Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 100 (2016) 128 – 135

Procedia Computer Science

Conference on ENTERprise Information Systems / International Conference on Project MANagement / Conference on Health and Social Care Information Systems and Technologies, CENTERIS / ProjMAN / HCist 2016, October 5-7, 2016

ENVISION: Assisted Navigation of Visually Impaired Smartphone Users

Shoroog Khenkar[a*], Hanan Alsulaiman[a], Shahad Ismail[a], Alaa Fairaq[a]
Salma Kammoun Jarraya[a,b,] and Hanêne Ben-Abdallah [a,b,]

[a] *FCIT, King Abdulaziz University, Jeddah, Saudi Arabia*
[b] MIRACL Laboratory, Sfax, Tunisia

Abstract

In this work, we propose ENVISION an assistance system for safe navigation of visually impaired smartphone users. The proposed system generates an intelligent decision to manage the navigation of visually impaired people based on the fusion of GPS technology directions and a new obstacle detection method. It copes with many challenges related to such application processing steps and inherent to constraints of the smartphone platform. These include illumination changes, diversity of background and road textures, low quality of the video streams, and low processing capacity. ENVISION uses a new method to detect static and dynamic obstacles robustly and accurately in real-time video streaming recorded by a smartphone with an average hardware capacity.

Peer-review under responsibility of the organizing committee of CENTERIS 2016

Keywords: Navigation tools; Obstacle detection; Visually impaired people; Machine learning

1. Introduction

The lack of visual information about the surrounding environment imposes major challenges on a visually impaired person in performing daily life activities, such as mobility. Indeed, navigation can be extremely difficult and

* Corresponding author. Tel.: + 966 536 864 860
E-mail address: skhenkar@stu.kau.edu.sa

Peer-review under responsibility of the organizing committee of CENTERIS 2016
doi:10.1016/j.procs.2016.09.132

dangerous, especially in unfamiliar environments. Recent attempts have been made to employ software solutions using different technologies and devices.

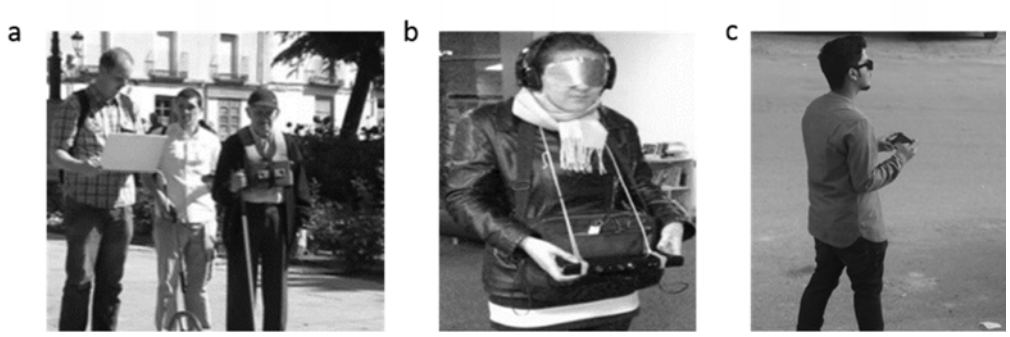

Fig. 1. (a) A visually impaired user carrying a stereo camera using a chest mounted harness [4]; (b) A blind-folded user equipped with the depth camera (a Microsoft Kinect) [5] ; (c) Our proposed smartphone ENVISION system does not require the user to carry any additional devices

Among the proposed solutions developed to assist visually impaired people to navigate safely is Microsoft 3D Soundscape [1]. It uses a headset that talks to the user through their routes in cities. In addition, it uses location information from Microsoft's Bing maps, which can be supplemented using tiny Bluetooth-enabled beacons stuck to lampposts around the city. A second assistance system is TrAVEl [2] which aims to assist the user in their bus routes. It uses GPS to track the traveller and bus arrivals to give a running commentary about bus stops and route, and to alert the traveller if he/she needs to change buses. On the other hand, ViaOpta Nav [3] walks a visually impaired person through their route by offering useful, turn by turn directions from the user's current position to their final destination. The aforementioned systems can produce specific semantic information such as "an obstacle in front of the user", "the arrival of a bus". However, they are restricted by requiring certain types of platforms or special types of hardware (*cf.* Fig. 1), providing information related to buses directions only *(e.g.*, TrAVEl), or not providing speech recognition capability (*e.g.*, ViaOpta Nav). Furthermore, these solutions do not provide real-time obstacle detection (neither static nor dynamic obstacle). These limits motivated us to propose ENVISION, which is a system capable of recognising speech, guiding the user to the requested destination, detecting static and dynamic obstacles and generating intelligent decisions to walk the user safely by avoiding obstacles and changing their route accordingly when needed.

The remainder of this paper is organised as the following: In section 2, we first present our proposed approach for ENVISION and then detail the new method for detecting static and dynamic obstacles in real-time video streaming. In section 3, we detail the major off-line work prerequisite to developing our proposed obstacle detection method. Experimental evaluation is discussed in section 4. Finally, the proposed approach is summarised and future work directions are presented in section 5.

2. Overview of ENVISION and its Obstacle Detection Method

The development of the ENVISION system has a three-fold objective. The first objective is to remove special hardware requirements and produce a solution deployable on an average range smartphone configuration. The second objective is to produce a solution that is a user-friendly as possible. To attain this second objective, our system uses speech recognition to receive destination requests from the user and it produces voice directives and alerts to avoid potential obstacles. The third objective is to produce a robust, high performant obstacle detection method.

Before detailing this latter objective which is the main focus of this paper, we next overview the conceptual architecture of ENVISION.

2.1. ENVISION Conceptual Architecture

As illustrated in Fig. 2, our proposed system ENVISION operates in four steps: (1) speech recognition, (2) path finding, (3) obstacle detection, and (4) merging phase.

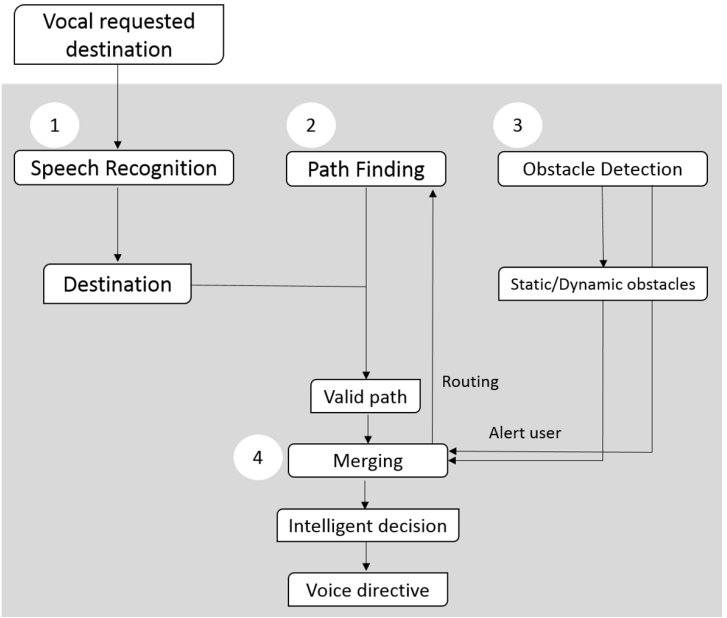

Fig. 2. Conceptual architecture of ENVISION

In step (1), it recognises the requested destination using (Google Voice API[†]) and passes it to step (2) to find a valid path to the destination. ENVISION implements step (2) using GPS technology (Google Maps API[‡] and Google Maps Directions API[§]). Furthermore, the robustness of ENVISION relies on its capability to detect static and dynamic obstacles during the navigation of the visually impaired people. Thus, we propose a new method to detect static and dynamic obstacles in video streaming recorded by smartphone (step 3). Finally, in the merging phase (step 4), our system generates an intelligent decision representing an appropriate voice directive and an alert to the user when an obstacle is detected. In the following section, we focus on the obstacle detection step since it is a critical step in our system.

2.2. New Obstacle Detection Method

In [6-9] works, Range-based and Intensity-based approach, these basically use laser scanners to obtain range and intensity values for 2D or 3D mapping. Other approaches use image feature descriptors, e.g. pixels appearance, image grid and interest points. However, there are assumptions and limitations of some previous approaches such as detecting a certain type of obstacles only or assuming that the ground is relatively flat or there are no overhanging obstacles and obstacles are relatively far from the user. To overcome some of these limits, the herein proposed method adopts a

[†] https://developers.google.com/voice-actions/
[‡] https://developers.google.com/maps/
[§] https://developers.google.com/maps/documentation/directions/

supervised learning process to produce *four* accurate adaptive prediction models PM_{high}, PM_{low}, PM_{left} and PM_{right} used as an input for our method. Each prediction model is used to treat one portion of the captured frame.

More specifically, our method for static and dynamic obstacle detection in real-time video streaming recorded by smartphone operates in the three main steps shown in Algorithm 1. First (line 1 of Algorithm 1), it divides each input frame into the five regions (high, low, left, right and unwanted region) schematised in Fig. 3. The purpose of this first step is to prioritise obstacles according to their relative distance from the user. The high region is the closest region to the user, while the low region is the furthest. The left and right regions are used for routing in case of an obstacle is detected in either the high or low regions. The fifth region, i.e. the unwanted part, is discarded due to the effect of user's feet on the prediction process. In the second step (line 2 of Algorithm 1), the regions are pre-processed to transform the representation of the regions and formulate vector descriptors using different image features, the produced vector will be passed as an input in the third step for the related PM (Prediction Model) PM_{high}, PM_{low}, PM_{left} and PM_{right}. In third step (line 3 of Algorithm 1), the received input is passed to the PM, and obstacles are detected based on the semantics of the input feature vector. PM_{high}, PM_{low}, PM_{left} and PM_{right} are prerequisite output of major off-line work. Further details of the off-line work are presented in the next section.

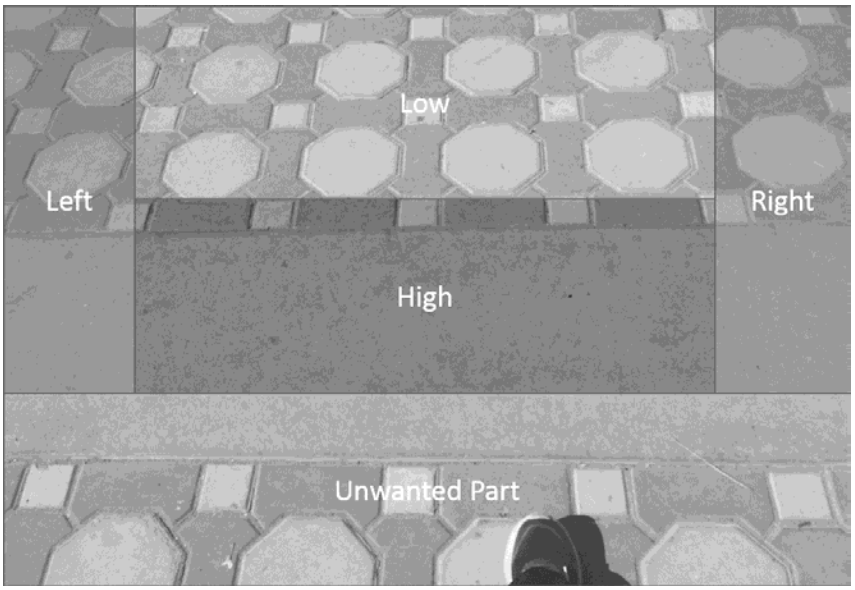

Fig. 3. Dividing input frames into 5 regions: high, low, left, right and unwanted region

Algorithm 1: Obstacle Detection

input : S a stream of input frames, ***PMhigh***, ***PMlow***, ***PMleft*** and ***PMright***

output: Voice directive

F Number of frames in stream **S ;**
for i ←1 to F **do**
1.1 Divide frame i into 5 regions;
2.1 Extract features of the regions;
3.1 **if** PM High (high) == obst **then**
3.1.1 **if** PM Left (left) != obst **then return** decision = go left;
3.1.2 **else if** PM Right (right) != obst **then return** decision = go right;
3.1.3 **else return** decision = stop;
3.2 **else** repeat the step 3.1 for low region using ***PMlow***;

3. Off-Line Work for Prediction Models Generation

In this section, we detail the steps followed for generating the four prediction models (***PMhigh***, ***PMlow***, ***PMleft*** and ***PMright***) which are related to the high region, low region, left region and right region, respectively. We have adopted supervised machine learning approach to classify regions into one of two predefined classes: Obstacle and Non-obstacle. The classification process maps input dataset x into one of the predefined classes y, using target function f. The input dataset is a collection of vectors. Each vector represents an instance of a region obtained from dividing an input frame into regions (line 1.1 of Algorithm 1). Each PM is generated using a learning algorithm of decision trees. Considering supervised learning constraints, the efficiency of the generated PM is increased with the increase in the size of the training set and the number of relevant features. Another pertinent setting to consider, the selection of the appropriate learning algorithm.

Accordingly, we started by considering large and representative corpus. We have manually classified about 52000 images into obstacle/non-obstacle. After that, we identified the set of features to build an *n*7* matrix (7 features) from our training corpus where *n* is the number of instances in the training corpus. We note that we have investigated the reduction of the number of features, however, using less than 7 features resulted in unsatisfactory results and using more than 7 features caused heavy processing time and more battery consumption. Thus, using 7 features was the ultimate solution. Each row represents an image of a region that either contains an obstacle or non-obstacle, and each column represents a feature. The last column indicates the class type, either 0 (non-obstacle) or 1 (obstacle). The selected features used to describe our dataset are: (Var 1): RGB color histogram [8], (Var 2): Grey scale histogram [8], (Var 3): HoG Histogram of Oriented Gradients [9], (Var 4): MSER Maximally Stable Extremal Regions [10], (Var 5): SIFT Scale-invariant feature transform [11], (Var 6): LBP Local Binary Patterns [12] and (Var 7): HSV color histogram [13].

After preparing the corpus, we selected the appropriate learning technique. We selected a well-established technique, namely the induction of decision trees. Compared to other supervised learning techniques, decision trees are easier to understand, implement and convert to if-else statements in any programming language without the need to provide any additional interpreters. In contrast, artificial neural networks (ANN), for instance, behaves like a "black box" where the flow of decisions can be more difficult to understand and explain in addition to the need of a separate interpreter for interpretation. In addition, to select one induction of decision trees algorithms, we examined six data mining algorithms of decision trees: Limited Search Tree Algorithm [14], ID3-IV [15], GID3 [16], ASSISTANT 86 [17], ChAID [18] and C4.5 [15]. We examined these algorithms in terms of several validation techniques including Precision (P) the retrieved instances that are relevant, and Recall (R) relevant instances that are retrieved. To do so, we split the corpus into a training data (70%) and testing data (30%). The application of these algorithms revealed that the best Precision and Recall are achieved by: C4.5 for High Region: Class A: precision (97%) and Recall (98%), Class B: precision (97%) and Recall (93%), ChAID for Low Region: Class A: precision (99%) and Recall (99%), Class B: Precision (99%) and Recall (99%), ChAID for Left Region: Class A: precision (99%) and Recall (98%), Class B: precision (98%) and Recall (99%) and ChAID for Right Region: Class A: precision (98%) and Recall (98%), Class B: Presidion (98%) and Recall (98%), note that class A represents the non-obstacle class while class B represents the obstacle class.

4. Experimental Results

In order to validate our contribution, ENVISION was submitted to an intensive set of three experiments aiming to evaluate its robustness in terms of its efficiency in managing the navigation of visually impaired people (experiments 1) and detecting static and dynamic obstacles from real-time video streaming (experiments 2), and acceptance by final end users (experiments 3). The following subsections provide a brief description of the dataset, validation techniques, experiments conditions and results for each experiment.

4.1. Experiment 1

For the first experiment, 445 frames obtained from real-time videos were used to evaluate the performance of the four prediction models ***PMhigh***, ***PMlow***, ***PMleft*** and ***PMright*** relative to high region, low region, left region and right

region receptively in terms of Precision and Recall given by equation (1) and (2) for class A (Non-obstacle), (3) and (4) for class B (Obstacle). There was no predefined assumptions or constraints on the testing data or the environment. In fact, the dataset covers many canonical challenges such as, the high speed of dynamic obstacles within the video streaming (*cf.* Figure 4a), the effect of different lighting conditions including sunny weather and large dark shadows of the trees and buildings on the ground (*cf.* Figure 4b) and the wide variety of textures and background colors and obstacles appearances (*cf.* Figure 4c).

$$Precision_A = \frac{n_{AA}}{n_{AA}+n_{BA}} \quad (1)$$

$$Recall_A = \frac{n_{AA}}{n_{AA}+n_{AB}} \quad (2)$$

$$Precision_B = \frac{n_{BB}}{n_{BB}+n_{AB}} \quad (3)$$

$$Recall_B = \frac{n_{BB}}{n_{BB}+n_{BA}} \quad (4)$$

Fig. 4. (a) Moving persons around the visually impaired user; (b) The effect of sunny weather and shadows of the trees; (c) The variety of textures, colours and appearances

Table 1 and Fig. 5 present, respectively, the quantitative and qualitative results obtained from experiment 1. The results shown in table 1 revealed the Precision rates for obstacle and non-obstacle classes are between 74% and 85%, and the Recall rates are between 70% and 88%. The best recall is recorded by ***PMhigh*** related to high region and the best precision is recorded by ***PMhigh*** related to high region. In summary, our system records average 74% for non-obstacle and 85% for obstacle recall rate and 83% average precision rate for non-obstacles and 77% for obstacles. These results affirm the robustness and the high performance of the prediction models [***PMhigh***, ***PMlow***, ***PMlef***t and ***PMright***] for obstacle detection in real-time video streaming recorded by smartphone with no constraints on the testing environment. Figure 5 represents the qualitative results for significant frames. These results illustrate the robust adaptive behaviour of our method.

Table 1. Quantitative results of experiment 1, Precision and recall rates recorded by PMhigh, PMlow, PMleft and PMright

PMhigh	Recall A = 70%	Precision A = 85%	Recall B = 88%	Precision B = 74%
PMlow	Recall A = 78%	Precision A = 83%	Recall B = 84%	Precision B = 79%
PMleft	Recall A = 72%	Precision A = 83%	Recall B = 85%	Precision B = 76%
PMright	Recall A = 77%	Precision A = 83%	Recall B = 84%	Precision B = 79%

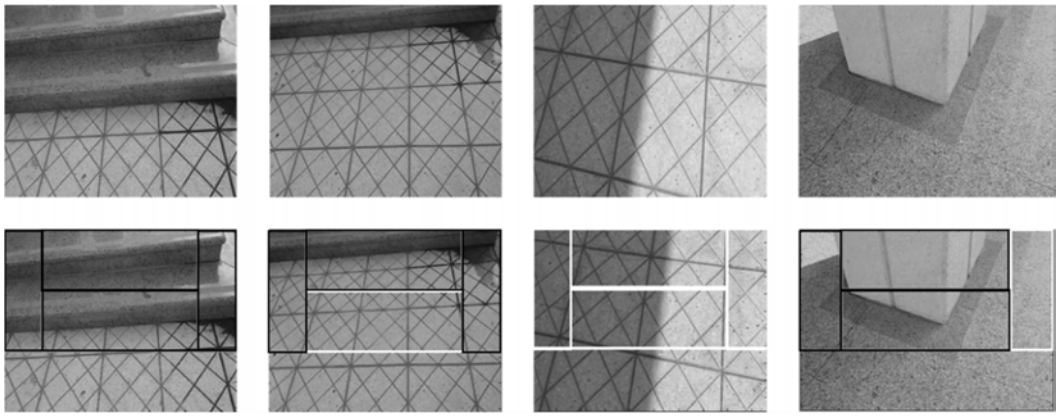

Fig. 5. Qualitative results, the first row presents the images before applying our method while the second one is after applying it, black rectangles indicate regions with obstacles, white rectangles indicate regions with no obstacles

4.2. Experiment 2

In the second experiment, we evaluated the performance of the system in terms of True/False Alerts. Alert Rate measures the number of correct alerts either TP (True Positive) or TN (True Negative) and incorrect alerts that are either FP (False Positive) or FN (False Negative), the data used were obtained from 3 different real-time video streams with different number of frames. No constraints or conditions were applied in this experiment. Table 2 represents the results of this experiment. The correct alerts rate are between 78% and 96% and the incorrect alerts rate are between 4% and 22%.

Table 2. Results of the proposed system in terms of true positive, true negative alerts, false positive and false negative alerts

Video 1 (35 frames)	TP= 96%	TN= 81%
Video 2 (18 frames)	TP= 85%	TN= 88%
Video 3 (18 frames)	TP= 88%	TN= 78%
Video 1 (35 frames)	FP= 4%	FN= 19%
Video 2 (18 frames)	FP= 15%	FN= 13%
Video 3 (18 frames)	FP= 13%	FN= 22%

These preliminary results indicate the effectiveness and accuracy of our proposed method. Recall that the ultimate goal of our work is to provide a new robust and accurate method for obstacle detection without using advance expensive technology that helps to contribute in the improvements of the provided software solutions to assist the visually impaired people to navigate safely in unfamiliar environments. The results obtained for our obstacle detection method provides further evidence for high levels of accuracy despite the few misclassification emerged from the challenges stated earlier above that can be reduced with further improvements to the method.

4.3. Experiment 3

For the third experiment, we have submitted ENVISION to five real visually impaired users to evaluate the system from their perspective. This experiment was conducted to navigate between buildings in the King Abdulaziz University campus. The aim of this acceptance test is to evaluate the usability of ENVISION for first time users. A user profile, post-task and post-test questionnaires were provided and answered by the participants. The results of this test indicates that 80% of the participants found ENVISION easy to learn and use. In addition, 100% of the participants expressed their admiration towards capabilities of ENVISION.

5. Conclusion

In this paper, we have presented ENVISION, an assistance navigation for the visually impaired smartphone users. The proposed system is based on four major processing steps including static and dynamic obstacle detection in real-time video streaming. Thus, a major off-line work was presented to generate four PM namely, $\boldsymbol{PM_{high}}$, $\boldsymbol{PM_{low}}$, $\boldsymbol{PM_{left}}$ and $\boldsymbol{PM_{right}}$ each related to a region to prioritize the decision. Significant challenges are imposed to our method including illumination light, noise induced by user and variety of textures. Our method for obstacle detection was submitted to three experimental evaluations with different aims, measures and conditions. According to the preliminary experimental results, ENVISION is efficiently capable to detect static and dynamic obstacles in dynamic environments with complex obstacles and interactions and it is designed to well fit the needs of the visually impaired users. We can conclude from these preliminary qualitative and quantitative evaluation that ENVISION is robust and efficient. As future work, besides extending these evaluations, we will consider obstacle recognition and classification which helps the visually impaired to better understand their environment.

References

[1] Warnick J. Independence Day. (2016). [online] Independence Day. Available at: URL: http://news.microsoft.com/stories/ [accessed 12 August 2015].
[2] NEO, New bus app for visually impaired and elderly commuters, The Straits Times. (2015). [online] The Straits Times. Available at: URL: http://www.straitstimes.com/singapore/transport/new-bus-app-for-visually-impaired-and-elderly-commuters [accessed 5 March 2015].
[3] V. Nav N. Corporation, ViaOpta Nav on the App Store, App Store. (2016). [online] Available at: URL: https://itunes.apple.com/us/app/viaoptanav/id908435532?mt=8%20www.viaopta.com/ [accessed 6 March 2015].
[4] Rodríguez A, Yebes J, Alcantarilla P, Bergasa L, Almazán J, Cela A. Assisting the Visually Impaired: Obstacle Detection and Warning System by Acoustic Feedback. Sensors 2012;12:17476-17496.
[5] Brock M, Kristensson P. Supporting Blind Navigation Using Depth Sensing and Sonification. In: Ubicomp '13 The 2013 ACM International Joint Conference On Pervasive And Ubiquitous Computing. New York: ACM; 2013.
[6] Hancock, Laser Intensity-Based Obstacle Detection and Tracking, Ph.D. dissertation, The Robotics Institute, Carnegie Mellon University; 1999.
[7] Hesch J, Roumeliotis S. Design and Analysis of a Portable Indoor Localization Aid for the Visually Impaired. The International Journal Of Robotics Research 2010; 29:1400-1415.
[8] Ulrich I, Nourbakhsh, I. Appearance-Based Obstacle Detection with Monocular Color Vision. In: The Seventh National Conference on Artificial Intelligence. Austin, Texas: Association for the Advancement of Artificial Intelligence (AAAI-00); 2000.
[9] Tapu R, Mocanu B, Zaharia T. A computer vision system that ensure the autonomous navigation of blind people. In: E-Health and Bioengineering Conference (EHB). Iasi, Romania: IEEE; 2013.
[10] VLFeat - Documentation MATLAB API, Vlfeat.Org. (2016). [online] VlFeat.org. Available at: URL: http://www.vlfeat.org/matlab/matlab.html [accessed 6 December 2015].
[11] Lowe D. Object recognition from local scale-invariant features. In: ICCV '99 Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2, Washington, DC, USA: IEEE; 1999.
[12] Nissimov S, Goldberger J, Alchanatis V. Obstacle Detection in a Greenhouse Environment Using the Kinect Sensor. Computers And Electronics In Agriculture 2015;113:104-115.
[13] Derhgawen A, Ghose D. Vision Based Obstacle Detection Using 3D HSV Histograms. In: 2011 Annual IEEE India Conference, IEEE; 2011.
[14] Catlett J. Megainduction: Machine Learning on Very Large Databases. PhD Thesis, School of Computer Science, University of Technology. Sydney, Australia. 1991.
[15] Quinlan J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers;1993.
[16] Cheng J, Fayyad U.M, Irani K.B, Qian Z (1988). Improved Decision Trees: a Generalized Version of ID3. In: Proceedings of 5th International Conference on Machine Learning. San Mateo, CA: Morgan Kaufmann; 1988.
[17] Cestnik B, Kononenko I, Bratko I. ASSISTANT 86: A Knowledge Elicitation Tool for Sophistical Users. In: Proceedings of the 2nd European Working Session on Learning. Sigma, Wilmslow, UK; 1987.
[18] Kass G. An Exploratory Technique for Investigating Large Quantities of Categorical Data. Applied Statistics. 1980; 29(2), p.119-127. doi: 10.2307/2986296