Object Detection via Receptive Field Co-occurrence and Spatial Cloud-Point Data

Luis A. Contreras[1], Abel Pacheco-Ortega[1], Jose I. Figueroa[1], Walterio W. Mayol-Cuevas[2], and Jesus Savage[1]

[1]Biorobotics Laboratory, Faculty of Engineering, National Autonomous University of Mexico
[2]Department of Computer Science & Bristol Robotics Laboratory, University of Bristol

***Abstract*— The use of image and spatial information together in mobile robots systems it is a promising field, due to the enhanced level of discrimination and efficiency that can be gained. In this paper we employ an RGB-D camera for object detection and clustering and develop methods that combine the two strands of information: first we clusterize potential objects by mean of their spatial position and then link geometry and co-occurrence histograms to enable reliable object detection. Experiments and design parameters are presented for example scenarios of object detection under clutter.** [1]

I. Introduction

Detecting what is in front of a robot is arguably one of the most important questions underpinning autonomous operation. The difficulty is not only on the way signals and prior data are to be combined but on the selection algorithms to deliver rapid and reliable answers. Object detection is no different in respect of the varied opportunities that materialize when an object has been correctly identified and positioned in space.

In terms of approaches in Robotics, visual-only sensing has been developed quite significantly mainly due to the affordability of cameras which has only recently began to be the situation for cameras that return 3D information as well.

Using an RGB-D sensor we are interested in developing methods that combine fast operation with reliable results from both depth and visual signals. In particular we look at a method that benefits from rapid visual and invariant texture description but enhanced with 3D information to increase object detection performance.

We further enhance the computational performance of the methods by exploiting the available scale and depth information.

By combining fast visual methods with 3D information we then demonstrate the use of the developed methods on the problems of location and object detection.

II. Related Work

At some level, place and object detection are very similar problems but that operate at different scales. In the case of place detection one could benefit from the stream of data to build-up certainty for location when in motion e.g. as in [1]. In the case of object detection it is 3D pose that is commonly desirable and dealing with the more likely case of occlusion from other objects. However the initialisation of both place or object detection is bound to have common approaches.

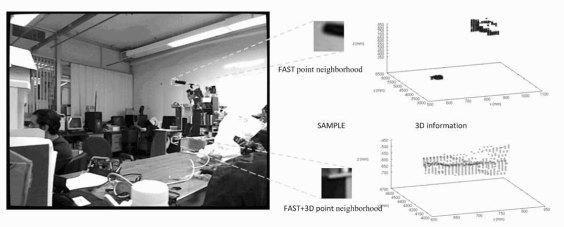


Fig. 1. Example of points detected by FAST, FAST+3D, with their respective neighborhoods and three-dimensional information

The overall processing pipeline for both place and object detection follows some well defined stages: salient features are detected, description that builds-in some level of invariance is performed, comparison of the extracted information (matching) with a database is made before a final step of decision making.

The techniques for image-only description that have been developed so far are many. But several steam out of the seminal work of the Scale Invariant Feature Transform (SIFT) by Lowe in [2] which has shown to have a degree of stability to changes in lighting, orientation and scale. The SIFT descriptor is built from the computation of histograms of gradients in regions of size 4x4 in 8 directions, obtaining, in the end, a feature vector of 128 elements. A good review of feature descriptors can be found in [3].

One of the key aspects of SIFT is the invariance to scale, but when tracking or if a depth sensor is available the need to compute this computationally involved stage is not necessary as some have observed [4], [5].

Other improvements to the processing pipeline have looked at the visual saliency and explored using faster methods. In [6], a combination of FAST [7] and the Shi-Tomasi operator [8] is used. FAST is a rapid assessment procedure which examines 16 pixels around a candidate feature point (Fig. 2(a)) to determine saliency.

The stages of detection and description have the purpose of characterizing scenes. The SIFT-type descriptors have a large dimensionality, because of this it is necessary to develop techniques for storage and search. Bentley in [9] developed search trees which allow to store and perform comparison

[1]This work was partly supported by PAPIIT-DGAPA UNAM under Grant IN-107609

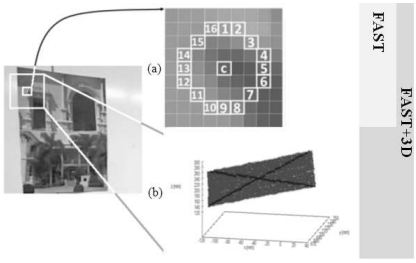


Fig. 2. Feature detection in FAST and FAST+3D

on a cluster of vectors which have been systematically stored in the tree. Later, Beis and Lowe in [10] made modifications to these trees which accelerated the search in high-dimensionality spaces.

Because isolated matching of salient described points is usually not enough to establish a positive detection, robust checks are usually performed with some well established algorithms such as RANSAC [11], Joint compatibility [12] or the notion of co-visibility e.g. [13].

Furthermore, techniques for searching for individual objects when its distance changes in a significant way are needed. As it happens in the environment of a mobile robot, detecting objects which are far away is a complex task due to sensor resolution and non-linear changes in accuracy. A way of solving this problem is using optical zoom but this attracts a number of control and attention selection issues.

Another approach to solve this is multiple-scale analysis, where mathematical techniques exist to extract spatial features which are not variant to scale changes [14], [15], [16], [17], [18].

In this work, receptive fields are employed, to make a representation of each pixel of the entire image as a combination of its color and the response to several filters (feature extraction). The next step is to apply histograms of co-occurrence to the output of the former algorithm, to represent the image as a combination of descriptors in a determined radius; in this way, geometrical information of the image is added to the features.

In summary, techniques to process place and objects are similar and the recent affordability of depth cameras makes it reasonable to ask what processes can be extended from previous works that have developed fast vision-only methods to tackle these problems. In this case we want to benefit from image data to quickly extract saliency and use geometry from the 3D information to expand on description algorithms and improve accuracy of detection.

In the next sections we describe our approach for the different stages of the above processing pipeline before describing our case studies and experiments are presented.

III. Place Localization System

A. Scale-Depth Function

Perhaps one of the first tasks is to calibrate the RGB-D sensor from the point of view of distance to objects.

In order to do this an object of known dimensions is used as input. The object is set at different distances and images of it are captured. By using this data a simple linear relationship can be obtained. In our case it is

$$y = 0.001716x + 0.2735 \tag{1}$$

where y (mm/pixel) represents the dimension in the image of the object at the depth x (mm).

Therefore, to get the scale factor of a reference image at an arbitrary depth $depth_i$ relative to a fixed depth $depth_0$ the following relationship is employed:

$$scale = \frac{y_0}{y_i} = \frac{0.001716 * depth_0 + 0.2735}{0.001716 * depth_i + 0.2735} \tag{2}$$

B. FAST+3D

As argued before, we would like to benefit from existing work on fast methods developed for vision-only sensing and enhance these with 3D data. One approach could be to use a 3D saliency detector first e.g. as in [19], but dealing with 3D data directly is necessarily more involved than using only a 2D signal. In this case we take the approach of using a rapid saliency 2D image discriminator which filters out vast areas of the image before we investigate 3D structure more closely.

In this respect we have combined the FAST detector that works on the RGB image followed up by a neighborhood check on the 3D data. This technique we therefore refer to as FAST+3D.

After FAST is computed the first filter removes all those points given by the FAST algorithm where the depth camera is unable of getting uniform three-dimensional data. The rationale is that if we are to use a local descriptor to be invariant from large viewpoint changes we want to ensure that the descriptor around the selected salient points is to be recoverable when needed.

A neighborhood analysis in 3D is thus performed around 2D salient salient points. The size of the neighborhood to be analyzed is determined by the depth-scale function; experimentally, a square with $25(pixels)$ by side is chosen as reference patch size of the neighborhood ω_0 around any point at a depth $depth_0 = 1700(mm)$ (See 2). The neighborhood size of any point at a given depth is stated as: $\omega_i = int\left(\omega_0 * scale + 0.5\right)$ substituting $\omega_i = int\left(25 * scale + 0.5\right)$

Chekhlov et al in [4] have demonstrated that predicting the scale of the feature points increases the efficiency in the stage of matching descriptors thus removing the need to compute scale invariant description from scratch. The RGB-D camera and the distance-scale analysis allows to determine the size of the neighborhood to analyze and avoid making predictions.

The set of points generated by FAST+3D is defined as

$$FAST + 3D = \{p_i \in P_{FAST} \mid gneig(p_i) = 1\}$$
$$gneig(p_i) = \begin{cases} 0 & \text{if } diag(p_i) \geq \tau \\ 1 & \text{other} \end{cases}$$
$$diag(p_i) = \frac{r}{2\omega_i + 1}$$

where P_{FAST} is the set of points computed by FAST, r is the number of pixels in the diagonals of the neighborhood where the difference of depths between the middle pixel

Fig. 3. Neighborhoods computed by (a) FAST and (b) FAST+3D. Notice how many regions that are less likely to be repeatable at different viewponts (real object corners and edges) are removed by FAST+3D.

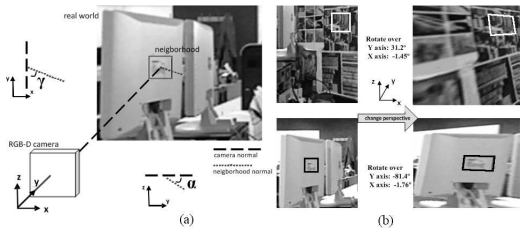


Fig. 4. Analysis of a neighborhood

and the analyzed pixel of the diagonal is not bigger than a threshold ρ, τ is the percentage of points which are outside the threshold which discards a point as a candidate. For this work, the value of ρ is defined experimentally with a value of 200 millimeters.

Note that this is only one way in which the neighborhood check can be performed. We chose this one because it is faster and proved reliable enough experimentally. A more thorough test would be to match a local 3D plane to every 2D FAST point on its corresponding 3D locality via RANSAC and measure deviation from planarity to determine saliency. But this is more time consuming and the viewing angle that can already be achieved with the faster simplification appears not to justify the extra burden.

The Fig. 1 shows example locations where a change in the position of the camera may generate a change in the neighborhood of the point returned by FAST. However, a point returned by FAST+3D would have a less significant change in its neighborhood due to the above checks. The Fig. 3(a) shows the points computed by FAST in a given scene, most of these points lack a stable neighborhood which means that the description of the region is not recommendable. In the Fig. 3(b) the points which are going to be described are shown, their number is smaller but the neighborhood more stable.

C. Planar neighborhood descriptors

As argued before, SIFT is still state of the art in terms of description performance. These descriptors have been shown to be robust to a number of conditions, however it presents problems on perspective changes with angles above plus or minus 45 degrees.

At this point, the scale of each FAST+3D salient point is available and therefore the size of the neighborhood (in pixels) which is going to be analyzed is easy to extract. This allows the information obtained with the RGB-D camera to be used to rectify the construction of the descriptor, in the same way that it is used in the description stage.

For each point returned by FAST+3D an analysis with RANSAC is applied to discard 3D outliers from the plane which better represents the neighborhood. Principal Component Analysis (PCA) is then applied to the outlier-free point set, where the eigenvector associated to the minimum eigenvalue is the normal of the plane to be searched.

With both the normal vector of the neighborhood and the plane of the image (y-axis) the angles α and γ are computed, which are the angles of the projection of these normal on the planes YZ and XY respectively, as shown in the Fig. 4(a). These angles are used to apply perspective changes to the image region in such a way that the normal vectors are parallel, the effect can be observed in the Fig. 4(b).

This transformed neighborhood is re-dimensioned to fit 25x25 pixels and it is described in the same way as in the SIFT algorithm (only the histogram of gradients part). In this way, each point detected by FAST+3D is stored as a vector whose structure contains the image and world localization and the descriptor with 128 elements.

In our experiments, to process a typical image with the combination of FAST+3D and the SIFT Histograms on the local neighborhoods is about 3-4 times faster than using SIFT alone (Table I) as well as being more viewpoint invariant.

D. Scene Matching

During the training stage, the descriptors of the image and their three-dimensional information, are stored in a k-d tree [9].

The matching of the image (testing stage) is done by comparing the euclidean distance between descriptors, the comparison process is sped up by using the Best Bin First algorithm [10].

Afterwards, the pairs of descriptors are processed by the RANSAC algorithm to discard outliers, so that at least four pairs of reliable descriptors are obtained; if this condition is not satisfied, it is considered that a match does not exist.

E. 6D Pose Estimation

When the relationship between the scene data and a previous set is determined, it is possible to perform the computation of the relative positions of the 3D information stored from the two viewpoints where the original data came from.

In this step, there are two sets of points with location information both in metric space as well as within its respective image. Finding the relationships between both coordinated systems which define them, given the relationships formerly established, the problem is reduced to the absolute orientation which has been thoroughly studied in photogrammetry.

There are many possibilities here in recent and early literature. At [20] Horn presents a closed-form which is applied to our method so can be computed the translation vectors T_i and the rotation vectors R_i which relate the two positions of the RGB-D camera where the same pattern is

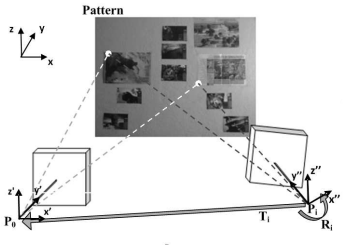


Fig. 5. Computing the absolute orientation of a pattern

observed, in other words, as it is shown in the Fig. 5, the localization of a given point P_i relative to the position P_0 (pattern) given the relationships between descriptors seen from different viewpoints can be performed.

IV. Object Detection System

The above steps offer methods that are fast and able to match previous RGB-D data to a current sample from a different 6D viewpoint. These are mainly useful for place detection but to perform small object detection in a specific place we need to add an extra step to deal with the more challenging aspects of small scale, larger viewpoint invariance (if the object is e.g. resting on a very different pose from the one observed) and occlusion by other objects.

A. Receptive Field Co-occurrence Histograms+Wavelet Transform

For the object detector the Receptive Fields Co-occurrence Histograms (RFCH) [21] are used. For this, a series of filters are applied to a reference image, as follows:

- The first derivative (**Gradient operator**) of the image $f(x, y)$ at position (x, y), defined as the vector:

$$\nabla f = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial x \end{bmatrix}$$

- The second derivative of an image (**Laplacian operator** of the image $f(x, y)$ at position (x, y), defined as the vector:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

As it is described by P. Chang *et al* [22], a co-occurrence histogram represents the amount of occurrences between pairs of arrays separated by a vector on the plane of the image $(\Delta y, \Delta x)$, and symbolically is represented as $CH(p_1, p_2, \Delta x, \Delta y)$.

The histogram is made invariant to rotations on the plane of the image by ignoring the direction of $(\Delta x, \Delta y)$ and just considering the value of the magnitude $d = \sqrt{(\Delta x)^2 + (\Delta y)^2}$.

In order to characterize an image, for each pixel a vector of length 5 is computed, using the HSV value , as well as the first and second derivatives. From those vectors, a set of clusters is obtained by k-means clustering. An index is assigned to each cluster and a set of indexes is computed by mapping every pixel in the image to its correspondent cluster index. To describe the image, the histogram of co-occurrences is applied to the set of indexes. Whenever an object wants to be detected in a new image, the same descriptors are computed for each pixel and the pixels which do not belong to the reference clusters are filtered out, as shown in the Fig. 6. From the set of filtered pixels, the co-occurrence histogram is computed through windowing and if the intersection between the histogram of the window and the reference clusters exceeds a threshold, an hypothesis of the location of the object is obtained.

Fig. 6. Stages of Segmentation by RFCH: a) Reference Image, b) Original scene, c) Segmented Scene

To improve the detection rate and reduce false positives, we carry on a Recognition Of Common Objects by Co-Occurrence analysis (ROCOCO analysis) which includes the object texture. As in [23], the Wavelet Transform is used as a texture descriptor of the interest object to reduce the amount of false positives. The Haar Wavelet Transform is proposed in this work, because it is fast enough to be used in real time applications. The transform consist in a filter bank, low pass and high pass, which is applied on the image. Be H (averages) and G (differences) a high pass filter and a low pass filter, respectively, for an image A, we have:

$$\begin{aligned} B &= \begin{bmatrix} H \\ G \end{bmatrix} A \begin{bmatrix} H \\ G \end{bmatrix}^T = \begin{bmatrix} H \\ G \end{bmatrix} A \begin{bmatrix} H^T \\ G^T \end{bmatrix} = \begin{bmatrix} HA \\ GA \end{bmatrix} \begin{bmatrix} H^T \\ G^T \end{bmatrix} \\ &= \begin{bmatrix} HAH^T & HAG^T \\ GAH^T & GAG^T \end{bmatrix} \end{aligned}$$

The texture descriptor is extracted from the variance $\sigma^2_{n,i}$ of the coefficients $c_{n,i}$ of the details of the image obtained from HAG^T, GAH^T y GAG^T on all its n levels.

To represent the texture of an image A, the texture descriptor vector is stated as:

$$\begin{aligned} T_{HWT} = [&(\mu_{1,1}, \sigma^2_{1,1}), (\mu_{1,2}, \sigma^2_{1,2}), (\mu_{1,3}, \sigma^2_{1,3}), \ldots, \\ &(\mu_{max,1}, \sigma^2_{max,1}), (\mu_{max,2}, \sigma^2_{max,2}), (\mu_{max,3}, \sigma^2_{max,3})] \end{aligned}$$

where M_{max} indicates the maximum scale. In this work, $M_{max} = 3$, a vector of 9 components is used to describe the texture. If the Euclidean distance between the reference vectors and the hypothesis vector exceeds a threshold, this new hypothesis is considered for the next stage.

Finally, by employing the depth information it is expected to make the system robust to large changes in scale between the reference and the search image by scaling the reference image according to the depth-scale function, as we present as follows.

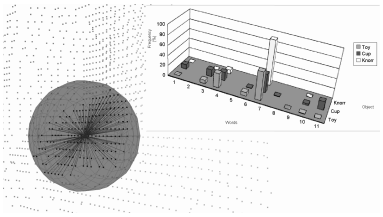
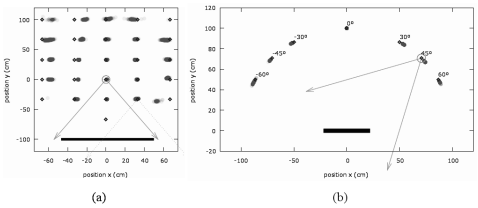


Fig. 7. Three-dimensional neighbors around a point



(a) (b)

Fig. 8. Comparison between Estimated Position and Angle against Real Position and Angle

ROCOCO+3D

In order to enhance the detection rate, the three-dimensional information of the object is used. For such purpose, a descriptor of a point is created:

1) A sphere is centered in an object point to be described.
2) The histogram of the distances inside the sphere from the central point to every other point is calculated.

This stage is similar to recent methods for object classification [24] but the addition of the image based co-occurrence data helps to get better results than only geometry as well as the image data helps to get to the area of potential interest faster.

To manage the local description we use a word vocabulary. In order to generate a vocabulary of the world, the histograms of every point in a complex scene which contains multiple volumetric shapes is obtained, and the clusters which group them in the best way is calculated.

For getting these histograms, we generate spheres with a specific $radius$ around every point in the object of interest. For this, we should keep in mind that we are working with 2D images which have the depth information, and not vice versa. So, we have to get the sphere in terms of pixels, generating first a 2D square in the image that encloses the sphere as follows:

$$Sside = \frac{radius}{0.001716(depth_i) + 0.2735}$$

where $Sside$ is the size in pixels of the square centered at the interest point at $depth_i$. For each pixel in this square we get the spatial information and calculate the Euclidean distance to the central one, and generate a full length vector, which finally is reduced by calculating the distance histogram, referred as $word$.

To describe an object, the word for each point is obtained, and a phrase (normalized histogram of words) is created as shown in Fig. 7 (histogram insert). The matching is done by intersecting these phrase with the database phrase.

V. Experiments

Based on the above, our modified image+3D methods are tested. In this case we use a MS Kinect sensor as the RGB-D camera set at a 2D resolution of 640x480 pixels.

A. Localization method experiments

In the first experiment, we evaluate the response of the sensor. An image pattern is captured at a controlled distance of 1 (m), in which the observed pattern is always set on a plane which is parallel to the plane of the camera image, then, the camera is displaced 33 centimeters in both coordinates axes x and y, whose positions are represented by the black crosses in the Fig. 8(a), and the localization method is executed 200 times on each position. The acceptance threshold values for our FAST+3D algorithm are 60 for the training stage, and 50 for the testing stage.

	SIFT			FAST+3D and SIFT descriptor		
Angle	% Times matched	Points per image	Execution time [ms]	% Times matched	Points per image	Execution time [ms]
60	2	779	2077.51	100	225	705.53
45	100	860	2210.41	100	290	735.59
30	100	889	2331.35	100	297	728.80
0	100	846	2582.45	100	265	706.31
-30	100	867	2442.66	100	193	560.79
-45	100	880	2429.34	100	152	735.59
-60	3	826	2254.08	94	181	632.03
avg		**850**	**2332.54**		**228**	**686.38**

TABLE I
COMPARISON BETWEEN SIFT AND FAST+3D

It is observed that the positions returned by our methodology have, in general, an adequate localization, where errors tend to increase as the camera goes further from the reference; the average running time is 574 (ms), having 449 descriptors stored in a k-d tree and an average of 157 descriptors generated by execution. The average errors are of 3.75 (mm) at the x position and 0.39 (mm) at the y position. However, there are positions where the amount of matched was minimum (1% of the executions in that position) or null, either because they are too far from the pattern and the information returned by the camera is insufficient or because they are too close to the pattern, around the zone where the camera is unable to return spatial information.

In the next experiment, the plane of the observed pattern was set on different angles over the Z axis relative to the plane of the camera image. The standard SIFT algorithm was executed in the same positions, to visualize its robustness under changes of perspective and to compare it against the proposed method (see Fig.8(b)), the table I shows the results obtained in the comparison.

The test shows that SIFT works fine with perspective changes up to 45 (degrees) but loses precision when the angle is increased, while our methodology of FAST+3D with a local planar neighborhood keeps making matches in most of the tests.The time spent by the SIFT algorithm to detect and compute the descriptors is longer due to the execution of the analysis of space-scale which it is not needed to perform in our approach as this is recovered from the depth values

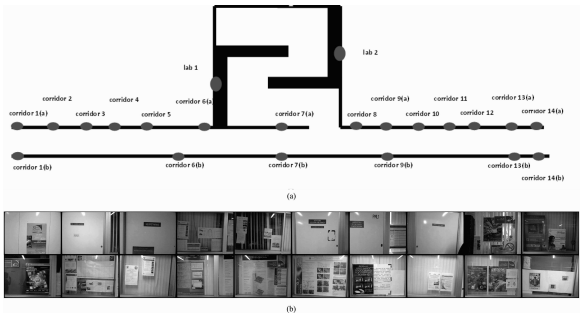


Fig. 9. a) map of locations b)From left to right, top to bottom¡: lab 1, lab2, corridor 1(a) corridor 1(b), corridor 2, corridor 3, corridor 4, corridor 5, corridor 6(a), corridor 6(b), corridor 7(a), corridor7(b), corridor 8, corridor 9(a), corridor 9(b), corridor 10, corridor 11, corridor 12, corridor 13(a), corridor 13(b), corridor 14(a), corridor 14(b)

returned by the RGB-D camera. Our method is able to match with perspective changes of 72 (degrees) and is faster.

Finally, the method is tested in a real environment, which is performed in sections of an office corridor.

In the training stage, a set of patterns are extracted from a number of positions. The location and the images of the patterns are presented in the Fig. 9. The lighting conditions and the amount of feature points on each reference pattern present some variation, due to this, it is able to observe the performance on each pattern.

In the testing stage, the camera is set an average of 168 times in random positions and facing all the time to the zone where it is expected to perform a location. Every time a matching is performed, the method returns the location where that pattern is captured. It is also worth mentioning that the motions in the location of the camera were done by hand and the execution of the method was continuous.

At some locations, where the lighting conditions are poor to such degree that the rate of matching was low, due that the amount of points returned by FAST+3D was insufficient; another important factor is that some captures of images are not stable enough because the camera is moving and the image was not sharp enough. For the characteristics of the work environment, a lot of flares are found in the the images, which has an influence in the performance. The Tab. II presents a summary.

B. Experiments for Object Detection Method

For the case of the object detector using the co-occurrences algorithm, an elevated amount of false positives in zones with a color distribution similar to that of the searching pattern is observed. In the case of the ROCOCO analysis with wavelets as texture descriptors a reduction in the false positives amount is observed, however the objects which look too similar in the image are detected as the same object.

Thanks to the three-dimensional descriptors, namely RO-COCO+3D, which take into account the geometry of the object, a better discrimination between similar objects is achieved. In Fig. 10, the discrimination between a cylindrical object and a rectangular one is observed. There are multiple objects, different scales and clutter. The algorithm was tested

Localization	Times	Execution time (ms)	% Localization	# FAST+3D points	# Points matched
labo 1	368	180.08	8.70	155	7
labo 2	127	496.34	97.64	215	48
corridor 1	160	245.10	54.38	128	30
corridor 2(a)	87	57.43	0.00	0	0
corridor 2(b)	181	235.36	74.59	105	23
corridor 3	387	73.81	1.81	26	6
corridor 4	200	184.07	63.00	42	10
corridor 5	148	384.65	64.86	195	25
corridor 6(a)	181	211.62	63.54	87	16
corridor 6(b)	83	70.84	0.00	0	0
corridor 7(a)	170	230.36	53.53	92	17
corridor 7(b)	70	402.02	62.86	146	17
corridor 8	86	746.56	90.70	373	51
corridor 9(a)	60	161.10	43.33	63	18
corridor 9(b)	107	204.22	71.96	62	17
corridor 10	225	157.92	14.22	73	7
corridor 11	265	303.05	64.53	148	21
corridor 12	256	285.50	50.39	171	25
corridor 13(a)	32	1832.60	65.63	904	84
corridor 13(b)	233	135.49	33.91	43	8
corridor 14(a)	216	394.76	64.35	200	37
corridor 14(b)	74	272.34	70.27	115	21

TABLE II

DETAILED DATA OF PLACES WHOSE MATCHES ARE ABOVE 50%

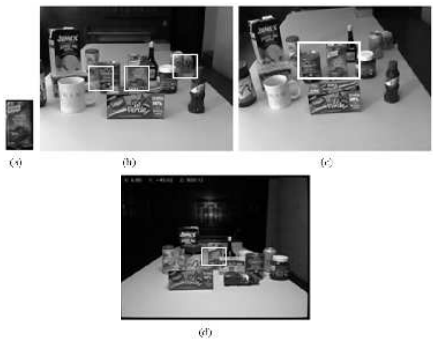


Fig. 10. Object Detection Test, a) Reference Image; Object Detection: b) RFCH, c)ROCOCO, d) ROCOCO+3D

on video sequences and the method runs at 5 (fps) for a single object look up and it is able to detect the reference object from different positions and angles on each frame (tracking). In Fig. 11 the detection and recognition rates are shown for the case of 14 objects, 11 of them were trained and the other three were left as control objects. It can be observed that while the detection rate frame by frame is low in some cases, when it detects an object the probability of being the desired object is high (recognition rate).

Also, through the scaling in depth, it is shown in the Fig. 12 the level of detection to varying distances, as well as the robustness to illumination changes.

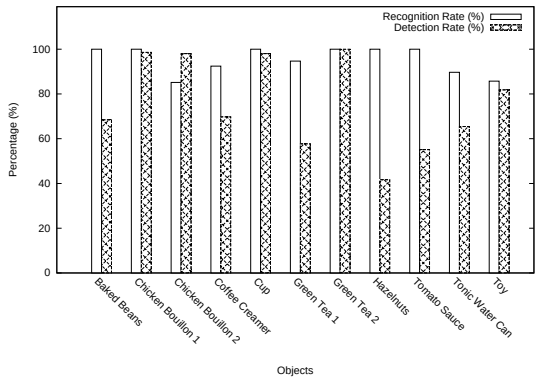


Fig. 11. Algorithm performance test: Detection and Recognition Rate

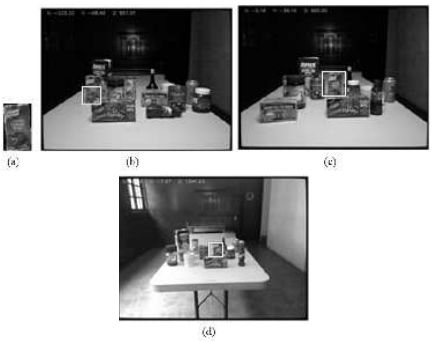


Fig. 12. ROCOCO+3D Object Detection Test, a) Object trained at 523 (mm); Object Detection at: b) 857 (mm), c) 869 (mm), d) 1347 (mm)

As this is an experimental approximation of a three-dimensional descriptor of an image, the performance are obtained by varying both the size of the word for each point given by the sphere radius, as well as the size of the phrase which describes an object, in which it was gotten an linear behavior for both cases. For the radius varying, we got an proportional model with slop m = 0.2522, it is, every 3.96 millimeters the time doubles itself. For the cluster varying, the slope was m = 0.0449, so for increasing the phrase by 22.27 words we have twice the execution time.

VI. Conclusions

In this paper we presented two methods for place localization and object detection using depth information from RGB-D cameras. For localization, we develop FAST+3D which takes advantage of a rapid 3D neighborhood analysis around 2D salient points in order to eliminate unstable ones. These points are then used to generate robust descriptors based on a histogram of gradients after a region normalization in 3D. The method is faster and more view invariant than standard SIFT.

For the object detector, we developed an improvement of the RFCH method combining texture and 3D information, the ROCOCO+3D analysis.

We tested these techniques in real environments for place and cluttered object detection with good speed and detection performances over baseline methods. It was also tested in the RoboCup@home 2012 scenario, having succesful results.

References

[1] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, November 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=993451.996342

[3] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 1615–1630, October 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1083822.1083989

[4] D. Chekhlov, M. Pupilli, W. Mayol-cuevas, and A. Calway, "Real-time and robust monocular slam using predictive multi-resolution descriptors," in *In 2nd International Symposium on Visual Computing*, 2006.

[5] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "A constant time efficient stereo slam system," in *BMVC*, 2009.

[6] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway, "Robust real-time visual slam using scale prediction and exemplar based feature description," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.

[7] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *In International Conference on Computer Vision*. Springer, 2005, pp. 1508–1515.

[8] J. Shi and C. Tomasi, "Good features to track," in *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, pp. 593 – 600.

[9] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, pp. 509–517, September 1975. [Online]. Available: http://doi.acm.org/10.1145/361002.361007

[10] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," 1997.

[11] J. D. Foley, M. A. Fischler, and R. C. Bolles, "Consensus: A paradigm for model fitting with apphcatlons to image analysis and automated cartography."

[12] J. Neira, J. Tardos; Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, 2001.

[13] C. Mei, G. Sibley, and P. Newman, "Closing loops without places," in *IROS*, 2010.

[14] F. A. Cheikh, A. Quddus, and M. Gabbouj, "Multi-level shape recognition based on wavelet-transform modulus maxima," in *Proceedings of the 4th IEEE Southwest Symposium on Image Analysis and Interpretation*. Washington, DC, USA: IEEE Computer Society, 2000, pp. 8–. [Online]. Available: http://dl.acm.org/citation.cfm?id=829535.831606

[15] R. M. Willett and R. D. Nowak, "Platelets: A multiscale approach for recovering edges and surfaces in photon-limited medical imaging," *IEEE Transactions on Medical Imaging*, vol. 22, pp. 332–350, 2003.

[16] D. L. Donoho, X. Huo, I. Jermyn, P. Jones, G. Lerman, O. Levi, and F. Natterer, "Beamlets and multiscale image analysis," in *in Multiscale and Multiresolution Methods*. Springer, 2001, pp. 149–196.

[17] B. Romeny, *Front-End Vision and Multi-Scale Image Analysis: Multi-scale Computer Vision Theory and Applications, written in Mathematica*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[18] M. Masood, "Multi-scale analysis techniques in pattern recognition systems," 2005.

[19] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "Point feature extraction on 3d range scans taking into account object boundaries," in *International Conference on Robotics and Automation*, 2011.

[20] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, pp. 629–642, 1987.

[21] S. Ekvall, D. Kragic, and F. Hoffmann, "Object recognition and pose estimation using color cooccurrence histograms and geometric modeling," *Image Vision Comput.*, vol. 23, pp. 943–955, October 2005. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2005.05.006

[22] P. Chang and J. Krumm, "Object recognition with color cooccurrence histograms," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, p. 2498, 1999.

[23] T.-W. Chen, Y.-L. Chen, and S.-Y. Chien, "Fast image segmentation and texture feature extraction for image retrieval," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 27 2009-oct. 4 2009, pp. 854 –861.

[24] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *The IEEE International Conference on Robotics and Automation (ICRA)*, 2009.