Vision and Distance Integrated Sensor (Kinect) for an Autonomous Robot

Paulo Rogério de Almeida Ribeiro, Fernando Ribeiro and Gil Lopes

***Abstract*—This work presents an application of the Microsoft Kinect camera for an autonomous mobile robot. In order to drive autonomously one main issue is the ability to recognize signalling panels positioned overhead. The Kinect camera can be applied in this task due to its double integrated sensor, namely vision and distance. The vision sensor is used to perceive the signalling panel, while the distance sensor is applied as a segmentation filter, by eliminating pixels by their depth in the object's background.**

The approach adopted to perceive the symbol from the signalling panel consists in: a) applying the depth image filter from the Kinect camera; b) applying Morphological Operators to segment the image; c) a classification is carried out with an Artificial Neural Network and a simple Multilayer Perceptron network that can correctly classify the image.

This work explores the Kinect camera depth sensor and hence this filter avoids heavy computational algorithms to search for the correct location of the signalling panels. It simplifies the next tasks of image segmentation and classification. A mobile autonomous robot using this camera was used to recognize the signalling panels on a competition track of the Portuguese Robotics Open.

***Index Terms*—Kinect, Sensors, Autonomous robot.**

I. Introduction

There is a huge variety of sensors such as temperature, strength, distance, sound and many others sensors that are employed in different applications. Robotics is one field which uses them extensively [26].

Darpa Grand Challenge [9] is one example involving many sensors of different kinds. In this task, several autonomous cars must drive itself around 240 Km on the desert, and more recently the urban challenge was performed on a city environment (a desert military city). The information obtained from the sensors is the only resource used to define the tasks, for example, to change the velocity or direction.

The Kinect camera was designed and developed by Microsoft [1] to be used by the video game console XBOX 360. The visionary idea was to develop a new way to play video games, enabling an interaction with the player without the traditional control joystick. Kinect was first planned to be a good video game tool, but soon it was found out that it could be used as a distance and vision sensor for robotic purposes.

The Portuguese Robotics Open [2] hosts robotics competitions, demos and a scientific meeting (International Conference on Mobile Robots and Competitions). One of these competitions is the autonomous driving league, where robots with maximum dimensions of 60x100x80 cm must drive autonomously in a track like a traffic road. The main compulsory challenge of this robot consists of being autonomous, following the track and its two tight curves without touching the outside white lines, avoiding collisions with the tunnel, be aware of the roadwork cones, stopping on the zebra crossing, obey to the traffic light and signalling panel, etc. On this context, one can show the **Formula UM**, see Figure 1, type autonomous robot developed in the lab, which already participates on this competition since 2009.

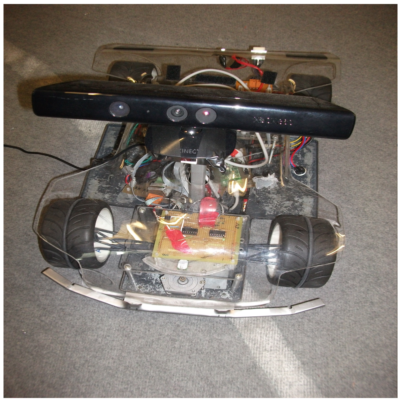

Fig. 1. **Formula UM** type autonomous robot

For the Portuguese Robotics Open - 2011 edition, the Minho Team decided to include one innovation and proposed to apply the Kinect camera as a distance and vision sensor. This camera can be used to filter the track signalling panels. The depth image obtained with this camera is divided into planes, and these contain near and far objects. These planes can be calibrated by the user and the threshold values render them. This filter provides an elimination of some undesired objects on the track using their distances, and only when the signalling panel is between the planes further steps are processed. This approach avoids complex algorithms of recognition and high computation times.

This work is devoted to present the Microsoft Kineck as a vision and distance integrated sensor. A real application to recognize the symbols from a signalling panel using the Kinect camera on an autonomous mobile robot is presented. Image segmentation is processed by morphological operators

Paulo Ribeiro is master student of Mechatronics Engineering: University of Minho - Portugal, e-mail: paulorogeriocp@gmail.com

Fernando Ribeiro, Associate Professor, Departamento de Electrónica Industrial - Robotics Group - Algoritmi Centre - University of Minho, Campus de Azurém, 4800-058, Guimarães, Portugal, e-mail: fernando@dei.uminho.pt

Gil Lopes, Auxiliar Researcher, Departamento de Electrónica Industrial - Robotics Group - Algoritmi Centre - University of Minho, Campus de Azurém, 4800-058, Guimarães, Portugal, e-mail: gil@dei.uminho.pt

and image classification is achieved by an artificial neural network. Although there are many kinds of neural networks, the Multilayer Perceptron network is the most used for this type of task and therefore it was decided to apply it.

This paper is organized as follows. In Section II the state of the art with some applications of the Kinect sensor is presented. Section III shows the Portuguese Robotics Open competition and describes the rules and challenges associated. Section IV brings up the results, including the filter, segmentation and classification. Section V discusses the obtained results. Finally, Section VI presents the conclusions and directions for further work.

II. State of the Art

This section presents the State of the Art of the Microsoft Kinect sensor and also the traffic signal recognition problem.

A. *Kinect sensor*

Microsoft Kinect was designed as a new revolutionary controller for the video game console XBOX 360, however, it was found useful as a video/depth sensor for robotics. Its application can be found on: 3D Scene Generation [30] [28] [25], Robotic [21] [27] [6] [28] [12] [4] [8] [17] and Interaction [7] [14] [28].

On 3D Scene Generation task the Kinect sensor was applied due the ability to generate a depth map of the environment [27] [28] [30]. The authors create a 3D map of the environment using the Kinect camera to perform their tasks.

On Robotic research field one can see its applications on Computer Vision, Autonomous Robot and Environment Mapping. On [27] the computer vision is applied over the depth map performed by the Kinect, which aims to control the altitude of a quadrotor helicopter. On [28] it is employed in order to enable the robot to understand the direction received from human's hand gesture. For autonomous Robot applications using Kinect some examples can also be found such as: Robocup@Home league [4] [8], Soccer League [17] and autonomous cars [6]. [6] applies a Kinect in navigation and in a target tracking task. For [12] the sensor is employed in order to do a Visual Odometry instead of the traditional odometry, which relies on measurement of the wheel motion.

Augmented Reality has as its major task the interaction, therefore, one can emphasise the Kinect as a way to simplify this task. [7] applies this sensor to propose an interaction using the body. On [14] the Kinect is employed to promote high interaction with medical images. The authors emphasise this interaction as a way to perform surgery when non-sterilisable devices cannot be used [14].

Since many authors apply the Kinect to build depth maps directly from the camera raw data [27] [28] [30] [25], image filtering can also be performed as suggested by [22]. Its approach has improved the results in the depth image stability by applying a filter.

Many researchers have begun to use the Kinect as a sensor, i.e., a different purpose for which it was initially designed, namely a video game controller. Its advantages are low cost, good accuracy and high rate [27], however, one can emphasise its low cost [21] [6] [28] [12] as the main advantage. Before this sensor, two cameras and a laser were employed to achieve the same task at a high cost. [7] states that the Kinect sensor reduces the computational cost and the complexity of the system.

The list below shows some device characteristic of this sensor:

1) RGB camera: 8 bit VGA resolution (640 x 480 pixels)
2) Depth sensor: Infrared laser
3) Distance Range: 0.7 meters until 6 meters
4) Angular field view horizontally: 57°
5) Angular field view vertically: 43°
6) Motorized base: Sensor tilting of 27°(up or down)
7) Resolution: 1.3 millimetre per pixel
8) Frame rate: 30 Hz
9) Programming: C/C++/C#
10) USB connection
11) Multi-array microphone

B. *Traffic Signs Recognition*

Traffic Signals Recognition has been a study subject of some research areas such as Artificial Intelligence, Computer Vision and Image Processing. This task can provide valuable input for autonomous cars or as a visual aider for a driver assistance system [20] [15] [19]. The main problems of this task are the signal's detection and recognition. Some solutions for these can be found on [19] [20] [13] [29].

Several techniques are available for image segmentation from the more conventional approach such as thresholding to a more sophisticated approach using Support Vector Machine (SVM) [20]. For the detection task the main approaches are the colour and shape information [10]. Finally, on the recognition task the usual classifiers are: Artificial Neural Network [29] [19], Support Vector Machine [20] [15], Fuzzy [13] and Genetic Algorithm [18].

A road-sign detection and a recognition system for all the Spanish traffic signs was implemented by Maldonado-Bascon [20]. It uses colour and shape information on the segmentation and detection tasks. The author's segmentation approach is tree based. For a specific set of colours only some shapes are possible. For instance, signs with red colour only have circular, triangular and octagonal shapes [20]. The next step (detection) is thus simplified. The segmentation method is divided into chromatic and white signs. While the first is resolved with thresholding (HSI colour space), the last is resolved with an achromatic decomposition. After a segmentation using colour information, the shape information is applied on the detection task with a linear SVM. The proposed detection process using shape features is invariant to translation, rotation and scale. Finally, the recognition stage is done by a SVM with Gaussian kernel, where blobs from the grayscale image are passed as an input for the detection method [20]. This work presents good results on many adverse conditions, such as cloudy and rainy and also at night.

A comparison of several segmentation methods in traffic sign recognition was also found [15]. In order to evaluate each one of these methods, the next steps of the traffic signs recognition task (detection, recognition and tracking) are held the same. In other words, only the first step (segmentation method) is changed for all the images. The measurements applied to evaluate the methods are: *Number of signs recognised*, *Global Rate of correct recognition*, *Number of lost signs*, *Number of maximum* (number of times that a method achieved the maximum score), *False recognition rate* and *Speed* (total execution time in seconds divided by number of used frames). The best segmentation method found by the author was the RGB normalised. More information can be obtained on [15].

Improvements made on the segmentation (approach by thresholding and achromatic decomposition [20]) can be found in the new methods developed [15].

III. Competition

This section describes the Portuguese Robotics Open, more specifically the autonomous driving league. The aspects covered in this section are: Rules, Symbols, Challenges, etc. Another subject presented in this section is the environment to prepare the **Formula UM** robot for the competition.

The Portuguese Robotics Open (Festival Nacional de Robotica in portuguese) [2] is devoted to gather people to share news ideas of robotics in Portugal. It is made of robotics competitions, demos and a scientific meeting (International Conference on Mobile Robots and Competitions). This paper focus the autonomous driving league, where robots must drive autonomously in a track, like a traffic road, surrounded by two parallel side lines and including two lines separated by a dashed mid line. The aim is to have the robot completing a double lap around the track, with the same starting and arrival points with the shortest time and with the least possible penalties [3]. The robot's maximum dimensions are 60x100x80 cm.

Besides driving autonomously the competition has other challenges, for example two tight curves, the road width, one tunnel, roadwork cones, a zebra crossing, signalling panels, traffic signs, etc. The track road and its challenges can be viewed on Figure 2. In the tunnel, the vision algorithms are influenced by the low light and therefore a good way to test their robustness.

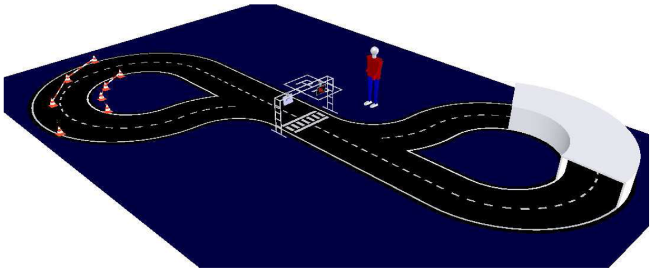

Fig. 2. Competition Track (Obtained from [3])

On Figure 2 one can see the signalling panels, between the eight-shaped curves next to where a human being is standing up. Since it is the main element of this work, more technical details about it are presented on Figure 3 (measures in meter unit). These signalling panels are used to indicate when the robot should start and stop operating similarly as if they were traffic lights. The signalling panels are made of a 17" computer flat screen on inverted position. Since the track has two ways there are two flat panels, each turned to its direction.

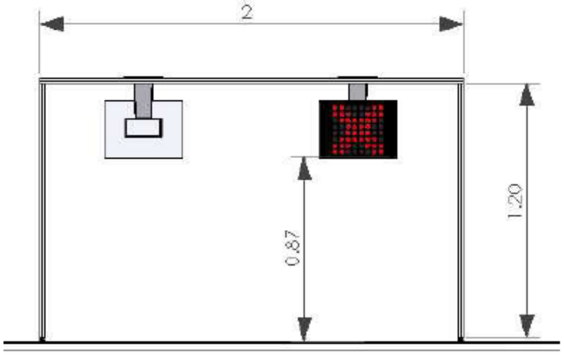

Fig. 3. Signalling Panels (Obtained from [3])

The symbols presented in the signalling panels have 600x600 pixels and are shown on the 1024x768 pixels resolution of the signalling panel. Figure 4 shows the used symbols named hereafter as specific traffic signs (STS). The associated functions to each symbol are: Stop (red X), Follow to the left (left arrow), Follow straight ahead (up arrow), Follow to Parking Area (right arrow) and End the Trial (coloured checkers), respectively.

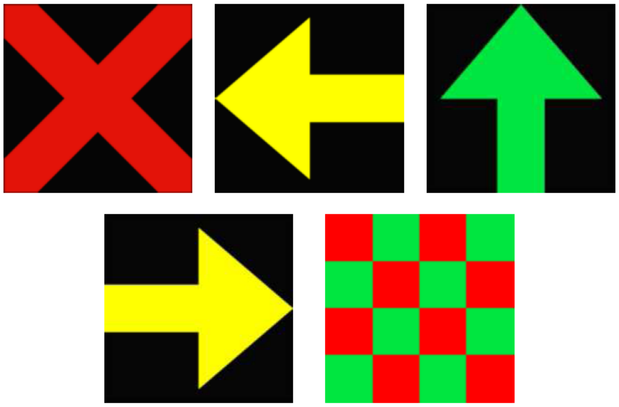

Fig. 4. Symbols used on signalling panels

The Follow to Parking Area symbol indicates the robot should go for a place on the right side, and this place has a distance varying between 50 cm and 200 cm from the signalling panel, marked with a **P** as shown on Figure 2 or with more details on Figure 5.

Since the track is available only during the competition and tests are necessary before the competition, a similar environment was created in the laboratory in order to imitate the real environment. A Cathode Ray Tube (CRT) monitor was used instead a computer flat screen. In order to imitate the road a gray carpet with white lines is used. This all scenario is more demanding to the algorithms than the real scenario and hence a good test bed for the algorithms. Figure 6 shows the similar created environment.

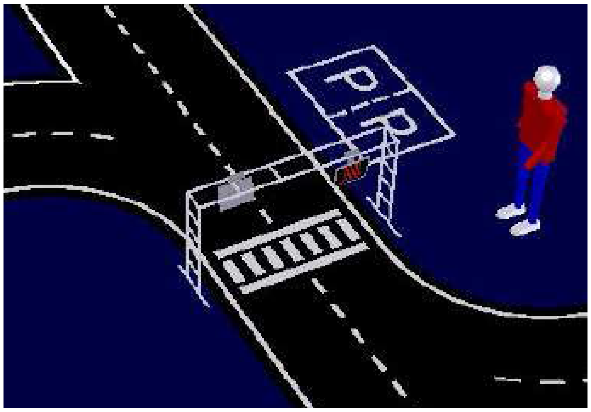

Fig. 5. Parking Area (Obtained from [3])

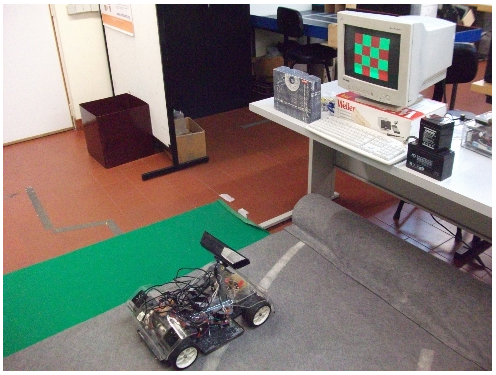

Fig. 6. Tests environment in the laboratory

Although there are visible differences between the flat panel and the one used during the tests (CRT), for a computer camera such as the Kinect there are some more visible imperfections in the captured image quality on the latter monitor including light reflections due to the CRT screen glass, image brightness and symbol quality. A working algorithm in the laboratory is an assurance of its feasibility during the competition. The symbols used in this work are rotated 180° from the original since in the laboratory environment the monitor is not in an inverted position. That does not change the algorithms nor the outcomes of them.

IV. Results

This section shows the obtained results with the Kinect sensor. The main task is to recognise the STS, however, in order to submit one region to the classifier some steps are required. First, one filter using the depth image from the Kinect is applied. Second, morphological operators are used on the segmentation task. Finally, the classifier, an Artificial Neural Networks is applied to recognise the STS.

The system was programmed in the C++ language and using the OpenCV library [23]. To access the information from the Kinect sensor a framework named OpenFrameworks [24] was used, more specifically OfxKinect. The code was developed using an Integrated Development Environment (IDE) named Code Blocks and it was all developed on the Linux platform using Ubuntu 9.10 distribution.

A. Filter

The Kinect has two integrated sensors as described before, namely vision and distance. A useful approach for this sensor is to combine these informations, for example, to use the information of one sensor together with the information from the other sensor. In other words, by using the grayscale image information from the depth image and applying it to the image from the RGB camera, near or far objects can be depicted and isolated. In the depth grayscale image, white objects represents short or near distances whereas black objects are at a far distance.

Figure 7 shows an image captured from the RGB camera of the Kinect, and Figure 8 shows this image as a depth grayscale image as described above.

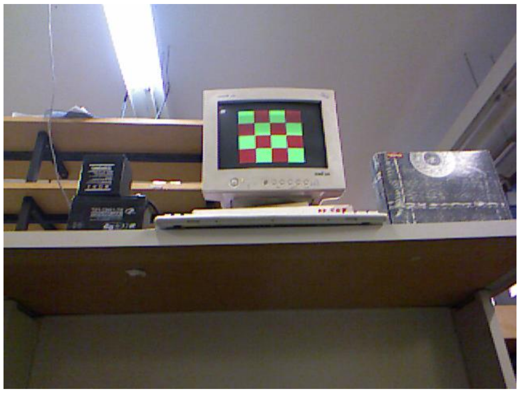

Fig. 7. Image captured from the RGB Camera of Kinect

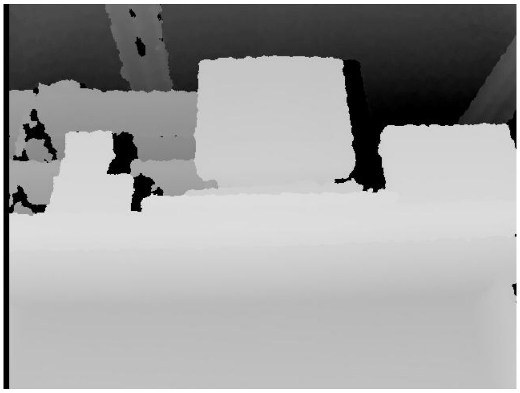

Fig. 8. Depth grayscale image

The depth image generated can be used to eliminate objects that are near or far, and hence it can reduce the computational costs of the next stage by eliminating some undesired regions presented on the image.

An approach to build this filter is to apply a threshold function to the depth image. This function obtains objects which have a required distance. This idea is the same of a *lowpass* and *highpass* filters. The results of this approach on Figure 8 can be seen on figures 9 and 10. Figure 9 is the result with the threshold function eliminating all objects that have pixel values over 210, which are the near objects, while on Figure 10 the threshold eliminates all pixels with values below 190, i.e., far objects.

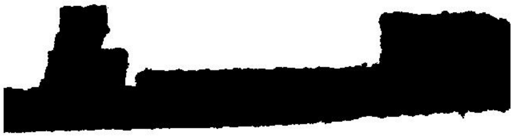

Fig. 9. Near Image - Threshold value: 210

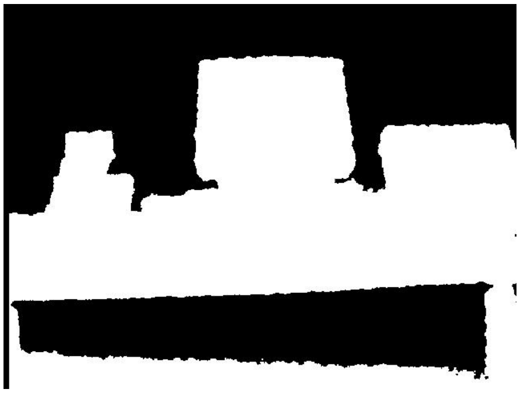

Fig. 10. Far Image - Threshold value: 190

The obtained results from Figure 9 demonstrate that the near objects are eliminated, while the desired object, in this case the CRT monitor, and some far objects are maintained on image. Meanwhile, on Figure 10 the elimination occurs for far objects. In this case they are farther than the CRT monitor, being behind the CRT and hence removed from the image.

One can see on Figure 9 the batteries (black objects on the left side of the CRT monitor on the RGB image of Figure 7), box on the right side of the CRT monitor, keyboard below the CRT monitor and the parts of the table are eliminated (black pixels on Figure 9). As well as on Figure 10 one can perceive the elimination of part of the table below the CRT monitor and objects that are farther than the CRT monitor.

The near and far values depend on the problem to solve. One advantage of this approach on autonomous driving is that the robot can detect the signalling panels with a certain distance value. Thus, after discovering this value only some adjustments are necessary.

The proposed approach minimizes the processing time, however, it is necessary to combine these images, near and far, in order to obtain the CRT monitor. The desired region can be obtained applying an **And** operator on Figures 9 and 10, resulting on an image containing elements with a specific region. It contains only the objects that are not very near and far objects, in other words, the desired object. Figure 11 shows the results from the **And** operator, while on Figure 12 sumarises all the described process.

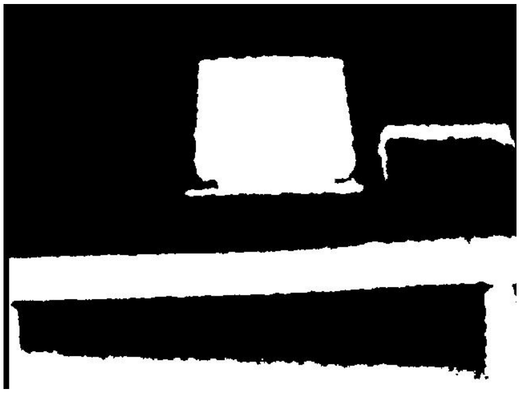

Fig. 11. Intersection Image

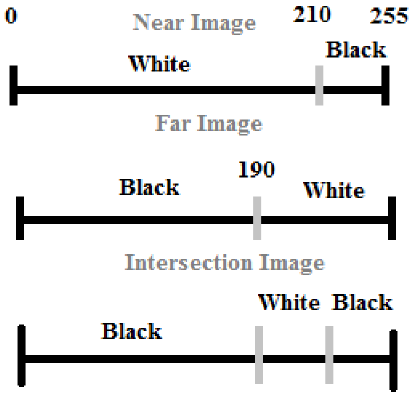

Fig. 12. Summary of the whole process

B. Segmentation

Since a filter is applied to the image, it reduces the processing time to obtain the region of interest (ROI), however, a segmentation is still required. The results from the filter, last subsection, demonstrates that it is still necessary to separate the elements.

Among many techniques that can be used to separate the elements and obtain the ROI, the morphological operators were chosen for this task on this work. Only an **Opening** operation from the morphological operators is enough. Instead of using the Opening operator it is also possible to apply the Erode and Dilate operations. Figure 13 shows the results with the Opening operator, while Figure 14 shows the results of Erosion and Dilation operations, both operations are applied using Figure 11 as the input.

It is visibly clear that there is a small difference between the results from Opening or Erode and Dilate operations, however, only one call to the Openning function is enough, whilst several successively calls to the Erode and Dilate functions are necessary. Therefore, for this work the Opening operator was adopted.

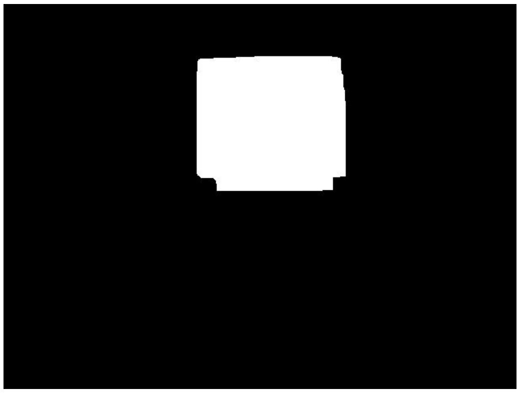

Fig. 13. Opening operation

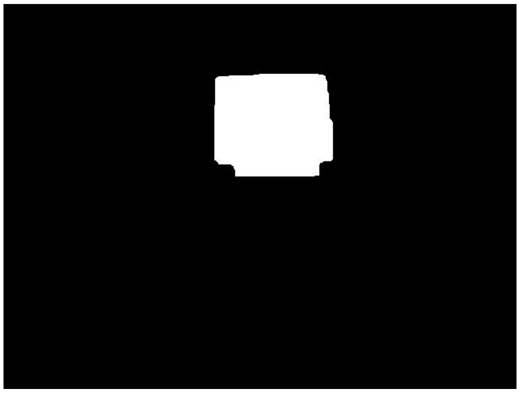

Fig. 14. Erode and Dilate operations

C. Classification

After the filtering and segmentation tasks, the image is divided in ROIs and the classifier is ready to recognise the STS. These ROIs are stored as Blobs.

As stated earlier, classification has been solved by authors by using Artificial Neural Networks (ANN) [29] [19]. Some advantages of this are: a) It can approximate any function [16]; b) it works with noisy data; c) a minimum knowledge about the problem is required [11]; d) a minimum knowledge of the input statistical distribution [5]; e) the ability to generalise for untrained situations.

The robot position affects the blob size, however, the input neurons number should be constant. In order to work with this constraint the resize operation is applied, in this case the blob size is 10x10 pixels.

Some tests were carried out finding the better input for the ANN. An analisys with two colour spaces (RGB and HSV) was applied. For the ANN input the blob values were used. Regarding the output neurons the applied approach was to use each output neuron for a STS. This means that for each STS the responsible neuron will produce a value near 1, while the other neurons should have output near -1. This approach is showed on Figure 15, where the first neuron is responsible for the STOP symbol. On Figure 16 the same approach is applied, however, the symbol is the PARKING symbol and the responsible neuron is the fifth. The sequence used for all the symbols are: Stop (1), Right (2), Front (3), Left (4) and Parking (5).

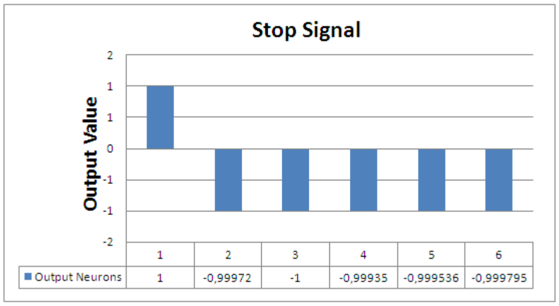

Fig. 15. ANN output for the Stop Symbol

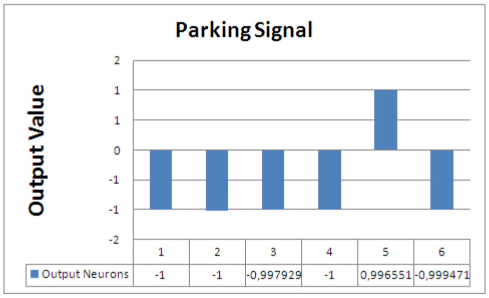

Fig. 16. ANN output for the Parking Symbol

After the tests with the two different colour spaces, other parameters could be tested such as the hidden layer numbers parameter. Table I shows the results obtained from training. The same sample set was used but different configurations were tested in order to do a comparison between them. The output quadratic error is then measured to be analysed. The instances applied are: With one (1HL) and two (2HL) hidden layers, RGB and HSV colour spaces. The comparison shows that the best result is obtained with the HSV colour space and two hidden layers, while the worst is obtained with a RGB and one hidden layer. The outliers on the worst situation are due to, in some cases, a mismatching in two neurons giving similar outputs.

TABLE I
DIFFERENT INPUTS FOR THE NEURAL NETWORK

	Sum	Average	Maximum	Minimum
HSV-2HL	0.024086	0.0000167847	0.001346	0
HSV-1HL	0.694077	0.000481998	0.035041	0
RGB-2HL	0.263052	0.000182675	0.009617	0
RGB-1HL	569.428941	0.395436765	4.213297	0

Other relevant measure taken into account is the computational cost and since this is to be applied in a real time application it should be taken into account. The time spent for each scenario is:

1) HSV-2HL: 0.042 seconds
2) HSV-1HL: 0.040 seconds

3) RGB-2HL: 0.039 seconds
4) RGB-1HL: 0.037 seconds

After the best parameters are found the chosen ANN is applied. The differences between the computational costs are not significant, and the time taken to convert one RGB blob into HSV blob only requires $2.6E-5$ seconds, thus the better scenario is the HSV with 2 hidden layers. Table IV-C shows all the parameters used with the ANN.

TABLE II
PARAMETERS

Parameters	Value
Input Layer	300
Hidden Layer 1	20
Hidden Layer 2	15
Output Layer	6
Learning rate	0.001
Momentum term	0.002
Desired Error	0.00007
Examples Number	240
Epochs	99999

The input size for the ANN is the blob size multiplied by three (each channel of the colour space). This approach is the same for RBG and HSV colour spaces. The input neurons on Table are 300, 10x10x3 (blob size 10x10 times 3 channels).

The output neurons size is increased by one unit in relation to the symbol number. The five top are the symbols and the last neuron represents objects that are different from the symbols. This last neuron is used because in some situations, depending on the robot's position, some obtained blobs do not represent the signalling panels. Figure 17 shows the above described situation; there are two blobs, the first represents the signalling panels while the second is a box where the monitor is above. This situation happens within the laboratory environment since the monitor should have a specific height and has to be standing over a box. However, on competition this does not occur because the signalling panel is held by the portico structure (see Figure 3). Therefore, a simple Opening operation can easily remove this structure.

The order adopted on each figure starting on Figure 17 is: a) top side with left image as the image captured with the RGB camera and the right image with the blob to be classified (red rectangles); b) bottom side with a set of images positioned being the first (leftmost) image the depth map, the second image is the near filter, the third image is the far filter, the fourth image is the intersection filter and finally the fifth and rightmot image is the result using the Openning operation. Figure 17 shows the near image filter in the fore plane with the chair removed, and in the far image filter many elements on the environment were also removed. The intersection image only contains the box and the monitor thus with the morphological operators they can be separated.

In order to obtain good outcomes the robot was placed at different distances from the CRT monitor, producing different horizontal angles of the camera in relation to the CRT monitor. Figure 18 shows a test with the robot close to the table, while on Figure 19 the robot is at an horizontal angle with the monitor. The bottom images on Figures 18 and 19 have the same notation from Figure 17 while the top images shows the original image (left) and the used blob (right) during the training (red rectangle), respectively.

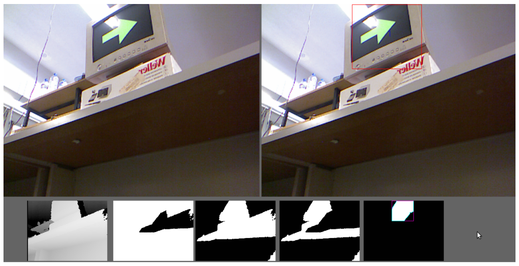

Fig. 18. Training: Robot is close to the table

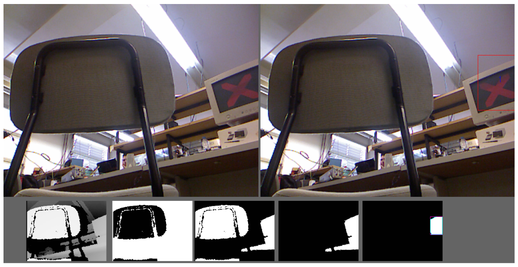

Fig. 19. Training: Robot is at a horizontal angle with the CRT monitor

Figures 20 and 21 shows the obtained results from the classifier. Figure 20 shows the robot at a horizontal angle nevertheless the stop symbol is recognized and on Figure 21 it recognises the parking symbol at a far distance.

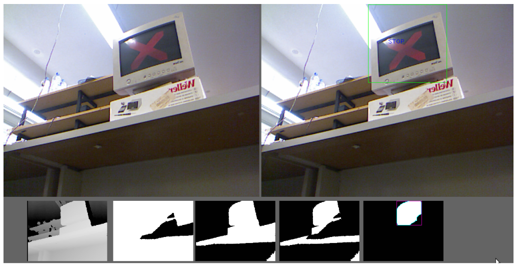

Fig. 20. Recognition: Robot is at an angle

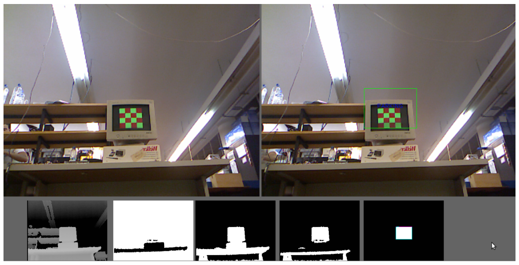

Fig. 21. Recognition: Robot is far

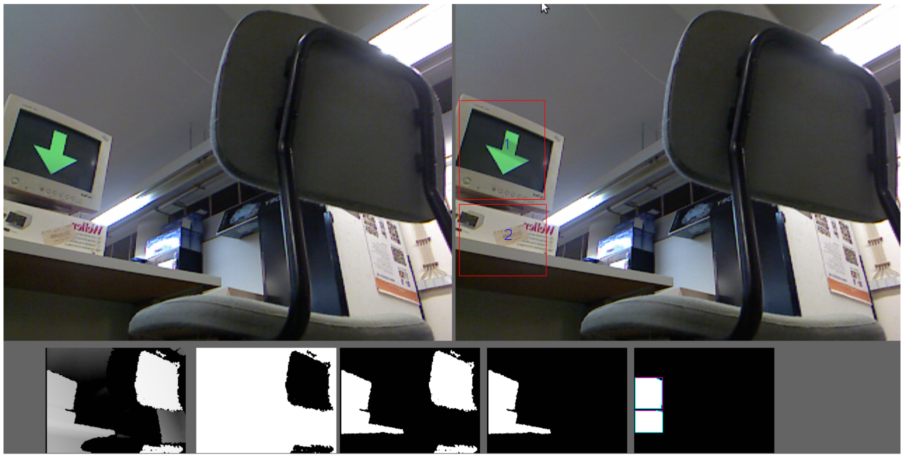

Fig. 17. Istance with two Blobs

V. Discussion

The Kinect sensor provides a major reduction on computational cost, since the filter eliminates undesired elements present on the environment. Other advantage of the depth sensor is the possibility to calculate the real distance in centimeters of each pixel. When a Blob is detected, the distance between the object and the camera can be determined. This distance can be an interesting feature for the car control task.

Some approaches could be used as an input for the ANN. One approach to reduce the input neurons size is to use some statistical rates, such as mean and standard deviation. In [19], instead of using an input of 2700 neurons (blob size of 30x30 pixels times the three channels R, G and B), it uses as an input three normalized average maximum pixel values (MR being Maximum Red, MG being Maximum Green and MB being Maximum Blue); 30 inputs from a vertical histogram (vh) and 30 inputs from a horizontal histogram (hh). Other approach could be to use some information from blobs, for example a centroid, an area or other property.

In fact, the last neuron can be removed. For elements not trained all the outputs should be near -1. This approach was not used due to the unstructured environment in the laboratory having objects similar to a STS. However, on competition with a more structured environment this neuron can be removed.

Since this is a real time application the time spent for each operation is a main point. The required time to filter and segment an image is 0.015 seconds, while the classification takes 0.042 seconds. Therefore, the total time is 0.057 seconds. The obtained computational cost is very short and the main point for this reduction is the filter, which is done with a fusion of the distance and the vision information.

An important feature not discussed yet is the Kinect ability to operate in the dark. Since the camera uses infrared light the depth image is the same on a dark or illuminated environment. This property can be useful for other applications such as when the robot is inside the tunnel in the competition.

VI. Conclusion

This paper demonstrates a practical application of the Kinect camera, developed by Microsoft as an interface for the game industry, but here used as a sensor. It has two integrated sensors, vision and distance, and the presented approach was to shown the gain obtained by combining these two sensors.

The application field is on an autonomous and mobile robot developed to compete on an autonomous competition, namely the autonomous driving league at the Portuguese Robotics Open. The Kinect sensor was useful and the developed filter reduces the required processing. Therefore, the distance information assists the vision information.

The obtained results from the filter simplifies the next steps, thus by using solely an Opening operation from the morphology it is possible to obtain the ROI. The classifier used was an ANN, the pixel values from blobs are used as an input and an output is generated where each neuron of the MLP network is responsible for one STS. This works also presented a comparison between the colour spaces RGB and HSV applied to a MLP with one and two hidden layers. The best result was with a HSV and two hidden layers. It was observed that the computational cost to convert RGB into HSV and two hidden layers instead of one, was not significant.

For further work, the pixel removing of the outer white areas of the computer screen (screen holder) would be necessary, in order to obtain only the black region that contains the STS. This approach should produce better results than the current approach.

Acknowledgment

The authors wish to thank all members of the Laboratório de Automação e Robótica, at University of Minho, Guimarães. Also special thanks to the ALGORITMI Research Centre for the opportunity to develop this research.

References

[1] Microsoft kinect. http://www.xbox.com/kinect [Accessed in: 15 Oct. 2011].

[2] Portuguese robotics open (festival nacional de robotica), 2011. http://robotica2011.ist.utl.pt/ [Accessed in: 28 Sep. 2011].

[3] Portuguese robotics open: Rules for autonomous driving league, 2011. http://robotica2011.ist.utl.pt/docs/2011_Conducao_Autonoma-regras-en.pdf [Accessed in: 28 Sep. 2011].

[4] Alempijevic, A., Carnian, S., Egan-Wyer, D., Dissanayake, G., Fitch, R., Hengst, B., Hordern, D., Kirchner, N., Koob, M., Pagnucco, M., Sammut, C., and Virgona, A. Robotassist - robocup@home 2011 team description paper: Robocup, 2011.

[5] Amoroso, C., Chella, A., Morreale, V., and Storniolo, P. A segmentation system for soccer robot based on neural networks. In *RoboCup-99: Robot Soccer World Cup III* (London, UK, 2000), Springer-Verlag, pp. 136–147.

[6] Benavidez, P., and Jamshidi, M. Mobile robot navigation and target tracking system. In *System of Systems Engineering (SoSE), 2011 6th International Conference on* (june 2011), pp. 299 –304.

[7] bin Mohd Sidik, M., bin Sunar, M., bin Ismail, I., bin Mokhtar, M., and Jusoh, N. A study on natural interaction for human body motion using depth image data. In *Digital Media and Digital Content Management (DMDCM), 2011 Workshop on* (may 2011), pp. 97 –102.

[8] Chacon, J. D., van Elteren, T., Hickendor, B., van Hoof, H., Jansen, E., knuijver, S., Lier, C., Neculoiu, P., Nolte, A., Oost, C., Richthammer, V., Schimbinschi, F., Schutten, M., Shantia, A., Snijders, R., van der Wal, E., and van der Zant, D. T. Borg - the robocup@home team of the university of groningen team description paper: Robocup, 2011.

[9] DARPA. Grand challenge. http://www.darpa.mil/ [Accessed in: 13 Oct. 2011].

[10] de la Escalera, A., Armingol, J., and Mata, M. Traffic sign recognition and analysis for intelligent vehicles. *Image and Vision Computing 21*, 3 (2003), 247 – 258.

[11] Egmont-Petersen, M., de Ridder, D., and Handels, H. Image processing with neural networks - a review. *Image processing with neural networks - a review. Pattern Recognition 35*, 10 (2002), 2279–2301.

[12] Fiala, M., and Ufkes, A. Visual odometry using 3-dimensional video input. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on* (may 2011), pp. 86 –93.

[13] Fleyeh, H. Traffic sign recognition by fuzzy sets. In *Intelligent Vehicles Symposium, 2008 IEEE* (june 2008), pp. 422 –427.

[14] Gallo, L., Placitelli, A., and Ciampi, M. Controller-free exploration of medical image data: Experiencing the kinect. In *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on* (june 2011), pp. 1 –6.

[15] Gomez-Moreno, H., Maldonado-Bascon, S., Gil-Jimenez, P., and Lafuente-Arroyo, S. Goal evaluation of segmentation algorithms for traffic sign recognition. *Intelligent Transportation Systems, IEEE Transactions on 11*, 4 (dec. 2010), 917 –930.

[16] Haykin, S. *Neural Networks: A Comprehensive Foundation*, 2 ed. Prentice Hall, 1999.

[17] Khandelwal, P., and Stone, P. A low cost ground truth detection system using the kinect. In *Proceedings of the RoboCup International Symposium 2011 (RoboCup 2011)* (July 2011).

[18] Liu, H., Liu, D., and Xin, J. Real-time recognition of road traffic sign in motion image based on genetic algorithm. In *Proceedings of the First International Conference on Machine Learning and Cybernetics* (November 2002), vol. 1, pp. 83 – 86.

[19] Lorsakul, A., and Suthakorn, J. Traffic sign recognition for intelligent vehicle/driver assistance system using neural network on opencv. In *The 4th International Conference on Ubiquitous Robots and Ambient Intelligence URAI 2007* (2007), pp. 279–284.

[20] Maldonado-Bascon, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gomez-Moreno, H., and Lopez-Ferreras, F. Road-sign detection and recognition based on support vector machines. *Intelligent Transportation Systems, IEEE Transactions on 8*, 2 (june 2007), 264 –278.

[21] Matuszek, C., Mayton, B., Aimi, R., Deisenroth, M. P., Bo, L., Chu, R., Kung, M., LeGrand, L., Smith, J. R., and Fox, D. Gambit: An autonomous chess-playing robotic system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (may 2011), pp. 4291 –4297.

[22] Matyunin, S., Vatolin, D., Berdnikov, Y., and Smirnov, M. Temporal filtering for depth maps generated by kinect depth camera. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011* (may 2011), pp. 1 –4.

[23] OpenCV:, 2011. http://opencv.willowgarage.com/wiki/ [Accessed in: 10 Jan. 2011].

[24] OpenFrameworks, 2011. http://www.openframeworks.cc/ [Accessed in: 19 Jun. 2011].

[25] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (june 2011), pp. 1297 –1304.

[26] Siegwart, R., and Nourbakhsh, I. R. *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.

[27] Stowers, J., Hayes, M., and Bainbridge-Smith, A. Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor. In *Mechatronics (ICM), 2011 IEEE International Conference on* (april 2011), pp. 358 –362.

[28] Van den Bergh, M., Carton, D., De Nijs, R., Mitsou, N., Landsiedel, C., Kuehnlenz, K., Wollherr, D., Van Gool, L., and Buss, M. Real-time 3d hand gesture interaction with a robot for understanding directions from humans. In *RO-MAN, 2011 IEEE* (31 2011-aug. 3 2011), pp. 357 –362.

[29] Vicen-Bueno, R., Gil-Pita, R., Rosa-Zurera, M., Utrilla-Manso, M., and Lopez-Ferreras, F. Multilayer perceptrons applied to traffic sign recognition tasks. In *Computational Intelligence and Bioinspired Systems*, vol. 3512 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005, pp. 865–872.

[30] Xia, L., Chen, C.-C., and Aggarwal, J. Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on* (june 2011), pp. 15 –22.