

PAPER • OPEN ACCESS

Vehicle Collision Detection Application Through Collision Video Files with Quadtree Algorithms

To cite this article: Jimmy Tandean *et al* 2019 *J. Phys.: Conf. Ser.* **1230** 012016

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the **collection** - download the first chapter of every title for free.

Vehicle Collision Detection Application Through Collision Video Files with Quadtree Algorithms

Jimmy Tandean, Endry, Alvin Wijaya, Winata Gunawan, Mawaddah Harahap

Faculty of Technology and Computer Science, Universitas Prima Indonesia

* mawaddah@unprimdn.ac.id

Abstract. The improvement of digital image processing technology is increasingly rapid. Image processing has been widely used to maximize the usefulness of the webcam interface or CCTV, which can be used to monitor traffic flow on the street. One of the frequent event occurs on the street is a crash (collision). This problem often arises in a collision event is a dispute about the party guilty in the collision. A collision detection application using Quadtree algorithm can solve this problem. This study aims to analyze and design intelligent systems for vehicle collision detection on street through a webcam camera. Using the Quadtree algorithm, a collisions that occur on vehicles can be detected by the regions division method such as Quadtree algorithm. Using this application the collision possibilities can be describe. Based on the results of the tests conducted, information was obtained that the accuracy of the application of the Quadtree algorithm in the detection of car collisions on the highway was 8: 2 where the value was obtained from the number of collisions detected.

1. Introduction

The improvement of digital image processing technology is increasingly rapid. In this modern era, a lot of emerging software that can simplify human life and work. The existing scientific disciplines including image processing has been widely used to maximize the usefulness of webcam or CCTV interface. With increasingly sophisticated technology, this discipline is increasingly being developed, so that one of them can be used to monitor traffic flow on the highway[1]. One of the frequent occurrences on the highway is a collision. Vehicle collisions are often occur today. Reporting from Republika.co.id, traffic accidents nationwide continue to increase every year. Since 2014 until the last year, the number is increasing. Throughout 2014 there were 95,906 cases, the following year in 2015 amounted to 98,970 cases, and the last 2016 increased to 105,374 cases with death victims recorded 25,859 people, seriously injured 22,939 people, minor injuries 120,913 people. However, compared to 2012, it decreased from 117,949 cases to 100,106 cases in 2013. To detect collisions, a collision detection application was needed.

Some studies related to vehicle monitoring, such as research [2-4] which discusses the introduction of cars in digital imagery using HOG-SVM. Other research is [5] which discusses the Design of Vehicular Based Car Accident Monitoring and Control System Ad Hoc Network (Vanet) Using Limit Switch Sensors and Rotary Encoder, where the purpose of research is the creation of a car robot prototype that can detect accidents and send information accident to prevent consecutive accidents. Research [6] discusses the design of Smart Vehicle to Detect Accident Early with Visual Reporting on Google Maps, where the research uses an accident detection device in the form of an Accelerometer sensor by detecting vibrations that occur on the X, Y, and Z axes and taking coordinates based on GPS data on when an accident occurs, the tool then reports quickly in the form of SMS using a GSM module. Several previous studies have not discussed the detection of collisions using CCTV cameras installed on each highway. Previous research requires the use of new tools or hardware that require considerable additional costs in practical application. To solve these problems, this study will use CCTV cameras to detect collisions on highways. The algorithm used in the collision detection process is the Quadtree algorithm.



Several studies that apply the Quadtree algorithm, such as research [7] which discusses the application of the Quadtree Algorithm for Digital Image Segmentation of Traffic Signs, which based on the results of the tests it seems quite clear that the Quadtree algorithm is capable of segmenting objects in the traffic sign input image. Quadtree algorithm segmentation in the segmentation model on the input image of traffic signs is quite able to find the basic components of the object and be able to find the basic color information used. Research [8-9] discusses the application of the Quadtree Algorithm for Collision Detection in Adventure of Upik Games and the application of the Collision Detection and Boidspada Algorithm for Collision Detection in Game Dokkaebi Shooter.

Using the Quadtree Algorithm, we can detect collisions that occur between cars or between car and motorcycle by the regions division method. Thus, we can describe the possibilities that will occur in a collision. There are several steps needed in processing are grayscale, binary image, segmentation, detection, tracking, recognition, and calculation. In the process of working, segmentation operations are carried out by passing images on high-pass filters. High-pass filters will pass or strengthen high-frequency components (such as edges or edges of objects) and will lower low-frequency components. As a result, images, high-pass filters are also used to detect the presence of edges (edge detection). In this case, edge pixels are displayed brighter while pixels instead of edges are darkened [10].

Based on the research, we intends to use the Quadtree algorithm to facilitate collision detection on highways by building applications that are capable of detecting collisions between vehicle objects using the Quadtree algorithm. Based on the background of the problem above we is interested in the title "**Vehicle Collision Detection Application Through Collision Video Files with Quadtree Algorithms**".

2. Methodology

2.1. The Quadtree Algorithm

One example of the m-ary tree is Quadtree. Quadtree is a m-ary tree, each of which has 4 branches. Branches on Quadtree can be vertices that have more branches as many as 4 branches Quadtree branches can also be directly in the form of leaves or a combination of branches and leaves. The leaves on the Quadtree usually contain information about something [11]. The description of the tree is in Figure 1 as follows.

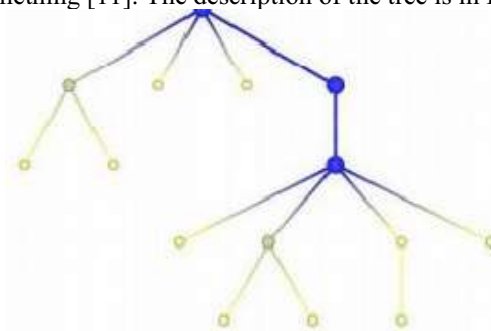


Figure 1. Quadtree Algorithm Model [12]

Many examples of quadtree and its application can be found. Quadtree is often applied in computer game to detect collision of two different object efficiently (for checking collisions in the arena of three-dimensional games used Oct-Tree, a tree with a maximum number of 8 branches on each node). Transportation vehicles can be driven by machinery or by driver. These vehicles are usually man-made (cars, motorbikes, trains, boats and planes), but some are not man-made and can still be called vehicles, such as icebergs and floating tree trunks. Non-motorized vehicles can also be driven by humans or pulled by animals, such as carts. In other terms, transportation is defined as an effort to move or move from one location to another by using a particular tool. Transportation has dimensions such as location (origin and destination), tools (technology) and certain needs. Data from the transportation ministry found in Figure 1 illustrates traffic accidents that occurred in 2008-2012 [13].

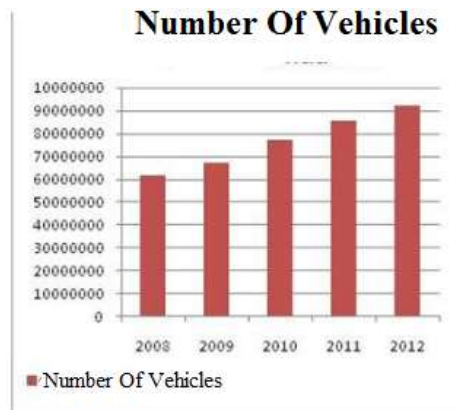


Figure 2. Increasement of Total Amount of Car Every Year [13]

Traffic accidents are a social problem and have become commonplace for the people of Indonesia because the intensity of the event is increasing. According to statistics from the World Health Organization (WHO), traffic accidents are one of the 10 highest cases of death in the world [13]. In particular, there were approximately 3500 traffic accidents every day in 2014.

The graph in Figure 2 shows that from 2008 to 2012 the number of motorized vehicles increased. With the increase in the number of motorized vehicles each year, the volume of highways is becoming more congested. It can be concluded that by proportionally the number of vehicles and the number of accidents with time, there is a relationship between the number of vehicles and the number of accidents. Studies show that the majority of the causes of traffic accidents are human error factors, for example driving a car with dangerous and abnormal movements [14].

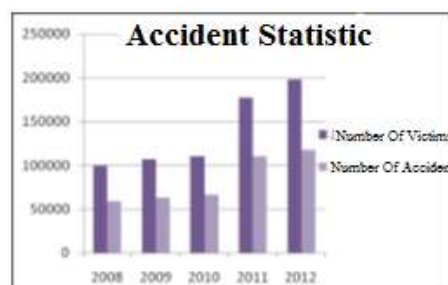


Figure 3. Yearly Traffic Accident Statistics

Steps to detect a collision with the Quadtree algorithm is to divide the area of the digital image into four equal-sized parts. If two or more objects are in the same area, divide the area into four equal-sized parts. Regional division continues until each object is not in the same area. The division of this region is done recursively. The following is an example of the distribution in Figure 2 below.

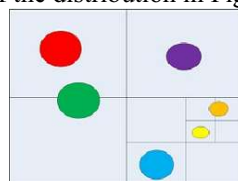


Figure 4. Quadree Objects Area Division

After mapping the objects into Quadtree, the objects which might collide will be detected. The following is an illustration of the detection of collision objects in Figure 3 below.

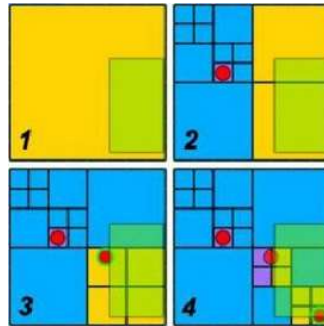


Figure 5. Object Areas Division

2.2. Process Analysis

In general, the Quadtree Algorithm can be explained, namely the first input and then the input image, then scanning the pixels, then Grayscale, then the initialization, quadtree, then the output then finished. The object detection process that will be designed implements the Quadtree algorithm which requires several preprocessing steps to meet the criteria for the object recognition process to be performed.

1. Grayscale Process

Grayscale image is a digital image that has only one channel value in each pixel, meaning that the value of Red = Green = Blue. These values are used to indicate the intensity of the color. Images displayed from this type of image consist of gray, varying in black on the part of the weakest intensity and white at the strongest intensity. Grayscale images are stored in an 8-bit format for each pixel sample, which allows 256 intensities. To change the color image that has a matrix value of each R, G and B to be a grayscale image with a GS value, then the conversion can be done by taking the average values of R, G and B so that they can be written as:

$$GS = \frac{(R + G + B)}{3}$$

Suppose the pixels to be converted have a color value R: 255, G: 126 and B: 168 so that the grayscale value is obtained:

$$GS = \frac{(255 + 126 + 168)}{3}$$

$$GS = \frac{549}{3}$$

$$GS = 183$$

After the grayscale process is complete, the process will continue with the digital image binary process. The binary process will start from the grayscale image input process and continue with the process of filling the threshold value. The standard threshold value in accordance with the Otsu Thresholding method is 150. The process will be continued by determining the binary value of the result pixel. This process will be repeated until all image pixels have been processed. This looping process will be done horizontally first by reading the pixels from column 1 to the column width of the input image. After that, the looping process will be done vertically by reading pixels from row 1 to the number of rows from the input image. Figure 9 is a flowchart algorithm used for image binaryization. Broadly speaking, the quadtree algorithm can be described in the following pseudocode

```

btnProses.Enabled = False
GetBoxArea()

'Tampilkan(kotak)
nJlh = 0
nIndeks = 1
For i = 1 To Area.GetUpperBound(0)
    If Area(i).Tepi <> "" And Area(i).Jlh > 500 And Area(i).Pilih And
Area(i).MinBaris <= ImgOriginal.Height \ 2 Then
        nJlh += 1
        TampilkanKotak(nIndeks, i)

```

```

        nIndeks += 1
    End If
Next

txtJlh.Text = nJlh          '-----

Dim stop_time As Date = Now
Dim elapsed_time As TimeSpan = stop_time - start_time
Dim Waktu As String

Waktu = elapsed_time.Minutes.ToString & " : " & elapsed_time.Seconds.ToString & " .
" & _
        elapsed_time.Milliseconds.ToString

txtLama.Text = Waktu
End Sub

```

2. Binarization Process

Image preprocessing process is the process of changing the image into the form of image binarization, so that the image will only have 2 types of colors, namely black and white. The binary process is absolutely done before the grayscale process is carried out because the image binary process is done by comparing the results of the grayscale image calculation. In this study, the otsu thresholding method will be used to obtain the threshold value used in the image binary process, so that it can be compared if the pixel has a grayscale value > from the threshold value, then the pixel will be set in white, otherwise the pixel will be set to black. In order to better understand the work process of otsu thresholding, a simple example is given below:

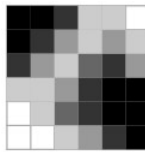


Figure 6. Example of Grayscale Image Input

The binarization process with the thresholding method is as follows:

- Create a histogram diagram for each color in the input image. The results obtained can be seen in the diagram below.

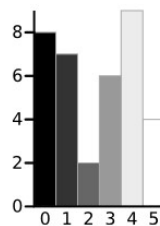
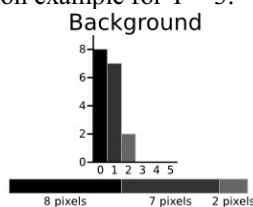


Figure 7. Diagram of Grayscale Image

- For each color element, calculate the value of weight, mean and variance. Computation example for $T = 3$.



$$\begin{aligned}
 \text{Weight } W_b &= \frac{8 + 7 + 2}{36} = 0.4722 \\
 \text{Mean } \mu_b &= \frac{(0 \times 8) + (1 \times 7) + (2 \times 2)}{17} = 0.6471 \\
 \text{Variance } \sigma_b^2 &= \frac{((0 - 0.6471)^2 \times 8) + ((1 - 0.6471)^2 \times 7) + ((2 - 0.6471)^2 \times 2)}{17} \\
 &= \frac{(0.4187 \times 8) + (0.1246 \times 7) + (1.8304 \times 2)}{17} \\
 &= 0.4637
 \end{aligned}$$

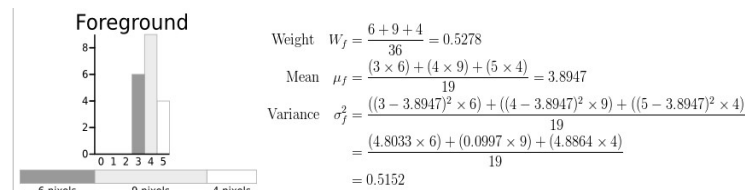


Figure 8. Computation Class Variance

Then compute within class variance

$$\text{Within Class Variance } \sigma_W^2 = W_b \sigma_b^2 + W_f \sigma_f^2 = 0.4722 * 0.4637 + 0.5278 * 0.5152$$

$$= 0.4909$$

- c. Finally, each color in the input image will be changed using the following rules:

If the color value < 3 , then change to black color

If not, then change to white

The results obtained can be seen in the following figure:

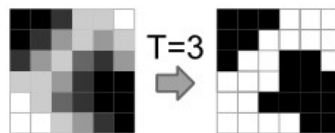


Figure 9. Binarisation Result

The process begins by entering the input image followed by scanning pixels. This process is representing the degree of gray in the input image in mapping pixels into numeric data in the 24-bit true color format. The pixels in the dimensions of the array are height and width according to the pixel size of the input image. Next is the grayscaling process, which is the process of determining colors from red, green and blue components into a single component with an 8-bit format. The process is continued in the initialization sequence, namely the process of dividing the input image area into quadratic tree format. The method used is to divide the input image into 4 quadrants as long as the pixels in the region have differences in intensity. The process runs in repetition for each quadrant with the indicator of repetition being completed there is no difference in the intensity of pixels or up to the size of the smallest quadrant. The process of color segmentation is first done in each region in a minimum size of quadrant and quadrant consisting of homogeneous pixel intensity. The merging process in all quadrants will combine each pixel connectivity in the neighboring quadrant. This process works in continuous looping until all quadrants are grouped according to the region in each object segment.

The application processes the input pixel of the vehicle in the data dimension to classify the pixels that are still connected and the similarity of pixel intensity. The process is carried out by the commands contained in the loop that work continuously in grouping pixels until all pixels are successfully marked. The first quadtree algorithm process is the initialization of object data and temporary data 1. Data. Searching pixels in temporary 1 data by arranging spatial dimensions is useful so that spatial does not come out of the dimensions of the digital image. Search that is marked with the same pixel intensity and does not belong to any segment will be classified as a segment of the object in the data object. Marked pixels are in the quadrant that does not have homogeneity so the pixels are copied into the temporary data 2. While the pixels contained in the quadrant have homogeneity, the quadrant is classified as an object segment. Pixels in the quadrant will be selected that still do not have free connectivity in the temporary data 2. After the search is finished as a marker that the process has obtained one object segment then the temporary data 1 will be reset and filled again from pixel data from temporary 2. Then look for pixels other free to initialize a new warning and temporary 1. The process will stop if there are no new pixels that can be initialized [15].

2.3. Sample

Took 10 samples of video traffic where each video was tested one by one for the collision detection. One of these videos is a video taken from Simpang Pos 2 Medan where the author managed to detect a possible collision that will occur with a red mark.

3. Results

Vehicle Collision Detection Application on Highway Through Webcam Camera Using the Quadtree Algorithm requires input in the form of a video file with the format *.avi, *.flv and *.mp4 which will then be captured by the video screen and stored into a digital image file. Examples of display output applications can be seen in the following picture:



Figure 10. Form Capture Image from Video

The process of detecting collisions on cars using the Quadtree Algorithm can be seen in the following figure:



Figure 11. Image detection process

Based on the results of tests conducted on the system designed, it can be summarized in table form as follows:

Table 1. Collision Detection Testing Result

| Testing | Type Of Accident | Accuracy |
|---------|------------------|--------------|
| 1 | Single | right |
| 2 | Single | not exactly |
| 3 | Single | right |
| 4 | Multiple | almost right |
| 5 | Single | not exactly |
| 6 | Single | almost right |
| 7 | Single | right |
| 8 | Single | right |
| 9 | Multiple | right |
| 10 | Multiple | right |

4. Conclusion

After completing the process of designing a vehicle collision detection application on a highway through a webcam camera using the Quadtree algorithm, some conclusions can be drawn as follows. The application is able to detect car collisions on the highway using the Quadtree algorithm. Based on the results of the tests conducted, information was obtained that the accuracy of the application of the Quadtree algorithm in the detection of car collisions on highways was 8: 2 where the value was obtained from the number of collisions detected. Collision detection failure occurs when the image entered is a picture of a condition after a collision, not an image when a collision occurs. The explanations namely numbers 1, 3, 7, 8, 9 and 10 were declared successful, and experiments 4 and 6 were stated to be almost successful because the collision points missed the detector slightly and experiments 2 and 5 were declared unsuccessful because the collision point with the detector was not appropriate.

References

- [1] Duman CK, Putra MYA, 2013, **Aplikasi Kamera Pendeteksi Mobil Menggunakan Pendekatan Pengolahan Citra**, ISSN 2089-8673, Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI), Volume 2, Nomor 3.
- [2] Himamunanto AR, Tora S, 2016, **Algoritma Quatree untuk Segmentasi Citra Digital Rambu Lalu Lintas**, Jurnal InFact Volume 1, Nomor 2.
- [3] Irawan F, Purnomo A, Alamsyah D, 2016, **Deteksi Mobil pada Citra Digital Menggunakan C-HOG dan Support Vector Machine**, STMIK GIMDP.
- [4] Angelo A, 2016, *A Brief Introduction to Quadrees and Their Applications*, Style file from the 28th Canadian Conference on Computational Geometry.
- [5] Virgono A, Sunarya U, Jauhariah SW, 2016, **Perancangan Sistem Pengendali dan Monitoring Kecelakaan Mobil Berbasis Vehicular Ad Hoc Network (Vanet) Menggunakan Sensor Limit Switch dan Rotary Encoder**, e-Proceeding of Engineering : Vol.3, No.1, ISSN : 2355-9365.
- [6] Susetiyo RAF, Triyanto D, Suhardi, 2016, **Rancang Bangun Smart Vehicle untuk Mendeteksi Dini Kecelakaan dengan Pelaporan Visual pada Google Maps**, Jurnal Coding, Sistem Komputer Untan, Volume 04, No.3 (2016), hal.53-63 ISSN : 2338-493X.
- [7] Alamsyah D, 2017, **Pengenalan Mobil pada Citra Digital Menggunakan HOG-SVM**, Jatisi, Vol. 1 No. 2 Maret 2017, ISSN: 1978-1520.
- [8] Rahim R, 2014, **Algoritma Quadtree untuk Pendeteksian Tubrukan pada Permainan Adventure of Upik**, Jurnal TIMES, Vol III No 2 : 13-18, 2014, ISSN : 2337 – 3601.
- [9] Musfiroh L, Jazuli A, Latubessy A, 2014, **Penerapan Algoritma Collision Detection dan Boids pada Game Dokkaebi Shooter**, Prosiding SNATIF Ke-1 Tahun 2014 ISBN: 978-602-1180-04-4.

- [10] Meenakshi PC, Meenu S, Mithra M, Leela Rani P., 2012, ***Fault Prediction using QuadTree and Expectation Maximization Algorithm***, International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868, Foundation of Computer Science
- [11] Varade S, Ingle M, 2013, ***Hyper-Quad-Tree based K-Means Clustering Algorithm for Fault Prediction***, International Journal of Computer Applications (0975 – 8887), Volume 76– No.5, August 2013. FCS, New York, USA, Volume 2– No.4, May 2012 – www.ijais.org.
- [12] Eppstein D, Goodrich MT, Jonathan Z. Sun, 2005, ***The Skip Quadtree: A Simple Dynamic Data Structure for Multidimensional Data***, arXiv:cs/0507049v1 [cs.CG].
- [13] Rahma MD, Siahaan DO, Soelaiman R, 2016, ***Rancang Bangun Aplikasi Perepresentasian Data Perilaku Pengemudi Mobil Berbasis Android Menggunakan Sensor Accelerometer dan Orientation***, JURNAL TEKNIK ITS Vol. 5, No. 2, (2016) ISSN: 2337-3539 (2301-9271 Print).
- [14] Ahfas A , Hadidjaja D, 2014 , ***Rekayasa Sistem Peringatan Dini untuk Keselamatan Pengendara Kendaraan Berbasis Mikrokontroler Atmega16***, Jurnal Teknologi Informasi DINAMIK Volume 19, No.2, Juni 2014 : 171-178, ISSN : 0854-9524.
- [15] Tzouramanis T, Vassilakopoulus M , Manolopoulos Y , 2000, ***Overlapping Linear Quadrees and Spatio-Temporal Query Processing***, The Computer Journal, Vol. 43, No. 4.
- [16] Sutoyo, T., et. al., 2009, ***Teori Pengolahan Citra Digital***, Penerbit Andi, Yogyakarta.
- [17] Kautsar HVA , Adi K, 2016, ***Implementasi Object Tracking Untuk Mendeteksi Dan Menghitung Jumlah Kendaraan Secara Otomatis Menggunakan Metode Kalman Filter Dan Gaussian Mixture Model***, Youngster Physics Journal ISSN : 2302 – 7371, Vol. 5, No. 1, Januari 2016, Hal 13 – 20.