Advanced Science and Technology Letters
Vol.51 (CES-CUBE 2014), pp.77-80
http://dx.doi.org/10.14257/astl.2014.51.18

Cloud Robot with Real-Time Face Recognition Ability

Shuqing Tian, Dilshat Saitov, Suk Gyu Lee

Yeungnam University, Korea
{tianshuqing@gmail.com, dilshat_saitov@yahoo.com, sglee@ynu.ac.kr}

Abstract. This paper presents a cloud robot system with real-time face recognition ability. Cloud robot systems which support face recognition have been use for surveillance. In order to recognize a face in real time, the images captured by camera have to be stored and then processed by face recognition algorithm. Benefit from cloud computing technology, robot can offload computational tasks to cloud server which composes of several computational units (PCs) and is capable of executing distributed tasks. Our experimental results demonstrate that as a implementation of cloud robotics system, the proposed cloud robot is capable of doing real-time face recognition.

Keywords: Cloud robot, cloud computing, real-time face recognition

1 Introduction

Cloud robotics is the combination of robots and cloud computing technology. Cloud computing technology allows robots to benefit from the powerful computational, storage, and communications resources of modern data centers. In addition, it removes overheads for maintenance and updates, and reduces dependence on custom middleware .

Robots with face recognition application have been developed and used in practice. In the paper [1], a novel approach for face recognition using multiple face patterns obtained in various views for robot vision was introduced. The paper [2] presents a design and experimental study of human-robot interaction via face recognition and image tracking. However, these implementations are limited since the computational ability of embedded robot. Benefit by cloud robotics technology, cloud robot with real-time face recognition application comes true. In our approach, the challenge of long latency between the robot and the cloud server has been solved which makes images be transmitted in real time. In this paper, we proposed our method on implementing a cloud-based and real-time face recognition system for robot.

2 The Architecture of Implemented Cloud Robot

In the implemented architecture, cloud server plays a role of “brain” in the system. However, one of the well-known challenges for using the cloud as a server is the long latency between the mobile device and the cloud server in comparison to localized computing and small-scale distributed computing called cloudlet [6]. Given this challenge, the proposed architecture must support high-speed data transmission and distribute computing load to achieve the minimal response time. Regard to data transmission, short range wireless communications have to be considered.

The Wifi protocol is employed to transmit image data in our approach. Bluetooth as a common-used protocol is used for send command between robot and cloud server in our system.

The architecture of our cloud robot is shown in figure 1. The robot operating system has been used as the operating system for cloud server. The implemented face recognition algorithm is an executable program which is called “node” in ROS. All developed libraries are stored in the “core function units”, which is separated from ROS build-in functions. Necessary information is stored in database, such as the system configuration, user information, command list and etc.. In the client side, the robot is supposed to support Wifi and Bluetooth communication.

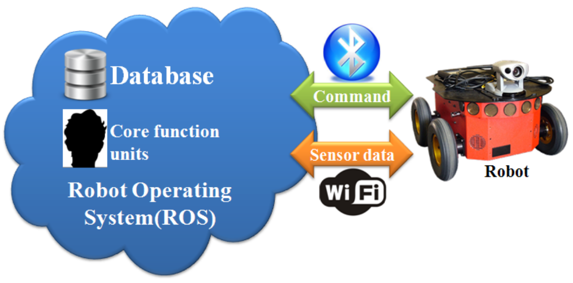

Fig. 1. Cloud robot system

3 Face recognition algorithm and implementation

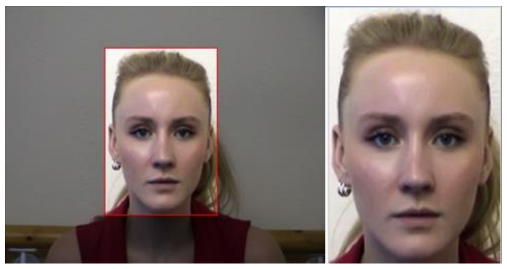

Fig. 2. Face detection result

As shown in Figure 2, the face detection determines the potential locations of the human faces within an image. We have utilized the Haar Features and Haar Classifiers described in [4] to perform face detection. This iterative approach begins with fairly primitive classifiers that group potential face candidates based on a small number of features. These simple classifiers in this initial stage have low computational complexity but must operate on a large amount of data, and they produce a large number of face candidates. The algorithm then progressively eliminates some of these candidates by using increasingly more sophisticated classifiers based on additional features at successive stages of the detection pipeline, such that the final stage outputs the detected faces with high confidence. Although the number of remaining candidates is significantly less at each successive stage, the complexity of the calculations increases at almost the same rate, and thus the overall computational complexity of each pipeline stage of this detection algorithm stay somewhat constant. In our implementation, we use a 32-stage pipeline.

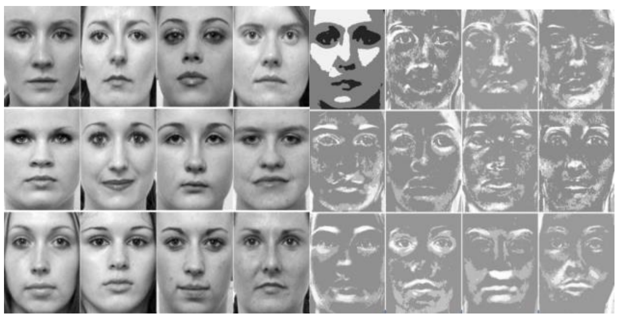

Fig. 3. Template image and eigenface

The face recognition determines the match-likelihood of each face to a template element from a database. The potential locations of the faces determined in the previous face detection phase are fed into this phase for recognition. The recognition algorithm yields one of a few potential results for each face candidate determined by the detector: (1) not a face, (2) a face, but not in the database, and (3) a face and in the database. We have employed the widely-accepted Eigenfaces approach [3], which calculates an orthogonal set of M Eigenfaces for a given training set of N faces, where M N. Each face from the original N faces can be represented as a point within the M dimensional space spanned by the M Eigenfaces. This permits a significant reduction in the amount of computation that has to be performed to recognize a face within a given database, as well as a significant reduction in the amount of storage required for the template images (database). To recognize a face, the algorithm simply transforms the face into a point within the M-dimensional space spanned by the Eigenfaces and calculates the Euclidean distances between the point of the face to be detected and the points of all template faces from the database. If the Euclidean distance is above a large threshold, meaning that the detection algorithm yielded a false positive. Otherwise, if the Euclidean distance is above a small threshold to all the template faces, the algorithm determines that the face is not in the database. In this case, if desired, this face can be added to the database by re-calculating the Eigenfaces and the Eigenfaces representation of the newly introduced face and updating the databases of all the cloud servers.

4 Conclusion

We have presented a cloud robot which excuses face recognition task. The architecture of presented cloud robot is an implemented prototype of cloud robotics. In the architecture, the ROS has been used to execute distributed tasks, like face recognition in our project. We have taken the advantage of cloud robotics which offloads computational intensive tasks form robot to cloud server. Robot supporting Wifi protocol can be the client side in our proposed system and communicate with the cloud server. In addition, if the robot carries a camera and is capable of sending image to the cloud server, face recognition function can be executed.

References

1. Hu, G., Tay, W. P., Wen, Y.: Cloud robotics: architecture, challenges and applications. Network, IEEE, May-June 2012, Volume: 26, page 21-28
2. Quintas, J., Menezes, P., Dias, J.: Cloud Robotics Towards Context Aware Robotic Networks. Proceedings of the IASTED International Conference, November 7 - 9, 2011 Pittsburgh, USA
3. RoboEarth, http://www.roboearth.org/
4. Google cloud robotics: http://www.google.com/events/io/2011/sessions/cloud-robotics.html
5. Rajesh, A., Vikas, R.E., Liu, B.b., Wu, X.j., Krishnamoorthy B. Foong F.K., A.Senthil K., Kang D.M., Goh W.K: DAvinCi: A Cloud Computing Framework for Service Robots. In: IEEE International Conference on Robotics and Automation,Anchorage, Alaska, USA, 2010.
6. Yuka, K., Toru, I., Yosuke, T., Masahiko, N., Miwa, U., Yoshihiko, M., Keijyu O.: RSi-Cloud for Integrating Robot Services with Internet Services. In: 37th Annual Conference on IEEE Industrial Electronics Society, Tokyo, Japan, 2011.