CameraDriver
Vision System for Autonomous Driving

Renato Silva and João Paulo Costeira

***Abstract*—The main purpose of this work is to develop a localization system based on computer vision to courses that are delimited by objects. This problem is based on the Formula Student Driverless competition on which a vehicle without driver must be able to locate on a cone delimited track.**

The developed system uses a 3D camera, which allows to calculate distances. The objects, in this case cones, are detected in the images through the YOLO system. The distances between cones are collected, and except for degenerated cases, the collection of every distances forms a unique map "signature".

The possibility of a prior recognition of the course allows mapping using the ORB-SLAM2 system. Through its post-processing, the global map of the course is constructed, which includes the localization of the cones and the distances signatures.

To estimate the course localization of the camera at every moment, the construction of the local map is based on the observed objects. The matching between the observed objects and the ones that are along the course is done using a search algorithm, which matches the local map with the global map, based on the distances signatures. Once they are matched, the localization and orientation of the camera is estimated.

The system was tested using synthetic and real courses which led to very promising results.

***Index Terms*—Object Detection on Image, SLAM, Greedy Algorithms, Graphs, Computer Vision.**

I. Introduction

THE Formula Student Driverless is an international university competition in the field of motor sports, in which it is intended to develop a full-scale competition vehicle capable of moving autonomously, and without driver, in tests where the courses are delimited by cones of different colors and sizes.

The possibility of carrying out a recognition walk to the course of the test, allows its mapping to be carried out. The collected information can be used to optimize the location system of the vehicle.

In this work, we intend to estimate the location of the 3-D camera along the couse, based on the position of specific objects. Through the mapping of the course made in the previous recognition and the location of the detected cones that delimit it, the global map of the course is constructed. This map corresponds to the collection of all distances between cones, which constitutes a unique "signature" for each map, except for degenerated cases.

For each instance, the cones are detected in the image and their location is obtained, allowing to construct a local map. The search of correspondence between the local map and the global map, based on the distance between objects, give us the object pairing, allowing us to obtain the localization and orientation of the camera.

II. Proposed Solution

The proposed solution for the problem of localization is capable of:

- Detecting objects on image;
- Mapping the course;
- Building a Global Map of the course;
- Building a Local Map of each frame;
- Corresponding objects between Local Map to Global Map;
- Obtaining the camera pose.

A. Object Detection

The YOLO[1][2] is used to perform the identification and location of objects in image. This system uses a single convolutional neural network to perform the identification and location of objects in image. Receives as input an image on which we want to detect and locate objects. This image is resized to a specific resolution, and divided into a network $S \times S$ cells, where each one is responsible for estimating B bounding boxes.

The image is processed by a Convoluted Neuronal Network trained for the detection of specific objects. After this processing, we obtain the respective bounding boxes defined by their deviation and size, (t_x, t_y, t_w, t_h), the probability of the object contained therein for each class $(p_1, p_2, ..., p_C)$, and its confidence score (p_o). This score reflects how secure the system is that the bounding box contains an object. The bounding box is defined by:

$$\text{Bouding Box} = (t_x, t_y, t_w, t_h, p_o, p_1, p_2, ..., p_C) \quad (1)$$

It is held the suppression of all resulting detections that have a confidence score below a certain threshold. The detections, with scores above the defined threshold, are subsequently used in the construction of the maps.

B. vSLAM

The simultaneous mapping and localization system used is the ORB-SLAM2[3], capable of processing information from stereo or RGB-D cameras.

This vSLAM algorithm presents an architecture divided into four threads and each one of those are responsible for performing a specific part of the system processing, which are:

- ***Tracking***: It tracks the location of the camera in each frame using the combination of features of the current

frame with the local map. The reprojection error is minimized by the application of *Motion-only BA*.
- ***Local Mapping***: Manages the Local Map, and optimizes it by performing *Local BA*.
- ***Loop Closing***: Detects large loops and corrects the accumulated offset by optimizing the position graph.
- ***Full BA***: Bundle Adjustment is applied to the entire map, obtaining an optimal solution of the structure and motion of the position graph.

In order to be able to support data from RGB-D and stereo cameras, this system performs a preprocessing of the acquired data, obtaining *Stereo Keypoints* and *Monocular Keypoint*.

In the case of stereo camera, ORB[4] features (Oriented FAST and Rotated BRIEF) are extracted from the two RGB images, and searched for a match between ORB features of the left image with the right image. If there is correspondence, they are considered *Stereo Keypoints* that are defined by 3 coordinates $x_{Stereo} = (u_L, v_L, u_R)$ where (u_L, v_L) are the coordinates of the ORB feature in the left image, and (u_R) is the horizontal coordinate of the same ORB feature in the right image. If there is no match, they are considered *Monocular Keypoint* and are defined by the ORB features coordinates in the left image $x_{Mono} = (u_L, v_L)$.

In the case of RGB-D camera, ORB features are acquired from the RGB image, and associated a depth value to them. If the depth value is valid, it is considered a *Stereo Keypoint*.

In order to maintain the coherence of the definition of the coordinates that characterize this type of point, the coordinates (u_L, v_L) correspond to the coordinates of the ORB feature in the RGB image, and the coordinate (u_R) corresponds to the horizontal coordinate of this feature in a virtual right image, obtained through the equation 2. In this, d corresponds to the depth value associated with the coordinates of the ORB feature in the RGB image, f_x corresponds to the horizontal focal length and b corresponds to the distance between the optical centers of the projector and the IR camera. If the depth value is not valid, it is considered a *Monocular Keypoint*.

$$u_R = u_L - \frac{f_x b}{d} \tag{2}$$

In case of *Stereo Keypoints*, these can still be cataloged as near or distant, taking into account their depth. If the depth value is less than forty times the value of the distance between the optical centers, they are considered *Close Stereo Keypoints*, which are responsible for obtaining a good estimate of depth, scale, translation and rotation. If the depth value is higher, they are considered *Far Stereo Keypoints*, which allow to obtain a good estimate of rotation.

The ORB-SLAM2 uses *Bundle Adjustment*[5], by the Levenberg-Marquardt method, in three circumstances, which are the optimization of the orientation and position of the camera (*Motion-only BA*), the optimization of a subframe of *Keyframes* and their *Keypoints* observable in a local window (*Local BA*), and the optimization of all keyframes and *Keypoints* present on the map after a loop has been closed (*Full BA*).

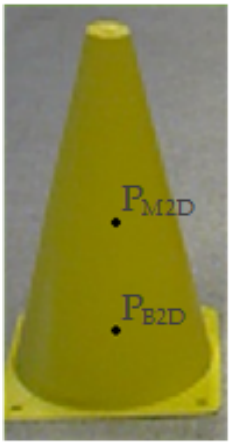


Fig. 1. Important points to determinate the localization of the cones.

For each sequence processed by ORB-SLAM2, translational vectors and rotation matrices are obtained for all frames present in the sequence as well as for Keyframes.

C. Global Map Construction

In order to construct a global map, the mapping of the course is realized. After processing the sequence with the ORB-SLAM2 and the YOLO, we obtain the rotation matrices R and the translation vectors T for each frame and the bounding boxes where the desired objects were detected. Through this information, the global map of the scenario is constructed, which corresponds to a representation in the form of a complete graph, where each object corresponds to a vertex and the weight of the edge connecting two vertices is the distance between the two objects.

Assuming that the bounding box defined by the coordinates in the RGB image $(u_{left}, v_{top}, u_{right}, v_{bottom})$ is able to correctly contain the object within it, two 2- D points of the detections are considered important, which are:

$$P_{M_{2D}} = \left(\frac{u_{right}+u_{left}}{2}, \frac{v_{bottom}+v_{top}}{2}\right) \tag{3}$$

$$P_{B_{2D}} = \left(\frac{u_{right}+u_{left}}{2}, \frac{3v_{bottom}+v_{top}}{4}\right) \tag{4}$$

From the depth image, we can get the 3-D points $P_{M_{3D}}$ e $P_{B_{3D}}$ of the 2-D points $P_{M_{2D}}$ e $P_{B_{2D}}$ through the equation 6, where f_x and f_y correspond to the focal length in x and y respectively, C_x e C_y correspond to the major point offset, (u, v) corresponds to the vertical and horizontal coordinates of the 2-D point in the RGB image and d corresponds to Z. These two 3-D points correspond to the location of a cone detected in the camera referential.

$$\begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} ud \\ vd \\ d \end{bmatrix} \tag{5}$$

$$X = \frac{(u - C_x)d}{f_x}$$
$$Y = \frac{(v - C_y)d}{f_y} \quad (6)$$
$$Z = d$$

For each frame, a rigid 3-D transformation corresponding to the rotation and translation obtained by the ORB-SLAM2 is applied to the points $P_{M_{3D}}$ and $P_{B_{3D}}$ of each detection in this frame, obtaining all detection locations in the same referential.

Two point clouds are obtained, one corresponding to points $P_{M_{3D}}$ and the other to points $P_{B_{3D}}$. For each of these point clouds, all locations of the detections are superimposed.

In order to detect the points that define the location of the objects, for each one of the point clouds its segmentation is performed taking into account the Euclidean distances between the points, where considering 2 points $a, b \in \mathbb{R}^3$, if the Euclidean distance between $d_{ab} = ||a-b||_2 = \sqrt{\sum_i (a_i - b_i)^2}$ is less than a threshold τ_d, it is assumed that a and b belong to the same cluster, that is, they define the location of the same object. The location of the k object on the map, P_k, is determined by the average of the points contained in the cluster C_k:

$$P_k = \frac{1}{N}\left(\sum_{i=1}^{N} P_i\right), \qquad P_i, P_k \in \mathbb{R}^3, \qquad P_i \in C_k \quad (7)$$

Through the locations of the cones associated with the points $P_{M_{3D}}$, an upper triangular matrix of distances M_{Map_G} is created among the various objects, the objects coordinates in the map $P_{Map_{GM}}$ and its labels $Labels_{Map_G}$, which define the type of object detected. Through the locations of the Cones associated with the points $P_{B_{3D}}$, only the objects coordinates in the map $P_{Map_{GB}}$.

D. Local Map Construction

The construction of the Local Map resembles to the construction of the Global Map. For each of the frames contained in the test sequence, the detections performed by YOLO are obtained, and therefore obtain the points corresponding to the location of the detected objects, $P_{M_{3D}}$ and $P_{B_{3D}}$. Unlike the construction of the Global Map, where a process of overlapping the various locations would subsequently take place, the construction of the Local Map is done from the location of the objects obtained in a frame, related to the camera referential.

Through the locations of the cones associated with the points $P_{M_{3D}}$, we obtain an upper triangular matrix of distances M_{Map_L} between the various objects in the frame, the coordinates of the objects in the local map of each $P_{Map_{LM}}$ and its respective labels $Labels_{Map_L}$, which define the type of object detected.

The location coordinates of the cones associated with the points $P_{B_{3D}}$ are listed are listed in $P_{Map_{LB}}$.

E. Solution Search

The correspondence between the Global Map and the Local Map is taken into consideration with the distances between objects of the Global Map, M_{Map_G}, and Local Map, M_{Map_L}, as well as their labels, $Labels_{Map_G}$ and $Labels_{Map_L}$.

This problem correspond to the search for a subgraph of a complete graph, that is, search the resulting graph of the Local Map in the resulting graph of the Global Map. These types of problems are classified as NP-Complete[6]. Given the complexity of this type of problem, a Greedy algorithm was developed capable of performing a Local Map search on the Global Map.

In these graphs, each vertex represents an object and each of these vertices has edges that connects it directly to all other vertices. Each edge has an associated weight, corresponding to the distance between objects that the connected vertices represent. Each vertex is also associated with the type of object to which it corresponds. This type of object may not be unique in the graph, since, for example, there may be multiple cones of the same color and size in the course delimitation. 4

F. Estimate Camera Pose

After matching the location of the local map objects on the global map, we want to know the location and orientation in which the camera is located.

This problem corresponds to the Orthogonal Procrustes [7], whose objective is to find the Rotation $R \in SO(3)$ and the Translation $T \in \mathbb{R}^3$ given the points $B_i \in \mathbb{R}^3$, which corresponds to the locations of the objects in the local map, $[P_{Map_{LM}}; P_{Map_{LB}}]$, and the points $A_i \in \mathbb{R}^3$, corresponding to the location of the matched objects in the global map,$[P_{Map_{GM}}; P_{Map_{GB}}]$, satisfying the following equation:

$$A_i = RB_i + T \quad (8)$$

To obtain R and T, the following least-squares problem is solved:

$$min_{R,T} \sum_{i}^{N} ||A_i - RB_i - T||^2 \quad (9)$$

In order to T, we observe that the translation corresponds to the difference between the centroids:

$$T = \frac{1}{N}\sum_{i}^{N} A_i - R\frac{1}{N}\sum_{i}^{N} B_i = \bar{A} - R\bar{B} \quad (10)$$

The problem can be rewritten in the form:

$$min_R ||\tilde{A} - R\tilde{B}||_F^2 \quad (11)$$

Where $\tilde{A}$ and $\tilde{B}$ correspond to:

$$\tilde{A} = A - \bar{A} \quad (12)$$
$$\tilde{B} = B - \bar{B} \quad (13)$$

The solution to this problem is known, corresponding to the singularity decomposition of the matrix $M = \tilde{B}\tilde{A}^T$, which corresponds to:

Course	Green Cones	Yellow Cones	Camera locations
Campera	241	220	66
KIP	294	256	95

TABLE I
NUMBER OF CONES DELIMIT THE COURSES AND CAMERA LOCATIONS

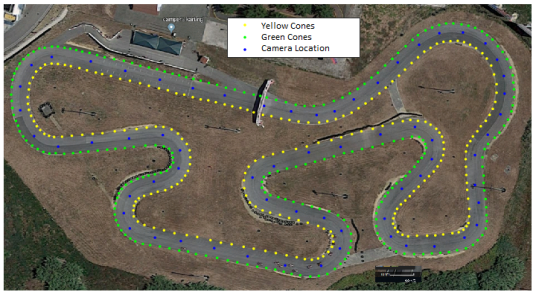
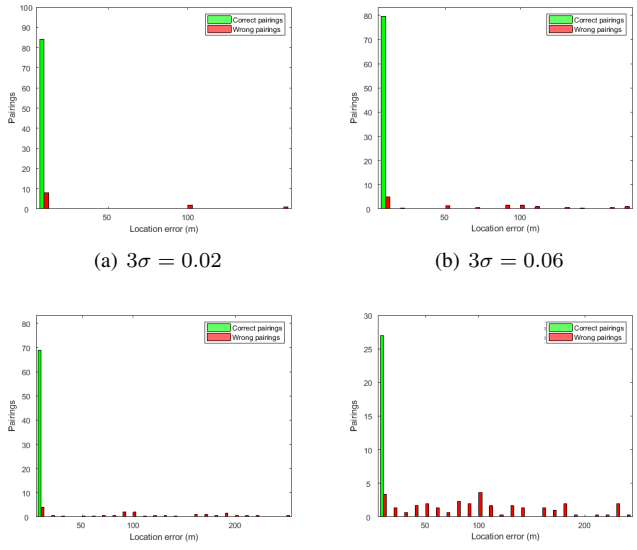


Fig. 2. Campera Karting course delimited by yellow and green cones. The blue dotes represent the locations of the camera.



(a) $3\sigma = 0.02$ (b) $3\sigma = 0.06$

(c) $3\sigma = 0.10$ (d) $3\sigma = 0.20$

Fig. 3. Location error in the KIP course by observing only 3 objects. The green and red bars represent respectively the correct and wrong pairing.

$$M = U\Sigma V^T \tag{14}$$

From this decomposition, we obtain the solution of the rotation matrix R for this problem:

$$R = U\Sigma' V^T \tag{15}$$

Where Σ' corresponds to:

$$\Sigma' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{bmatrix} \qquad d = det(UV^T) \tag{16}$$

Once the rotation matrix R is calculated, the equation 10 is solved, obtaining the vector T. These matrices correspond to the orientation and translation of the camera in the Global Map referential

III. RESULTS

Multiple experiments are performed on the search algorithm using synthetic courses, allowing the evaluation of the matching capability of local map objects, subject to different noise values, with the global map, as well as location and orientation error. The trained YOLO is evaluated according to the detection capability of the cones. The construction of the global map is tested in the section to assess its consistency with the scenario.

The developed system is tested using two different scenarios.

A. Solution Search

To test the operation of the search algorithm, two synthetic tracks were constructed corresponding to the tracks of the Campera Karting and the International Karting Palmela (KIP).

The table I shows the number of cones that delimit the two courses and localizations of the camera. On figure 2, is represented the delimited Campera Karting course delimited.

For both paths, tests are performed with different numbers of observable objects at each camera location. The chosen ones are the ones closest to the camera, and their location is disturbed by Gaussian noise for different values of standard deviation.

The results for $Max_{Error} = 0.10m$ are presented in the table II. Each test was performed three times, and presented their average.

This results demonstrate that the developed search algorithm allows to obtain a good pairing in situations where there is a small variation of the location of the objects. In these cases, the observation of a greater number of objects allows in the heuristic search to obtain the correct pairing. If the deviation of the object location is very high, it is impossible to ensure that the algorithm is able to perform the pairing correctly.

Obtained the results for the pairing of the objects between the maps, an interpretation of the interference of the various levels of noise in the estimation of the location and orientation of the camera was performed. On figures 3 and 4 are presented the results for the estimation of the location and orientation, respectively.

In cases where the correct pairing occurs, the estimates of the location and orientation of the camera in the route have a low error. On the other hand, a wrong pairing results in a wrong estimation of the location and orientation of the camera.

B. Object Detection

YOLO has been trained to identify green, yellow, blue and orange cones. To assess its detection capacity, it was tested through a sequence containing throughout the images the four types of trained objects.

In the figure 5 two of the sequence images are displayed after detecting objects, with the overlapping of their bounding box and object type.

					Gaussian noise											
		Without noise			$3\sigma = 0.02$			$3\sigma = 0.06$			$3\sigma = 0.10$			$3\sigma = 0.20$		
	#Objects	C	W	N	C	W	N	C	W	N	C	W	N	C	W	N
Campera	3	66	0	0	55	11	0	51	14	1	45	19	3	18	29	19
	4	66	0	0	65	0	1	64	0	2	44	13	9	10	3	53
	5	66	0	0	66	0	0	65	0	1	47	0	19	8	0	58
	6	66	0	0	66	0	0	65	0	1	43	0	23	1	0	65
KIP	3	95	0	0	84	11	0	80	14	1	69	20	6	27	32	36
	4	95	0	0	95	0	0	93	0	2	79	0	16	17	0	78
	5	95	0	0	95	0	0	94	0	1	70	0	25	6	0	89
	6	95	0	0	95	0	0	95	0	0	58	0	37	3	0	92

$Max_{Error} = 0.10m$; C: Correct pairings; W: Wrong pairings; N: Solution not found.

TABLE II

RESULTS OF THE TESTS PERFORMED TO THE SEARCH ALGORITHM

		Gaussian noise														
		$\sigma = 0.01$			$\sigma = 0.02$			$\sigma = 0.03$			$\sigma = 0.04$			$\sigma = 0.05$		
	#Objects	C	W	N	C	W	N	C	W	N	C	W	N	C	W	N
Campera	3	64	2	0	64	2	0	60	5	1	61	4	0	59	7	0
	4	66	0	0	65	0	1	64	0	2	64	0	2	65	0	1
	5	66	0	0	66	0	0	65	0	1	65	0	1	65	0	1
	6	66	0	0	65	0	1	65	0	1	66	0	0	66	0	0
KIP	3	89	6	0	91	4	0	92	3	0	90	5	0	87	7	1
	4	95	0	0	95	0	0	95	0	0	95	0	0	95	0	0
	5	95	0	0	94	0	1	94	0	1	95	0	0	95	0	0
	6	95	0	0	95	0	0	95	0	0	95	0	0	95	0	0

$Max_{Error} = 6\sigma$; C: Correct pairings; W: Wrong pairings; N: Solution not found.

TABLE III

RESULTS OF THE TESTS PERFORMED TO THE SEARCH ALGORITM

Cones	#Detected		#Missed		#Correct		#Wrong	
Green	729	99.5%	4	0.5%	729	100%	0	0%
Yellow	651	88.8%	82	11.2%	649	99.7%	2	0.3%
Blue	732	99.9%	1	0.1%	732	100%	0	0%
Orange	690	94.1%	43	5.9%	678	98.3%	12	1.7%
Total	2802	95.6%	130	4.4%	2788	99.5%	14	0.5%

TABLE IV

YOLO RESULTS FOR THE DETECTION ON THE CONES

The results, presented in table III-B, show that 95.6% of all cones throughout the sequence were detected. All the cones detected in the green and blue colors correspond to their correct color. For yellow and orange cones, only 0.3 % and 1.7 % of the detections, respectively, do not match the corresponding color.

IV. REAL TESTS

In order to test the system in real situations, two scenarios were constructed. The first scenario consists of only four cones, each of different color, where it is intended to verify the operation of the system. In the second scenario, eight cones of two different colors are positioned, in order to create ambiguity with respect to the object label.

A. Scenario 1

In this test we consider a scenario consisting of four cones, each of different color. For the construction of the global map, a sequence consisting of 60 images was used. After its processing, we obtained the estimated location of the cones, represented in the figure 6.

The estimated distances between the various cones are close to the actual scenario, where the mean error of distances corresponds to 3.38cm.

Because we do not have the ground truth of the location and orientation of the camera along the sequence, it is only possible to perform a subjective interpretation of the location and orientation estimated. On figure 7 is represented the estimated localization of the camera along the course.

To evaluate the orientation estimation, on figure 8 is represented three estimated localizations and orientations, taking into account the field of vision of the camera, which corresponds to 61.47º, to visually identify the visible objects and respective position on image.

By the results, we can conclude that the system in this scenario was able to correctly estimate the camera pose in 88.9% of the images where at least 3 objects were detected.

B. Scenario 2

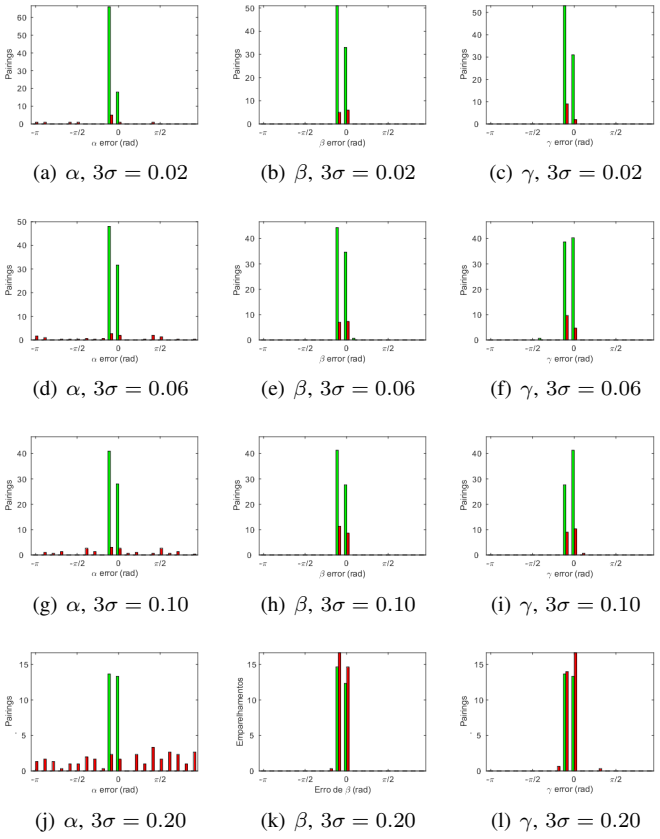
For the second test we consider a scenario consisting of eight cones, four of green and four of blue, in order to create ambiguity with respect to the object label.

For the construction of the global map a sequence consisting of 350 images was used. The estimate of the location of the cones is represented in the figure 9.

The pairing results are presented on table V. The search algorithm was able to match the local maps of 74.8% of the sequence images. The estimation of location and orientation

Cones	#Tests	#Solution found		#Solution Not Found		#Low detections	
Scenario 1	1033	871	84.3%	36	3.5%	126	12.2%
Scenario 2	405	303	74.8%	85	21.0%	17	4.2%

TABLE V
RESULTS FOR THE PAIRING DURING THE 2 SCENARIOS



(a) α, $3\sigma = 0.02$ (b) β, $3\sigma = 0.02$ (c) γ, $3\sigma = 0.02$

(d) α, $3\sigma = 0.06$ (e) β, $3\sigma = 0.06$ (f) γ, $3\sigma = 0.06$

(g) α, $3\sigma = 0.10$ (h) β, $3\sigma = 0.10$ (i) γ, $3\sigma = 0.10$

(j) α, $3\sigma = 0.20$ (k) β, $3\sigma = 0.20$ (l) γ, $3\sigma = 0.20$

Fig. 4. Orientation error in the KIP course by observing only 3 objects

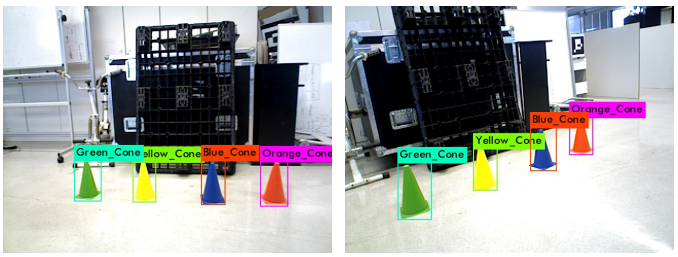


Fig. 5. Sequence images with overlapping of their bounding box and object type estimated by YOLO

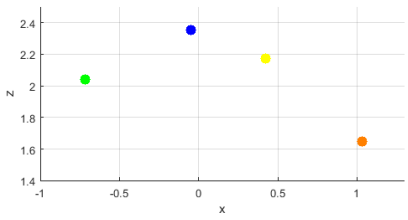


Fig. 6. Estimated location of the cones on scenario 1

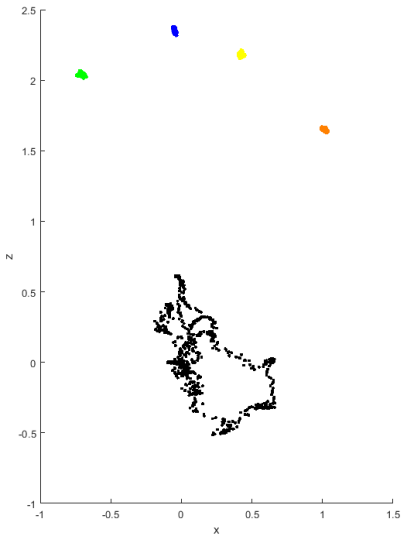
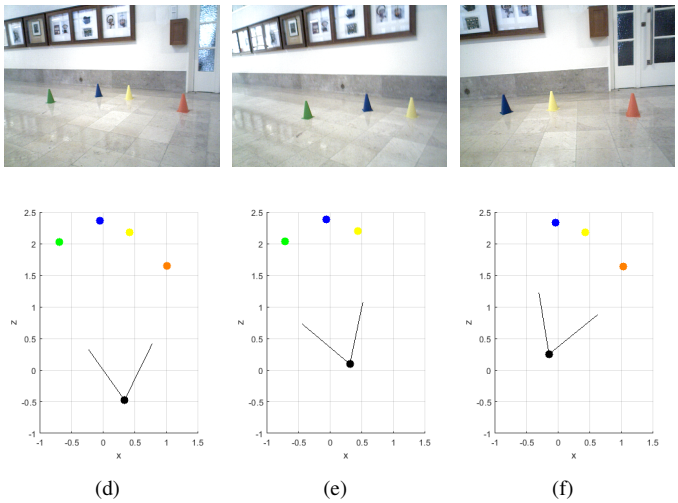


Fig. 7. Estimated camera and cones localization on scenario 1. The camera localizations are represented by black dots, and each cone by a dot of its respective color.



(d) (e) (f)

Fig. 8. Localization ans orientation error of the camera on scenario 1

are consistent with the test sequence used, allowing to consider its correct estimation for the 303 images where the pairing of the cones with the global map occurred.

On figure 10, is represented the estimated localization of the camera along the course. To evaluate the orientation estimation, on figure 11 is represented three estimated localizations and orientations

The estimation of location and orientation are consistent with the test sequence used, allowing to consider its correct estimation for the 303 images where the pairing of the cones

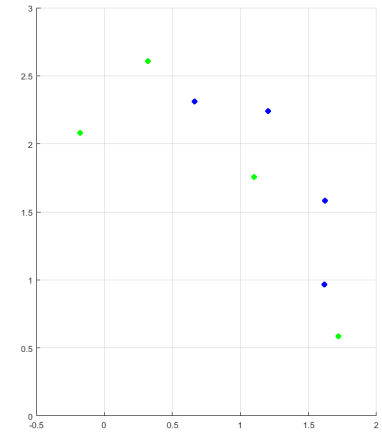


Fig. 9. Estimated location of the cones on scenario 2

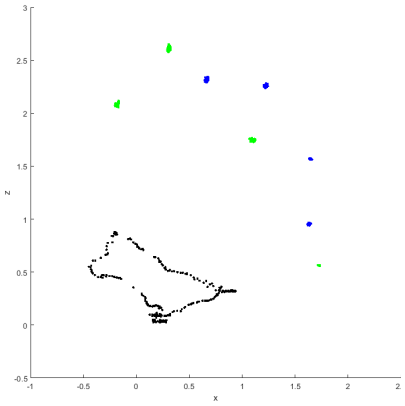
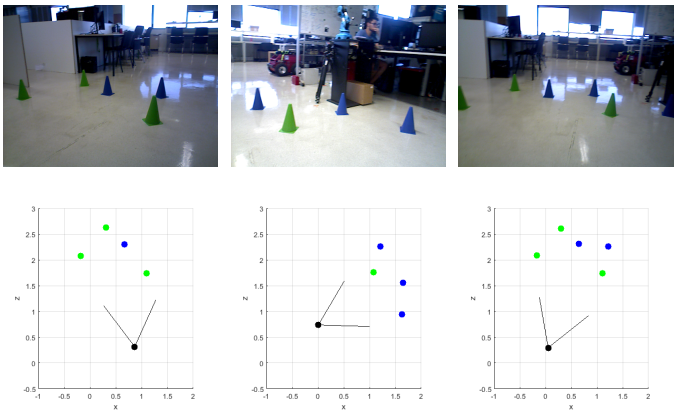


Fig. 10. Estimated camera and cones localization on scenario 2. The camera localizations are represented by black dots, and each cone by a dot of its respective color.



(d) (e) (f)

Fig. 11. Localization ans orientation error of the camera on scenario 2

with the global map occurred.

V. Conclusion

The present dissertation focus on the problem of estimation the camera pose along a previously known course delimited by multiple equal objects. It is proposed a solution, where using the distance pattern between observable objects, it is intended to correspond a local map with a global map through a developed search algorithm.

Each of the methods presented in the proposed solution were tested individually to ensure its correct operation. The training performed to the YOLO algorithm allows to detect cones of four different colors with a good certainty. The constructed global maps are close to the actual dimensions. The search algorithm shows robustness to the presence of small variation of the distances between objects. On the real situation tests, the system shows good results on the estimation of the localization and orientation of the camera.

The tests performed to the developed system show that the correct estimation of the camera pose at each instant is directly associated with the correct pairing of local and global maps, which reinforces the need to construct maps close to the real.

References

[1] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, 2015.

[2] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, 2016.

[3] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, 2017.

[4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011.

[5] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ser. ICCV '99. Springer-Verlag, 2000, pp. 298–372.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[7] J. Dijksterhuis, P. Gower, J. Gower, G. Dijksterhuis, and C. Dijksterhuis, *Procrustes Problems*, ser. Oxford Statistical Science Series. OUP Oxford, 2004.