Proceedings of the 34th Chinese Control Conference
July 28-30, 2015, Hangzhou, China

Real-time Traffic Cone Detection for Autonomous Vehicle

HUANG Yong, XUE Jianru

Visual Cognitive Computing and Intelligent Vehicle Lab

Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University 710049
E-mail: jrxue@mail.xjtu.edu.cn

Abstract: Traffic signs recognition is a basic task for autonomous vehicle. Among the numerous traffic signs, traffic cone is a very important mark used to guide cars where to go. A method based on vision and radar sensors information fusion is proposed to detect traffic cone in this paper. The algorithm mainly includes two parts: finding where the obstacle is in the image and recognizing whether it is a cone. We use homography to calibrate camera and radar, from which the radar data can be mapped on the image and a small corresponding image patch can be easily cutout. Then, a method based on contour feature called chamfer matching is used to determine whether the obstacle in the image patch is a cone. The approach has been tested on our autonomous vehicle, which shows it can guarantee both effectiveness and instantaneity.

Key Words: traffic cone, autonomous vehicle, real time, calibration, chamfer matching

1 Introduction

With the development of researches on autonomous vehicle, intelligent technology such as object recognition is becoming a more and more hot topic. Among the numerous objects, traffic cone as a guided traffic sign is needed to be recognized.

For autonomous vehicle, to recognize cone has two tough problems. 1) The speed of autonomous vehicle is becoming higher, which can reach 40km/h in urban city. The vehicle must see and know the cone at least 30m away if it wants to switch to the cone-guided mode, meanwhile, most of cones are 35cm width and 70cm high, which is a very small size in camera's and radar's point of view, especially in such a long distance. This particularity makes sensors hard extract features. 2) The instantaneity becomes the second problem.

In this paper, camera and radar sensors are fused to solve the instantaneity problem. To solve calibration between camera and radar, paper [4, 5] comes up with an automatic method called homography. They both find the correspondence between radar data and camera data by detecting features in images, which is complex and can't guarantee accuracy if error exists in the feature extraction procedure. Instead of automaticity, we find the homography relationship manually, which shows simpleness and effectiveness. Then the obstacle's radar data is used to find the corresponding image patch. This approach avoids searching cones in the whole image and guarantee instantaneity.

Although famous detectors such as HOG+SVM [2] have a satisfied detection result, they are hard to be applied to real-time requirement applications. Moreover, the little difference of traffic cones makes it hard to collect positive samples, which is troublesome for classifier based methods. Paper [1] presents detecting cone using color feature only, which is not so robust if there exists an area with the same color in a complex environment. To get a better recognition result, we combine both shape and color feature. A method called chamfer matching [7] is used to determine the candidate cone region in the image patch. Then color feature is added for further judgment to make a more reliable result.

The rest of this paper is organized as follows. In section 2, the homography is used to get image patch as for recognition. In section 3, we present the method to recognize image patches using chamfer matching and color feature. Section 4 shows the experiments results. Finally, we conclude in section 5.

2 Image Patch Acquisition

Object detection in one whole image usually can't guarantee instantaneity, meanwhile, too many useless information will cause false detection. With fusion of camera and radar's information, we cutout image patches containing obstacles and do recognition in the patches, which conquers the two above-mentioned problems.

2.1 Homography Review

Homography means the correspondence between two sets of points P and P'. P and P' have homography relationship if they satisfy the following three conditions.

- P and P' have the same number of points.
- Every point's coordinate in P has the corresponding point in P'.
- The points $\mathbf{x}$ in P are all in a plane, the same with P'.

The relationship is a three-by-three matrix $\mathbf{H}$ called homography matrix.

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \tag{1}$$

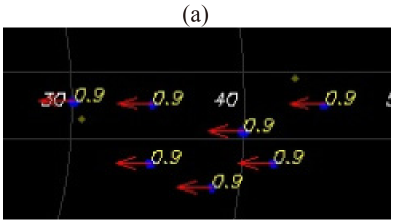
Where $\mathbf{x}$ and $\mathbf{x}'$ are homogeneous coordinates.

2.2 Calibration

For an obstacle like cone, the four-layer radar we use can only give two-dimension position information, which means it cannot give the obstacle's height value .We cluster them as a square box, containing length and width attributes. The blue dot in Fig. 1 (b) visually displays the radar data of cone in image (a).

*This work is supported by National Natural Science Foundation of China(NSFC) under Grant 61273252 and 91320301.

(a)



(b)

Fig. 1 (a) one image containing cones (b) the radar's data of cones in (a)

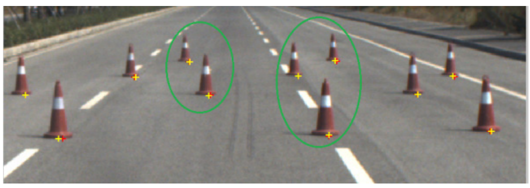
To find the corresponding pixels on image for each radar data, we need the calibration of camera and radar. A common method [5] calibrates radar and camera separately, following the procedure $\mathbf{x}_l \xrightarrow{t} \mathbf{x}_w \xrightarrow{t} \mathbf{x}_c \xrightarrow{t} \mathbf{x}_i$ to map the radar data on image. $\mathbf{x}_l$, $\mathbf{x}_w$, $\mathbf{x}_c$, $\mathbf{x}_i$ is the obstacle's position in radar coordinate system, world coordinate system, camera coordinate system and image coordinate system respectively, t means transformation. Each transformation has a deviation more or less, so the three transformations can't guarantee the accuracy unless the calibration between camera and radar is perfect. Instead of the repeated transformations, paper [4, 5] propose using homography for calibration automatically by detecting a designed feature in images, which is a bit of complex and can't ensure accuracy if error exists in feature extraction.

We find a simpler and effective way based on homography. Firstly, put some objects with small section like cones on a flat road. The small section makes their radar data be a dot like Fig. 1(b). Then follow two steps:

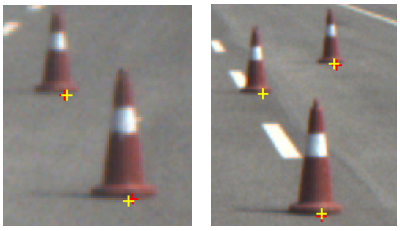
Step 1: For each cone's position $\mathbf{x}_l$ radar gives, the corresponding center position $\mathbf{x}_i$ of the cone in the image needs to be marked manually and $\mathbf{x}_i$ must be on the road. There should be more than four pairs of $(\mathbf{x}_l, \mathbf{x}_i)$, actually the more the better.

Step 2: For the pairs $(\mathbf{x}_l, \mathbf{x}_i)$ acquired in Step 1 satisfies the three conditions of homography, homography matrix $\mathbf{H}$ can be calculated using linear method [6].

Fig. 2 shows the homography result. The red cross '+' is $\mathbf{x}_i$, and the yellow cross '+' is $\mathbf{H}\mathbf{x}_l$,they are almost at the same position in the image.



(a)



(b) (c)

Fig. 2 Homography result of $(\mathbf{x}_l, \mathbf{x}_c)$

2.3 Image Patch

Section 2.2 presents how to get the corresponding position in image for an obstacle center, which can be used to cutout image patches containing the cone according to the simple imaging theory

$$h = \frac{fH}{pD} \tag{2}$$

(f is camera's focus, p is camera's pixel size, H is cone's height, D is distance between camera and cone, and h is pixel's number cone's height takes).

Since the cone's height H and width W are all known and D can be given out by radar data, we can cutout the image patch roughly. To make sure the full cone is in the patch, sufficient margin is needed. Some samples are given in Fig. 3, (a)~(c) are positive samples and (d) ,(e) are negative samples. Patch (b) contains many cones because they are so close that radar regards them as one obstacle. This is also a big challenge for recognition. Patch (d) only contains a part of the car.

(a) (b) (c) (d) (e)

Fig. 3 Image patches cutout using method proposed

3 Cone Detection

3.1 Chamfer Matching Introduction

We consider template matching is a good way to detect traffic cone because the cone to be recognized have the same shape, color and size. In a mass of matching algorithms, chamfer matching which uses shape as feature is a robust method taking instantaneity into consideration.

Chamfer matching is done between a binary image(BW) and a template. First, the BW to be matched should be transformed to a distance image(DT)[7]. Fig. 4 shows the result. (b) is the idea shape of the cone in image (a) and we call the white pixels foreground while the black pixels background, (c) is the DT of (b). For each pixel in (c), the pixel value stands for the distance between that pixel and the nearest foreground pixel of (b), so the pixel color in (c) changes gradually from white to black foreground-centered. Then the template of cone is matched on DT(c) through rotation, translation and resizing.

The mean square average is chosen to minimum the matching measure

$$D(T,I)=\sqrt{\frac{1}{n}\sum_{i=1}^{n}v_i^2} \quad (3)$$

Where v_i is the distance value and n is the points' number of shape. In application, the template is considered matched on at location where matching measure $D(T,I)<\alpha$, α is a threshold. The smooth property of DT makes it more robust to optimize the target function (3). Fig. 4 (d) shows matching result. The green pixels are the template's best match position on DT through rotation, translation and resizing. The matching position is right but the scale has been narrowed compared with Fig. 4 (b).

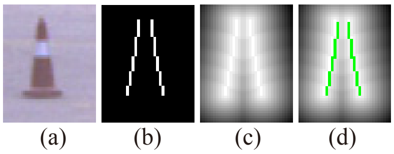


Fig. 4: (a) cone patch (b) idea cone shape of (a) (c) DT of image(b) (d) chamfer matching result

3.2 Edge detection

The first step to use chamfer matching is contour detection, or edge detection more specific. The precision of cone's edge detection affects matching result directly. The reason chamfer matching have a good result in Fig. 4 is that the edge feature is the same with template. But how to detect a satisfied edge itself is a difficult problem. Malik's team make a significant contribution to this field and some state of the art methods [6, 7] have been came up with. They train a contour detector using brightness, color and texture as input features, then use this detector to judge every pixel is a contour point or not. The method performs very well, especially on nature images. However, it is seconds even minutes time costing, which can't meet real-time requirement. Taking time into account, we use canny [12] edge detector.

Canny is a famous edge detector combining both effectiveness and instantaneity. It has two main parameters to control the result of edge detection, the low threshold and the high threshold. It's hard to keep all the edge points of cone while filter out noise. Fig. 5(b) shows the edge detection on (a), the noise points we don't want also has been detected out. Some other factors like shadow, lane line will make it harder for detection. Fig. 5(e) is the edge result of (d), the shadow cause false detection seriously, and (f) is the false matching result. The reason why it causes false detection and how to solve this problem will be discussed in section 3.4.

To get rid of the false detection of cone edge, we use edge's direction information θ :

$$\theta=\arctan\left(\frac{G_y}{G_x}\right) \quad (4)$$

(G_y and G_x are gradients respectively, which can be calculated in gray image).

Since every edge points have a direction and cone is supposed to stand on the road, we consider the direction $|\theta|\in[\frac{\pi}{12},\frac{\pi}{6}]$ is the main edge points of cone. Fig. 5 (g) is edge points after direction chosen on (e). The image shows the edge points whose direction don't satisfy the requirement are filtered out, including all the noise edge points and a small part of cone's edge. The lost edge points of cone won't affect matching. (h) is the chamfer matching result using (g), the right position has been detected out compared with (f).

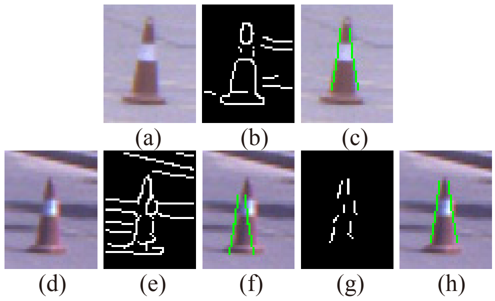


Fig. 5 (a)~(c),(d)~(f)are original image, canny edge detection result and chamfer matching respectively. (g) edge points after direction choosing on (e) (h) the matching result using (g)

3.3 Threshold Chosen

Each image patch will get a score $D(T,I)$ (3) after using chamfer matching, we need to judge the detection is cone or not using formula

$$Cone=\begin{cases} true\leftarrow D(T,I)<\alpha \\ false\leftarrow D(T,I)\geq\alpha \end{cases} \quad (5)$$

To select a proper α , we choose 400 positive samples and 600 negative samples manually. Then using normal density estimation to estimate the probability density function (PDF) of matching score respectively. The PDF is drawn in Fig. 6 (a), the blue line is positive sample's PDF and the red one is the negative sample's PDF.

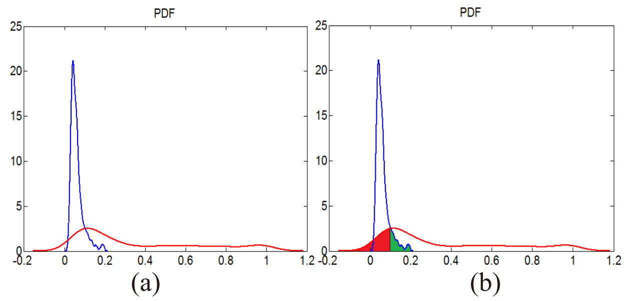


Fig. 6 (a) Probability density function (PDF) of cone's and non-cone's matching score. The blue one is the cone's PDF and the other is the non-cone's PDF. (b) Error probability of classification

According to minimum error of Bayes Decision theory [13] on two classes

$$p(w_i\mid\mathbf{x})=\frac{p(\mathbf{x}\mid w_i)P(w_i)}{\sum_{j=1}^{2}p(\mathbf{x}\mid w_j)P(w_j)} \quad (6)$$

(Where $\mathbf{x}$ is the feature, in this paper it is matching score. w_i is the class).

We can select α as 0.1 from Fig. 6 (a). Fig. 6 (b) shows the error probability of classification. The red region area is 0.2, which means it has 20% probability to classify non-cones as cone. And the same meaning with the green region, whose area is 0.1.

3.4 Color Judgment

Although chamfer matching has a good result, there are two phenomena will cause false detection.

- The edge feature extracted is similar to the cone template. Fig.7 (d) shows a false detection on image (a).
- The second and also the most serious one is the obstacle has a very complex edge feature, which will leads to false detection as expected.

Fig.7 (f) shows the edge feature of (e), and (g) is the DT image. All pixels value in (g) are very small due to the foreground pixels in (f) are full of the whole patch. No matter how the template translates, rotates and scales, the matching score is always very small. This property makes chamfer matching hard to use in clutter environment. (h) shows the false matching result, the yellow number on (h) is the matching score 0.082.

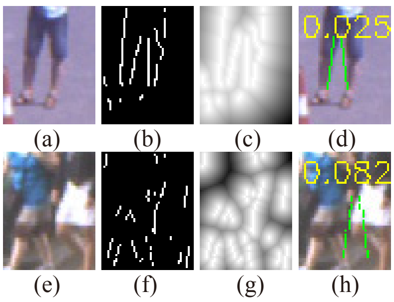


Fig. 7: Two false detection.(a)~(d) is original image, edge feature, DT and match result, the same with (e)~(h).

To make matching result better, we further use color information. This is reasonable because all cones have a main color: red.

Many methods based on color information for recognition problem have been proposed up. Paper [10] comes up with a color histogram matching method, which have a good result in a particular scene. But it's hard to be applied in outdoor environment because the matching algorithm is designed too simple. A more robust histogram matching method called Earth Mover's Distance (EMD) [11] may solve the problem but it's too expensive. Considering chamfer matching has been used beforehand, a simple color segmentation can satisfy our application.

Since the position of cone in the image patch has been detected out, we can use the color information inside the template contour. First, a small box (ROI) needs to be cutout like Fig.8 (a).Then we count percentage the red region takes in H channel to judge whether it is a cone. The whole algorithm steps:

Step 1: Cutout small box from the image patch according to the template position and size.

Step 2: Transforming ROI from RGB space to HSV space. Selecting red pixels in H channel whose value is in $[0,0.1)\bigcup(0.85,1]$. Then counting the red pixels percentage ϕ of ROI. Using the same method in section 3.3, the positive sample's PDF (the blue curve) and negative sample's PDF (the red curve) are show in Fig. 8 (b). The red color's percentage of positive samples mainly distribute between 0.2 and 0.8, whose PDF shows a Gauss likeness model. So we choose $[0.2, 0.8]$ as the cone's judgment interval, and the green region area shows the probability of mistaking non-cone as cone. The interval can be adjusted according to real application. If one wants a higher accuracy rate, the interval can be narrowed, which of course will cause a lower recall rate.

Table 1: Detection Result on 400 positive samples and 600 negative samples

Method	Recall Rate	Accuracy Rate
Chamfer Matching(CM)	90.0%	69.2%
CM + Color	82.3%	84.5%

Table 1 shows the detection result on training dataset mentioned in section 3.3. The result proves color information can improve accuracy rate under a tolerable lower recall rate.

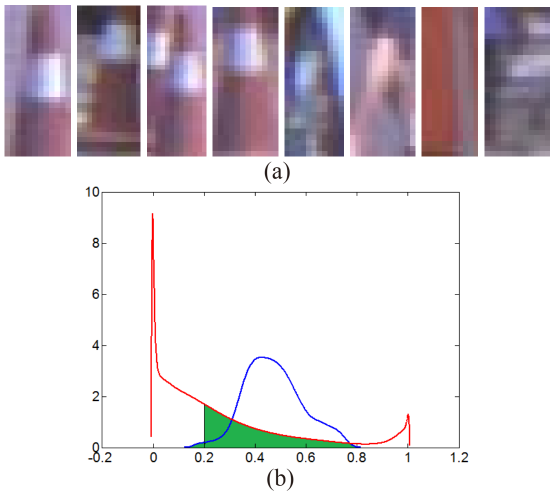


Fig. 8: (a) Small box containing cone and non-cone extracted according to chamfer matching (b) PDF of cone and non-cone's red color percentage

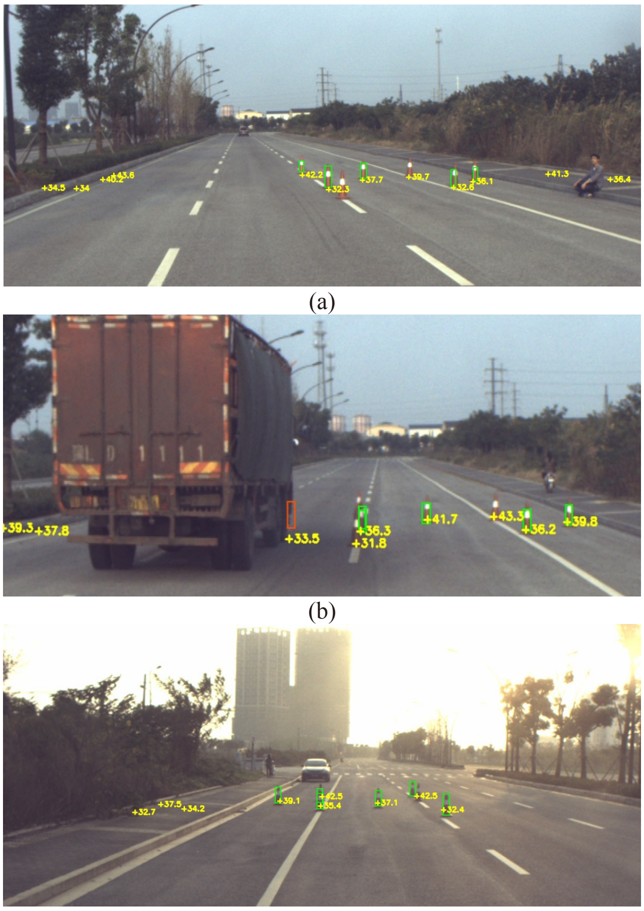
4 Experiments

The proposed real-time cone detection algorithm has been evaluated on our autonomous vehicle, which installs a four-layer laser radar and a 1294×964 resolution basler industrial camera with a 8mm lens. This big resolution of camera makes cone have enough pixels for edge detection in a long distance. The algorithm is implemented in C++ language on a matrix IPC with an i7 core CPU and it costs about 4ms for one image patch's detection.

Although chamfer matching can resize the template to match images theoretically, to make algorithm more robust, we prefer to make different size's templates so the image can be matched with a sizable template. In our application, we only detect cones which are 30~45 meters away from the autonomous vehicle. So we make the three templates at 32m, 37m and 42m, which are used to match cones whose distance is in [30m, 35m), [35m, 40m) and [40m, 45m) respectively.

The two main thresholds: chamfer matching score α and red pixels percentage ϕ, both use values that have been discussed in this paper in section 3.3 and 3.4 respectively.

Our autonomous vehicle has tested the algorithm in different scenes. Fig. 9 (a) shows one image which is taken back to the sun. The yellow '+' means radar data mapped on the image and the number near it denotes distance the obstacle from the vehicle. An obstacle which has been detected as a cone is surrounded by a green box. There are seven cones in the image but only six radar data are given out because two of them are so close that radar regards these two cones as one obstacle. One of the seven cones is undetected out. The red box in Fig. 9(b) means it is accepted by chamfer matching but rejected by color judgment. Fig. 9 (c) shows one more challengeable image captured facing the sun. All of the cones have been detected out, which demonstrates the illumination robustness of our proposed algorithm. Cones which are in a long distance ($\geq$ 35m) are almost all detected out.



(a) (b) (c)

Fig. 9 Detection result of images captured back to the sun and facing the sun

5 Conclusion

The method proposed in this paper mainly focus on two problems: calibration between radar and camera, cone detection. It uses homography to calibrate radar and camera, then mapping the radar data on image. According to imaging theory, an image patch containing a cone or non-cone obstacle can be cut out. Then chamfer matching algorithm is used to judge the obstacle in it is a cone or not. To make result more reliable, color feature is also used.

By combing information from two sensors: radar and camera, our algorithm can detect cone effectively and real-time. Our future work will focus on detecting more fixed shape objects like pedestrian or cars based on this method.

References

[1] SHI J, WANG G, WANG J. Recognition of traffic cone in micro-traffic environment[J]. *Journal of Hefei University of Technology (Natural Science)*, 2012, 12: 014.

[2] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C].*Computer Vision and Pattern Recognition*, 2005. *CVPR* 2005. *IEEE Computer Society Conference on. IEEE*, 2005, 1: 886-893.

[3] Yang H, Liu X, Patras I. A simple and effective extrinsic calibration method of a camera and a single line scanning lidar[C]. *Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE*, 2012: 1439-1442.

[4] Sugimoto S, Tateda H, Takahashi H, et al. Obstacle detection using millimeter-wave radar and its visualization on image sequence[C].*Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. IEEE,* 2004, 3: 342-345.

[5] Gao D, Duan J, Yang X, et al. A method of spatial calibration for camera and radar[C].*Intelligent Control and Automation (WCICA), 2010 8th World Congress on. IEEE*, 2010: 6211-6215.

[6] Richard Hartley, *Multiple View Geometry in Computer Vision, Second Edition*, New York: Cambridge University Press,2003:314-320.

[7] Borgefors G. Hierarchical chamfer matching: A parametric edge matching algorithm[J]. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1988, 10(6): 849-865.

[8] Martin D R, Fowlkes C C, Malik J. Learning to detect natural image boundaries using local brightness, color, and texture cues[J]. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2004, 26(5): 530-549.

[9] Arbelaez P, Maire M, Fowlkes C, et al. Contour detection and hierarchical image segmentation[J]. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2011, 33(5): 898-916.

[10] Swain M J, Ballard D H. Color indexing[J]. *International journal of computer vision*, 1991, 7(1): 11-32.

[11] Rubner Y, Tomasi C, Guibas L J. The earth mover's distance as a metric for image retrieval[J]. *International Journal of Computer Vision*, 2000, 40(2): 99-121.

[12] Canny J. A computational approach to edge detection[J]. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1986 (6): 679-698.

[13] Sergious T, Konstantinos K. *Pattern recognition, Forth Edition*. America: Academic Press Inc.2008:13-18.