRIS+ORB	756	261	84,90	3,07	30	ORB +SIFT	4895	1544	334,48	4,62
10	HARRIS+FREAK	756	217	303,40	0,72	31	AKAZE +BRISK	13330	3216	395,30	8,14
11	HARRIS+AKAZE	-				32	AKAZE +BRIEF	13330	4011	370,55	10,82
12	HARRIS+SIFT	756	265	138,35	1,92	33	AKAZE +ORB	13330	3315	376,98	8,79
13	FAST+BRISK	17330	3073	65,95	46,59	34	AKAZE +FREAK	13330	3204	575,26	5,57
14	FAST+BRIEF	17330	4754	28,25	100,27	35	AKAZE +AKAZE	13330	3437	645,25	5,33
15	FAST+ORB	17330	4124	20,85	197,76	36	AKAZE +SIFT	13330	3606	477,36	7,55
16	FAST+FREAK	17330	3067	273,28	11,22	37	SIFT +BRISK	13540	2401	433,88	5,53
17	FAST+AKAZE	-	-	-	-	38	SIFT +BRIEF	13540	3168	427,95	7,40
18	FAST+SIFT	17330	5559	145,08	38,32	39	SIFT +ORB	-	-	-	-
19	BRISK+BRISK	26144	4891	1927,9	2,54	40	SIFT +FREAK	13540	2371	667,16	3,55
20	BRISK+BRIEF	26144	7206	1868,13	3,86	41	SIFT +AKAZE	-	-	-	-
21	BRISK+ORB	26144	4912	1880,68	2,61	42	SIFT +SIFT	13540	2497	749,04	3,33

Table 3. Top three Detector + Descriptor Pair

No	Combinations	KP's	Matches	Time	Ratio
1	FAST+ ORB	17330	4124	20.8537	197.76
2	FAST+ BRIEF	17330	4754	28.2515	168.27
3	FAST+ BRISK	17330	3073	65.9564	46.59

When the given images are used as input data, Table 2 shows the keypoints found, the points that match the points and how long it takes for this match to be made. When these matches and the duration are divided, the ratio called ratior emerges. When these ratio data are examined, the most performing (matches / time) combinations are given in Table 3. When Table 3 is examined, the FAST method appears to be most performance detector. In addition to fast as descriptor, ORB, BRIEF and BRISK methods are seen as the most performance combinations when used.

4. Conclusions and Recommendations

In this study, while doing some software load in autonomous vehicles, in our study, this task was to calculate the imapct time of the vehicle in front. While calculating the TTC, we used the detector and descriptor pairs to detect the distinctive features of

the preceding vehicle. As can be seen from the results, the most efficient detector and descriptor pairs were obtained.

5. Acknowledge

We thank Aaron Brown (Software Engineer at Mercedes-Benz Research & Development North America) for providing us his knowledge to electrical engineering, robotics and deep learning behind the autonomous vehicles. We thank Dr. Andreas Haja (Professor at University of Applied Sciences Emden) for his ADAS and computer vision knowledge. We thank Gürol Çokünlü (Digital Transformation Manager at Otokar). This work was supported by the Otokar Automotive and Defence Industry.

References

Min, K. W., Han, S. J., Lee, D. J., Choi, D. S., Sung, K. B., & Choi, J. D. (2019). SAE level 3 autonomous driving technology of the ETRI. *2019 International Conference on Information and Communication Technology Convergence (ICTC)*.

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

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, *110*(3), 346–359.

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. *2011 International Conference on Computer Vision*.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to SIFT or surf. *2011 International Conference on Computer Vision*.

Remondino, F. (2021). Detectors and descriptors for photogrammetric applications. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Dahl, A. L., Aanæs, H., & Pedersen, K. S. (2011). Finding the best feature detector-descriptor combination. *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*.

Li, Q., Mao, Y., Wang, Z., & Xiang, W. (2009). Robust real-time detection of abandoned and removed objects. *2009 Fifth International Conference on Image and Graphics*.

Seki, Y., Ohya, J., & Miyoshi, M. (1999). Collision avoidance system for vehicles applying model predictive control theory. *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*.

Mukhtar, A., Xia, L., & Tang, T. B. (2015). Vehicle detection techniques for collision avoidance systems: A Review. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2318–2338.

Matsuzaki, T., Kameda, H., Tsujimichi, S., & Kosuo, Y. (1999). Manoeuvring target tracking using constant velocity and constant angular velocity model. *SICE '99. Proceedings of the 38th SICE Annual Conference. International Session Papers (IEEE Cat. No.99TH8456)*.

Chen, W., Zhou, G., & Giannakis, G. B. (1995). Velocity and acceleration estimation of Doppler Weather Radar/LIDAR signals in Colored Noise. *1995 International Conference on Acoustics, Speech, and Signal Processing*.

Bosnak, M., & Skrjanc, I. (2017). Efficient time-to-collision estimation for a braking supervision system with Lidar. *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*.

Shen, Q. (2021). Design of backward collision warning and avoidance system when on-street parking using LIDAR. *2021 2nd International Conference on Computing and Data Science (CDS)*.

Rosten, E., & Drummond, T. (2005). Fusing points and lines for high performance tracking. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*.

[Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. *Computer Vision – ECCV 2006*, 430–443.

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. *Computer Vision – ECCV 2010*, 778–792.