# Vehicles Recognition Based on Point Cloud Representation

Patrik Kamencay(✉), Robert Hudec, Richard Orjesek, and Peter Sykora

Department of Multimedia and Information-Communication Technologies, University of Zilina, Univerzitná 1, 010 26 Zilina, Slovakia
{patrik.kamencay, robert.hudec, richard.orjesek, peter.sykora}@fel.uniza.sk

**Abstract.** The following article is dedicated to techniques for recognition of vehicles on the road. By using 3D virtual models of vehicles, it is possible to create database of point cloud. The SSCD algorithm for training and testing was used. First for each 3D model the point clouds were created. Then from each point cloud one hundred pictures were rendered from different projections. Creation of filtered dataset was done by selection six angles from these projections. This dataset contains 100 models of vehicles divided into 5 classes. In summary, final non-filtered dataset contains 10 000 pictures, filtered dataset consist of 600 pictures. Dataset was used in support vector machine (SVM) and convolutional neural network (CNN) for training and testing in ratio 80:20. The result for SVM was 40%, this was done because non-filtered dataset contains many similar projections. Moreover, the size resulted in long duration of experiment (<90 h). Therefore, other experiments were done with filtered dataset. In filtered dataset, best result in SVM was 79% with RBF kernel. For the next experiment, CNN was used. With data augmentation the result was 80%, without 89%.

**Keywords:** Point cloud · Deep learning · Convolutional Neural Network SVM · Stereo system · 3D model · Vehicle · CNN · SSCD

## 1 Introduction

In the computer vision, the main idea of object recognition is creation and representation of the objects in particular classes. Which characterize appearance of these objects and in this way, determine the class for the unknown object [1–4].

In this paper, we are focusing on recognition of 3D modeled vehicles. There were chosen 5 classes for the recognition, namely sedan or coupe, small cars, SUV, cabriolets and trucks.

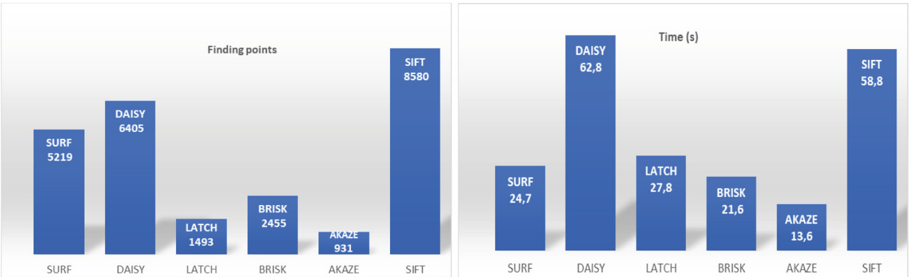
To develop a system, which is capable of acceptable vehicles point cloud recognition, the good training and testing data were required. Collection and creation of these dataset in the real-world roads and highway is very hard and time consuming. Thus, we used virtual 3D models of vehicles for our experiment. It is the new approach to simple and fast creation of dataset.


T. Kováčiková et al. (Eds.): INTSYS 2017, LNICST 222, pp. 79–84, 2018.
https://doi.org/10.1007/978-3-319-93710-6_9

The rest of the paper is organized as follows: the first part compares point cloud detection methods. The second part describes selected point cloud method. In fourth chapter, the creation of dataset is described. The experimental results can be found in the last chapter.

## 2 Key Point Detection

The key point detection is very important step during point cloud creation process. For the most optimal search and point detection, few algorithms were tested. Final algorithm was selected with respect to operational speed and the amount of found points. Tested algorithms were: DAISY [5], AKAZE [6], LATCH [7], SIFT [8], SURF [9] and binary descriptor BRISK [10]. All the results are displayed in the Fig. 1. The Fig. 1a represents amount of founded points. The DAISY and SIFT found the biggest amount of key points. The Fig. 1b represents the duration of the algorithm. The fastest was AKAZE, but it found the smallest amount of key points. The best ratio points/time has SURF. This method will be used.




a) b)

**Fig. 1.** Comparison of local detection method: (a) amount of key points, (b) time consuming.

## 3 Key Point Detection

By using stereo camera system, we obtain pair of images (one from left camera and one from right camera). These images are processed with algorithm which transforms them into point cloud [11–21]. This process consists of 8 steps:

- The first step loads left and right picture for the given model (see Fig. 2A).
- The second step deletes the camera deformation – chip and lens distortion what makes spherical and chromatic aberration (see Fig. 2B).
- The third step deletes the background and thus segments the vehicles. All the parameters are set according to calibration (see Fig. 2C).
- The fourth step is to find key points on left and right images (see Fig. 2D).
- The fifth step connects those points which are identical for left and right images (see Fig. 2E).

- The sixth step is rectification which filters the points and searches only for pairs on epipolar lines (see Fig. 2F).
- The seventh step is to calculate distant of these pairs from camera.
- The last step is to create the depth maps and point cloud using the points mentioned above (see Fig. 2G).

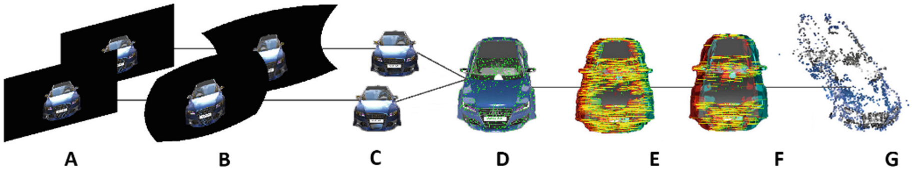



**Fig. 2.** Point cloud process.

For However, for these images, rendered pictures can be used. The renders are from virtual 3D models of vehicles. It is possible to replace this system with 3D virtual scene with real parameters and measurements. To make the pictures look genuine and photorealistic, HDRi maps must be used. The map provides information about natural sunlight and surrounding environment.

In addition, a precise configuration of cameras and distortion lenses are required. The simplest way to obtain point cloud from 3D models is by using SSCD [22] algorithm which can make point cloud directly from existing 3D models. This method was used in this paper.

## 4 Database

For testing purposes, as was mentioned previously we used 3D virtual models. The dataset was divided into training and testing subsets. These point clouds are subsequently projected into 6 different front views from different angles and positions.

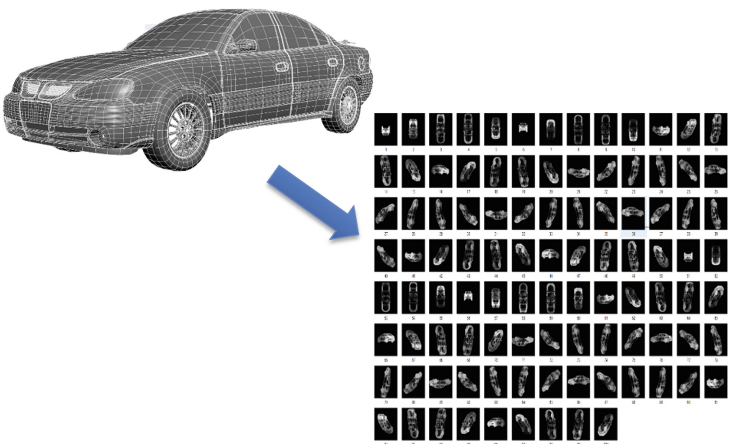


**Fig. 3.** 3D model SSCD projections.

Each projection had a maximum resolution of 200 × 200 pixels because of training and testing speed.

The database consists of 100 models divided into 5 classes. These are namely sedan or coupe, small cars, SUV, cabriolets, and trucks. Thus, database consisting of 600 pictures. Some examples of projections are shown in Fig. 3.

## 5 Testing and Results

The testing was done by using MATLAB and Python environments. The database was divided in ratio of 0.8. This ratio can be modified. The experimental results show the SVM implementation achieving 40% accuracy on the original non-filtered dataset. This result was due to high number of similar projections in the dataset, while the extensive size of dataset resulted in long duration of undertaken experiments (<90 h). To overcome this problem, the original dataset was reduced through the filtering, resulting in 79% accuracy for SVM method with RBF kernel. The additional experiments were conducted using CNN method, achieving higher accuracy of 80% without data augmentation, while the significant improvement of 89% was achieved with data augmentation (DA) applied (see Table 1).

**Table 1.** Results SVM, SVM with kernel RBF, CNN and CNN with DA.

| | SVM | SVM + RBF | CNN | CNN + DA |
|---|---|---|---|---|
| Non-filtered dataset | 40% | – | – | – |
| Filtered dataset | 46% | 79% | 80% | 89% |

## 6 Conclusion

Vehicle recognition based on point cloud were presented. By using 3D virtual models of vehicle, we were able to create dataset of point cloud for training and testing purposes by using SSCD algorithm. Then we used Convolutional Neural Network and SVM for vehicle recognition. System with CNN and data augmentation achieved 89% accuracy. In this case data augmentation increased accuracy from 80% to 89%. The precision can be improved by beforehand data processing, especially picture segmenting, higher resolution. Furthermore, to achieve better results in CNN [23] we can use different kernels and change parameters. It is possible to improve CNN accuracy on smaller dataset by using data augmentation which increase amount of input data by using various operations applied on input images.

## 7 Future Work

In future development, we would like to create consistent dataset of real vehicles and generated vehicles. We would like to focus on improving of precision in recognition and cooperate with transporting companies in order to acquire more data. Moreover, we would like to develop a consistent system for car detection and recognition in real time.

**Acknowledgements.** The paper was prepared under the support of the ERDF European Regional Development Fund, project No. ITMS26220120028, "Centre of excellence for systems and services of intelligent transport".

## References

1. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision, Thomson Learning, Part of the Thompson Corporation (2008). ISBN: 10: 0-495-24438-4
2. Cao, T.P.: Object Recognition (2011). ISBN 978-953-307-222-7
3. Khoshelham, K.: Accuracy analysis of kinect depth data. In: ITC Faculty of Geo-information Science and Earth Observation, University of Twente, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVIII-5/W12, 2011 ISPRS Calgary 2011 Workshop, 29–31 August 2011, Calgary, Canada (2011)
4. Bertozzi, M., Broggi, A., Fascioli, A., Nichele, S.: Stereo vision-based vehicle detection. In: Proceedings of the IEEE Intelligent Vehicles Symposium 2000 Dearbon (MI), USA, 3–5 October 2000
5. Lowe, D.G.: Object recognition from local scale-invariant features. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2. IEEE (1999)
6. Pablo, F., Alcantarilla, J.N., Bartoli, A.: Accelerated-KAZE features, Fast Explicit Diffusion for accelerated features in nonlinear scale spaces. In: British Machine Vision Conference (BMVC), Bristol, UK (2013)
7. Levi, G., Hassner, T.: LATCH: learned arrangements of three patch codes. Department of Mathematics and Computer Science, The Open University of Israel (2015)
8. Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. In: Proceedings of Computer Vision and Pattern Recognition, Alaska, USA (2008)
9. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). https://doi.org/10.1007/11744023_32
10. Leutenegger, S., Margarita, Ch., Siegwart, R.Y.: BRISK: binary robust invariant scalable key points. In: 2011 IEEE International Conference on Computer Vision (ICCV). IEEE (2011)
11. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision. IEEE (1999)
12. Kim, J.S., Kim, H.W., Kweon, I.S.: A camera calibration method using concentric circles for vision applications. In: ACCV2002, Melbourne, Australia (2002)
13. Zhang, Z.: Camera calibration with one-dimensional objects. IEEE Trans. Pattern Anal. Mach. Intell. **26**(7), 892–899 (2004)
14. Faugeras, O.: Three-Dimensional Computer Vision: A Geometric Viewpoint. MIT Press, Cambridge (1993)
15. Mihalik, J., Michalcin, V.: 3D motion estimation of human head by using optical flow. Radioengineering **15**(2), 37–44 (2006)
16. Jiang, G., Zhao, C.: Camera calibration based on 2D-plane (2010)
17. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, New York (2003)
18. Loop, Ch., Zhang, Z.: Computing rectifying homographies for stereo vision. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE (1999)

19. Kupper, A.: Location-Based Services, Fundamental and Operation. John Wiley and Sons, Ltd., West Sussex (2005)
20. Grauman, K., Leibe, B.: Visual recognition. University of Texas at Austin - Computer Science (2009)
21. Mikusova, M.: Crash avoidance systems and collision safety devices for vehicle ocupants. In: MATEC Web of Conferences: Dynamics of Civil Engineering and Transport Structures and Wind Engineering, DYN-WIND 2017, vol. 107 (2017)
22. Šatnik, A., Orjesek, R., Hudec, R., Kamencay, P., Jarina, R., Talapka, J.: A novel approach for 3D model recognition based on SSCD. In: ELEKTRO, pp. 96–99 (2016)
23. Krizhevsky, A., Sutskever I., Geoffrey, E.H.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (2012)