Fig. 10 shows the average time and average number of searches to search each tree level in each image. When the objects were distributed widely or evenly as in image 3 or image 11, the search time was short. When there were many spaces without objects as in image 1, image 2, or image 6, the search time was long. This occurred because the proposed search algorithm searches sequentially from the top left to the bottom right. This phenomenon can be understood in the context of the proposed algorithm having the same disadvantage that occurs in sequential search algorithms, i.e. that the time complexity is O(N) in the worst case scenario, and the algorithm becomes more inefficient as the size of the array becomes larger [Parmar *et al.*, 2015].

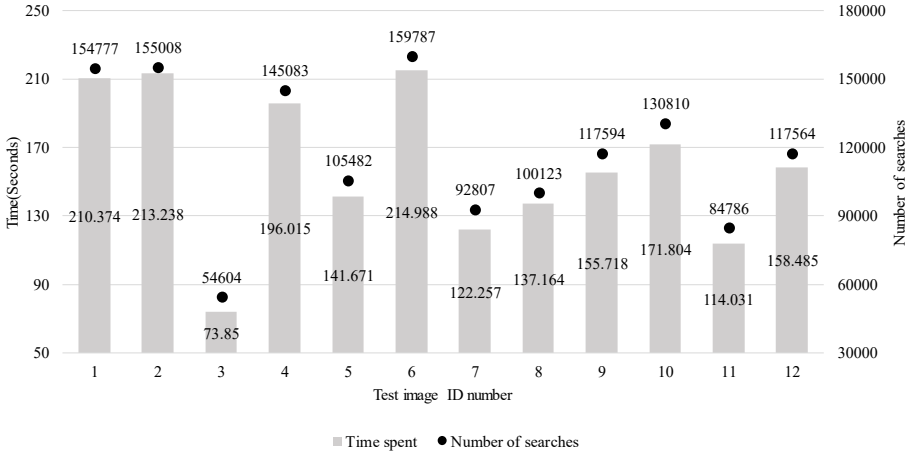

Fig. 10. The average time spent on searches and average number of searches for each test image.

The experiments showed that in the worst case, the search was performed for a similar time and number of pixels as the brute force search. However, in the best case, if the number of searched tree levels was *l*, the objects' distribution could be known by searching 4^*l* pixels, and the difference in computation time became larger as the number of levels became smaller. Furthermore, it was possible to find the approximate distribution of the objects more quickly than the brute force search in the experiments using the images from Fig. 8. Therefore, it can be seen that the proposed method is more efficient in terms of the search time and number of searches than the brute force search when finding the approximate distribution of objects.

5. Conclusions and Future Research

This paper has proposed a scheme that uses a quadtree structure to search for and save objects in order to quickly search for the approximate location of objects in a certain space. Experiments have shown that the proposed method can find the approximate distribution and area of objects without searching an entire *n*m* space that has *n* number of horizontal pixels and *m* number of vertical pixels. Future studies will first extend the size of the space to FHD and test the search time and number of searches. Second, we aim to gain the algorithm's performance improvement by applying other search method.

Acknowledgments

This paper was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (Ministry of Education) [NRF-2017R1D1A1B03035884, NRF-2018R1D1A1B07045642].

References

[1] Parekh, H.S.; Thakore, D. G.; Jaliya, U. K. (2014): A survey on object detection and tracking methods. International Journal of Innovative Research in Computer and Communication Engineering, 2 (2), pp. 2970-2978.
[2] Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, Liu, X.; Pietikäinen, M. (2019): Deep learning for generic object detection: A survey. International Journal of Computer Vision, 1-60.
[3] Samet, H. (1985): Data structures for quadtree approximation and compression. Communications of the ACM, 28 (9), pp. 973-993.
[4] Samet, H. (1984): The Quadtree and related hierarchical data structures. ACM Computing Surveys (CSUR), 16 (2), 187-260.
[5] Pixabay is available on https://pixabay.com.
[6] Parmar, V. P.; Kumbharana, C. K. (2015): Comparing linear search and binary search algorithms to search an element from a linear list implemented through static array, dynamic array and linked list. International Journal of Computer Applications, 121 (3), pp. 13-17.