Laboratório VISGRAF
Instituto de Matemática Pura e Aplicada

Object Recognition using RGB-D images:
Some preliminary results

Noslen Hernandez
Luiz Velho(orientador)

Technical Report TR-2012-05 Relatório Técnico

October - 2012 - Outubro

Object Recognition using RGB-D images. Some preliminary results

Noslen Hernández Luiz Velho

Abstract

This research on Object Recognition using RGB-D images was performed during the three month pos-doctoral stay of PhD. Noslen Hernández at the VISGRAPH Lab at the Institute of Pure and Applied Mathematics (IMPA). This note show the ideas that were explored as well as the experimental results reached. Some conclusions and future line of work are given at the end.

1 Introduction

With the recent emergence of fast and inexpensive RGB-D sensors, like the Microsoft Kinect [1] camera, the integration of visual and geometric information of an scene has became easier to obtain. Such kind of devices outputs a colored image and the depth of each pixel in the scene. This opens the opportunity to take advantage of both intensity (color) and depth information.

Object recognition is one of the applications found by RGB-D technology. Depth data have been used in many different ways to help the object recognition task. New geometrical descriptors have been proposed [2, 3, 4, 5] that enables to extract the shape information from depth images. Among the most used are Spine images [6] and Normal Aligned Radial Feature (NARF) [7].

Many of the existing approaches for object recognition on RGB-D images combines some of the aforementioned geometric descriptors with image descriptors like SIFT [8], SURF [9] or HOG [10]. Many of them focus on the extraction of SIFT or HOG descriptors on grayscale images generated from depth data [11]. Also, a new family of features that turns any pixel attribute (gradient, color, local binary pattern, etc) to patch-level features, called kernel descriptors [12, 13, 14], have been generalized to depth data [15]. An Instance Distance Learning [16] algorithm was introduced also to carry out this fusion.

In this report it is investigated a new way of combining appearance and geometrical information, but inside the feature descriptor. Contrary to the existing approaches, instead of fusing the results given by different images and geometrical descriptors, we intend to localize the keypoints and generate the descriptor taking into account both source of information. This ideas were carried out modifying the SIFT descriptor. In this way, the color and depth information are combined inside SIFT.

An scheme for object recognition based on the bag of features framework is proposed. Bag of features is probably one of the most popular image representations. Cluster Ensemble methods are used for the construction of the codebook or dictionary. Based on that dictionary, histograms are built and used as image descriptors. A new similarity measure for the comparison of those histograms is introduced. For the experiments, the RGBD Object Dataset [11] was used.

2 Combining Depth and Color information inside SIFT

In this section we will describe the modifications done to the standard SIFT algorithm in order to take into account both, color and depth information. In the setting of RGB-D images, on each spatial coordinate (or pixel), we have four channels, three storing the color and one with the depth values. Suppose that we have the RGB image converted to grayscale, then we can assume that on each pixel we have a two-dimensional vector $p = (i, d)$, containing intensity and depth values. Our general idea is very simple: we want SIFT to extract features taking into account those two channels (two-dimensional vectors) jointly, instead of extracting SIFT features, separately, from color and depth images.

The SIFT algorithm have six basic steps. Our modifications focused mainly on step 3, in which the localization of the keypoints is perform. On this step, we intend to identify a unique set of keypoints using the intensity and depth images instead of two different set of keypoints, one for each image. Minor changes are done to the following steps in order to recover the descriptors.

1. Construct scale space
2. Take difference of Gaussians
3. Locate DoG Extrema to localize the keypoints
4. Get rid of bad keypoints
5. Assigning keypoints orientations
6. Generate SIFT descriptors

The scale space and the difference of Gaussians (DoG) were calculated independently for each channel. In this way, this step gives the same results as if they were computed separately on the intensity and depth images. Figure 1 shows the DoG images for a particular example in the database.

The next step is to find the keypoints through the localization of the DoG extrema. As in the standard SIFT algorithm, we will look for the keypoints by comparing each point with its neighbors in the DoG images (including the scales below and above) and choosing those who have no neighbors greater (or smaller) than it. But, as it is shown in Figure 2, on each spatial location for the DoG image, we have a two-dimensional vector. For that reason, the comparison

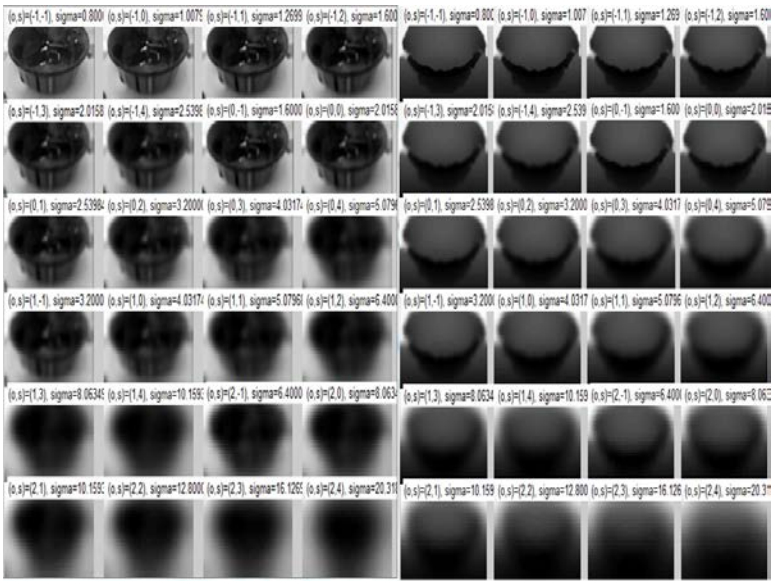

Figure 1: Difference of Gaussian (DoG) for a) intensity and b) depth images

will be made using the L_2 norm of such vectors. In this way, $p_g = (i_g, d_g)$ is going to be an optimum if $\|p_g\| > \|z_g\|, \quad \forall z_g \in V$ or if $\|p_g\| < \|z_g\|, \quad \forall z_g \in V$, where V is the set of neighbor points.

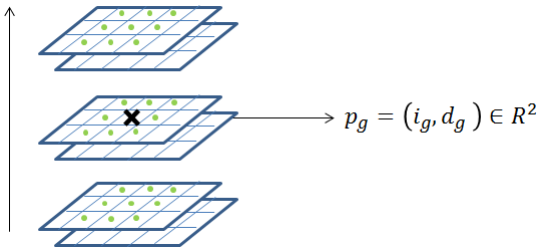

Figure 2: Comparisons in the DoG for RGB-D images

Notice that this criterium for finding the keypoints establish, in some way, a trade-off between the importance that have each spatial point on each image. It is clear that points who are extrema on the intensity image and also on the depth image will result extrema under this criterium. On the other hand, points who are extrema only for one channel may or may not be extrema when analyzing using the criterium for the two channels.

Figure 3 shows the keypoints found on the intensity and depth images when they are processed independently, (button row) and the keypoints found using the introduced criterium that fused both information.

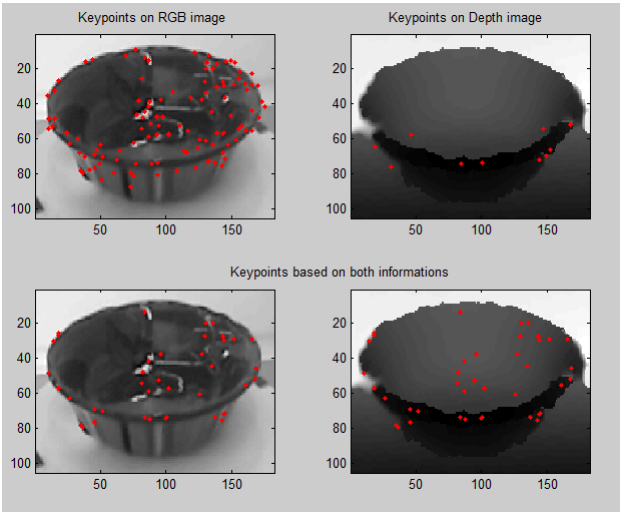

Figure 3: Keypoints found processing intensity and depth images independently (upper row) and keypoints found using both informations jointly (button row)

The number of keypoints found on intensity images is usually greater than the one found on depth images. The number of keypoints found with the new criterium, is usually smaller than the one found on the intensity image and greater than the one found on the depth image. This behavior can be observed in the example above. It can be seen also that we have obtained a unique set of keypoints for both images. What follows now is to determine what keypoints remains and generate the descriptors. All this will be done also using the intensity and depth images.

The analysis of which keypoints remains was done on each image separately, following the criteria of the standard SIFT. Those keypoints rejected on both images, were the ones who were rejected at the end. The orientation assignment and generation of SIFT features were done separately on each image and concatenated in one descriptor of dimension 256. If one image generates p descriptors (due to the assignment of various orientations) for a keypoint, and the other generates q, a total of pq descriptors are generated, consisting in the combination of all versus all. Figure 4 shows an example of the frames calculated processing the intensity and depth images independently (upper row) and jointly (button row).

The introduced way for achieving SIFT descriptors have some advantages and drawbacks worth of mention. One of the advantages is that, contrary to some approaches in the literature in which SIFT is calculated independently on the intensity and depth images, and then concatenated to form a descriptor of dimension 256 we have used a criterium that take into account both images at the same time. In this way we obtain a unique set of keypoints, whos localizations are considered important for both images. Other advantage is that

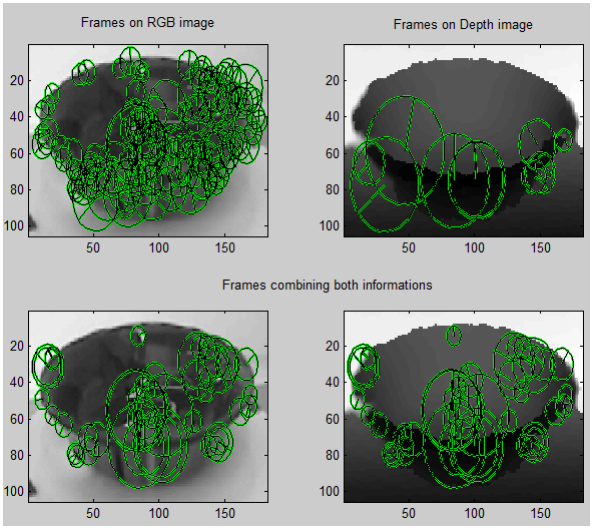

Figure 4: Gradient's orientation and magnitudes found processing intensity and depth images independently (upper row) and using both informations jointly (button row)

the number of keypoints will be reduced (and to some extent controlled). This have been an aspect of interest in the literature achieved for example using dense SIFT, thresholding the number of SIFT or using adaptive non-maximal suppression. One of the problems found to the introduced approach is that sometimes it generates very few keypoints, reducing the number of features dramatically. Figure 5 shows two examples in which this problem is present.

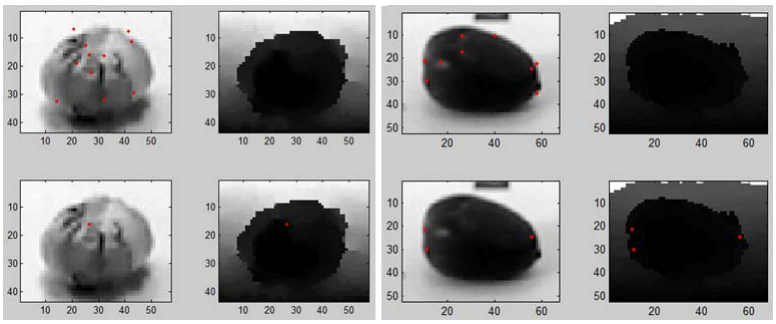

Figure 5: Keypoints found processing intensity and depth images independently (upper row) and keypoints found using both informations jointly (button row)

In the upper row of Figure 5 appears the number of keypoints found processing the intensity and depth image independently. It can be seen that for both examples, the garlic and the lime, few keypoints appear in the intensity

images while no keypoints appear in the depth image. When applying the new criterium, which looks for a trade-off between both source of information and reduces even more the number of keypoints, we obtain just one keypoint for the garlic and three keypoints for the lime. This happens mainly due to the information given by the depth image.

Another problem of this approach is that sometimes the information extracted from the depth image to construct the descriptor is not so useful. As we establish a unique set of keypoints for both images, we are forcing the algorithm to calculate part of the descriptor on that location that may have no useful information. This happens frequently when depth images have poor quality. Figure 6 show two examples of this matter.

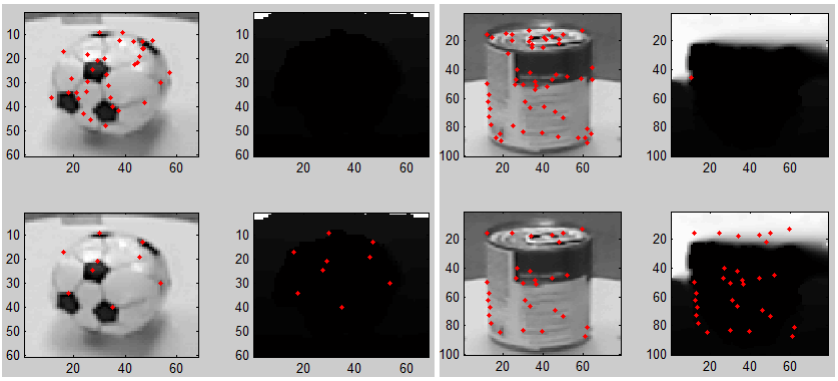

Figure 6: Keypoints found processing intensity and depth images independently (upper row) and keypoints found using both informations jointly (button row)

Notice that in these two examples the localization of keypoints again are mainly a consequence of the information given by the intensity image. Contrary to what happen in Figure 5, here the number of keypoints is not drastically reduced but we will force the algorithm to calculate descriptors in areas of the depth image where there is no so useful information. This can provoke noise and redundancy in the descriptor.

3 Bag of features. Construction of the codebook

The bag of features approach have been widely used for object recognition. This approach works by clustering local features vectors, such as SIFT descriptors. The objective is to construct a codebook (or visual word vocabulary) that contain a compact representation of the local images features and then, use that vocabulary to build histogram of visual keywords that describe the image. A graphical representation of this process can be seen on Figure 7.

The general steps of the bag of features approach can be summarized as follows:

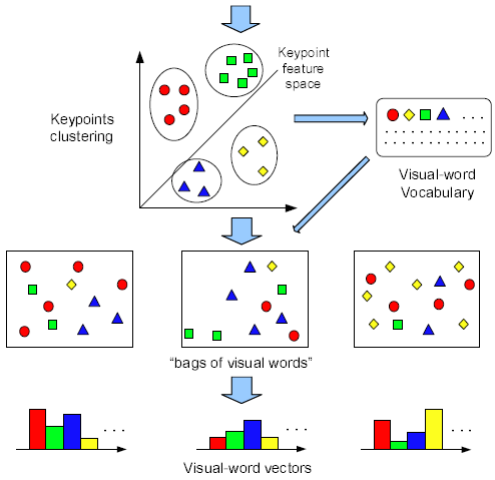

Figure 7: Graphical representation of Bag of Feature

1. Detection of region of interest using a keypoint detector
2. Description of the detected regions
3. Clustering of the descriptors to obtain a vocabulary (or codebook) of visual words. Each cluster correspond to a visual word
4. Representation of an image as a histogram of visual words

Here, we will use for the steps 1 and 2 the SIFT descriptors for RGB-D images introduced in the above section. So, each region of interest will be described by a 256-dimensional vector.

For obtaining the vocabulary or codebook mentioned on step 3, there have been used many cluster algorithms. The k-means [18] algorithm is among the most used methods for this, because it is simple and fast. Also, other clustering algorithms (like k-nearest neighbor [19]) as well as other strategies for handling quantization (e.g., term weights, soft assignment, non-uniform distributions) have been explored. The construction of the codebook is a key step in the bag of feature approach. Notice that the vocabulary play a crucial role in the description of images. The construction of the codebook is not an easy task. For example, when k-means is used, it is necessary to specify the number of clusters k (as in many other cluster algorithms), but how to know beforehand how many words (clusters) the codebook has? Also, k-means is extremely sensitive to the initial setting. In this way, for any cluster algorithm choose on this step, we could found some dangerous drawbacks. For that reason our idea here is to obtain the codebook or vocabulary by using Cluster Ensemble methods [20, 21].

Cluster ensemble consist of generating a set of clustering from the same dataset and combining them into a final clustering. This idea of combining different clustering results (*clustering ensemble* or *clustering aggregation*) emerged

as an alternative approach for improving the quality of the results of clustering algorithms. Given a set of objects, a cluster ensemble method consists of two principal steps: *Generation*, which is about the creation of a set of partitions of these objects, called ensemble and *Consensus*, where a new partition (consensus partition), which is the integration of all partitions obtained in the generation step, is computed. Figure 8 shows a graphical representation of the cluster ensembles process.

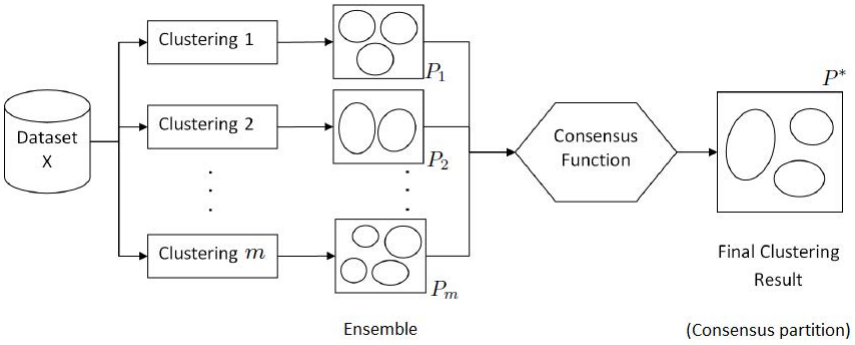

Figure 8: The basic process of cluster ensembles.

The cluster ensemble approach used here can be described through two basic steps:

i) Construction of the ensemble: At this step, several cluster methods (or the same method with different parameters) are run in order to obtain different partitions of the whole space, say $P_1, P_2, ..., P_m$. For example, we can run k-means with different values of k, hierarchical clusters with different linkage methods, etc.

ii) Combine the multiple partitions to obtain the consensus partition P^*: At this step, the following optimization problem is solved:

$$P^* = \min_P \sum_{i=1}^{m} \Gamma(P, P_i) \tag{1}$$

where Γ is a suitable dissimilarity measure (or distance) between partitions. Due to its known structural properties, we will use for comparing partitions the Mirkin distance [20], defined as:

$$\Gamma(P_i, P_j) = \sum_k n_k^2 + \sum_{k'} n_{k'}^2 - 2 \sum_k \sum_{k'} n_{kk'}^2 \tag{2}$$

where n_k is the number of elements in the $k-th$ cluster of partition P_i, $n_{k'}$ is the number of elements in the $k'-th$ cluster of partition P_j and $n_{kk'}$ is the number of common elements in clusters k and k'.

The optimization problem will be solved using the simulated annealing method. This method models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy. The general idea of the simulated annealing algorithm can be described as follows. At each iteration of the algorithm, a new point is randomly generated. The distance of the new point from the current point, or the extent of the search, is based on a probability distribution with a scale proportional to the temperature. The algorithm accepts all new points that lower the objective, but also, with a certain probability, points that raise the objective. By accepting points that raise the objective, the algorithm avoids being trapped in local minima, and is able to explore globally for more possible solutions. The algorithm systematically lowers the temperature (annealing schedule), storing the best point found so far. The temperature is a control parameter that determines the probability of accepting a worse solution at any step and is used to limit the extent of the search. As the temperature decreases, the algorithm reduces the extent of its search to converge to a minimum. The algorithm stops when the average change in the objective function is very small, or when any other stopping criteria are met.

To solve our particular optimization problem, we need to specify some parameters to the simulated annealing algorithm. The first one is an initial state, in our case an initial partition. This could be done just by selecting randomly a partition from the ensemble or by choosing the partition in the ensemble that minimizes the objective function. But, it is known that giving a good starting point in this kind of algorithms help the optimization process, so a better initial point was proposed. It can be theoretically demonstrate that all pairs of elements that are in the same cluster for more than a half of partitions in the ensemble, will also be in the same cluster in the consensus partition [22]. So, we will use this criterium to generate the initial partition P_0. Another parameter to be specify is the way in which new points will be randomly generate, also known as neighborhood generation function. Given a current solution or state (i.e., a partition), we will defined as a neighbor partition, the partition obtained by merging two randomly selected clusters. The objective function was defined in Equation 1 and the annealing schedule used was $T = T_0 * 0.95^i$, where T_0 is the initial temperature and i the iteration number.

Once having a final clustering result P^*, we need to select a prototype for each cluster which will conform the codebook. We have choose as prototypes the mean of each cluster.

4 Image representation and comparison

The codebook or visual words are used to represent images. Given an RGB-D image, the SIFT descriptors introduced in section 2 are extracted. Those

descriptors are mapped into visual words, and the image is represented as a "bag of visual words". More specifically, images are represented by histograms, that means a vector containing the occurrence of each visual word in the image.

This vector is used as feature vector in the classification task. In this work we have used Support vector machines (SVM) [23] for classification. The SVM can be used with any of the usual kernel functions like linear kernel, Gaussian kernel, polynomial kernel, among others. As we are working with histograms, a more suitable kernel to use can be any kernel design for histograms, like the histogram intersection kernel [24] or the χ^2 kernel [25]. In this work we have introduced a new kernel function for histograms which takes into account the intrinsic structure of this kind of data called weighted kernels for histograms.

Before introducing the weighted kernels for histograms we will present some auxiliary definitions. Let X be an ordered set of m disjoint categories and $H(X)$ the set of all histograms of X.

Definition 1. *(Positive weight function) A function $\omega : H(X) \times H(X) \to \mathbb{R}$ is said to be a positive weight function if for all $h, h' \in H(X)$ the following conditions hold: (i) $0 < \omega(h, h') \leq 1$, and (ii) $\omega(h, h) = 1$.*

Observe that a positive weight function takes higher values as the histograms being compared become more similar.

Definition 2. *(Negative weight function) A function $\nu : H(X) \times H(X) \to \mathbb{R}$ is said to be a negative weight function if for all $h, h' \in H(X)$ the following conditions are satisfied: (i) $0 \leq \nu(h, h') < 1$, and (ii) $\nu(h, h) = 0$.*

Contrary to the positive, the negative weight function takes higher values when the histograms being compared are less similar.

Now, we are able to define the family of weighted kernels.

Definition 3. *(Weighted kernels for histograms) Let H be a set of histograms. A weighted kernel to comparing histograms is a function $k_{\omega\nu} : H \times H \to \mathbb{R}$ given by:*

$$\begin{aligned} k_{\omega\nu}(h, h') &= \sum_i (\omega_i(h, h')(\delta_i(h)\delta_i(h') + \lambda_i(h)\lambda_i(h')) \\ &\quad -\nu_i(h, h')(\delta_i(h)\lambda_i(h') + \lambda_i(h)\delta_i(h'))), \end{aligned} \tag{3}$$

where ω_i and ν_i are positive and negative weight functions on $H(X)$, respectively and $\delta_i(h)$ and $\lambda_i(h)$ are functions defined as:

$$\delta_i(h) = \begin{cases} 1, & h(i) < h(i+1); \\ 0, & \textit{otherwise.} \end{cases} \qquad \lambda_i(h) = \begin{cases} 1, & h(i) > h(i+1); \\ 0, & \textit{otherwise.} \end{cases} \tag{4}$$

This means that $k_{\omega\nu}$ computes the similarity between two histograms h and h' by checking for common peaks in both histograms. At the i-th bin, if h and h' have the same behavior with respect to the previous bin (e.g., in both histograms the i-th bin is greater (or smaller) than the $(i-1)$-th bin)

then, $k_{\omega\nu}$ adds a positive weight $\omega_i(h, h')$; otherwise $k_{\omega\nu}$ adds a negative weight $\nu_i(h, h')$. That's the reason why high positive weights and low negative weights takes place when histograms are similar and vice versa (This fact explains our definitions of weighted functions). In this way $k_{\omega\nu}$ is taking into account both, the distribution of the peaks and the information at each bin for the comparison of histograms.

The following Lemma stay that the similarity measure introduced in equation (3) is a true kernel. A rigorous proof of this will be given upon request.

Lemma 1. *Any measure $k_{\omega\nu}$ as in (3) is a semi-definite positive kernel.*

5 Experimental results

5.1 Description of the database

We have used for the experiments the RGB-D Object dataset [11]. This is a large dataset consisting of cropped and segmented images of 300 objects, grouped into 51 categories. There are between three to twelve instances in each category. The images are collected with an RGB-D camera that can simultaneously record both color image and depth data at 640×480 resolution. Each object is recorded from three viewing heights (30^o, 40^o and 60^o above the horizon) while it rotates on a turntable.

In our experiments, we use the same setup as in [11], distinguishing between category and instance recognition. In the experiments done so far, we have only test the category level recognition. In category level recognition, we randomly leave one object out for each category for testing and train the classifiers on all views of the remaining objects. We used around 30 views at each height, giving around 90 views per instance or 24000 RGB-D images.

5.2 Category recognition

We first compute the SIFT features introduced in section 2 for all the images. This gives around one million of features. We randomly selected a subset of 300000 features to build the codebook. This 300000 features conform the training set used for cluster ensemble. Note that, even selecting the third of all the extracted features, the number of data points is large. This amount of data demand cluster algorithms that do not require to store the square inter-distance matrix (due to memory capacity) or cluster algorithms designed for large datasets.

The different partitions were obtained running:

- k-means with the following values of k = 300, 400, 500, 600, 800, 1000, 1500, 2000. Also the k-means algorithm was run several times (with random initialization) and the better result was selected. The number of repetition was set to 20. The Euclidean distance was used inside k-means.

- hierarchical cluster with different linkage methods, specifically: Ward linkage, Median linkage and Centroid linkage. The Euclidean distance was also used inside the algorithm.

We obtained a total of 28 partitions, which means 28 different clustering results. With those partitions in the ensemble we computed the initial partition given to the simulated annealing algorithm. The proposed algorithm for obtaining the initial partition also scale with the number of data points. For that reason, obtaining the different cluster results and the initial partition takes a long time. The simulated annealing was executed in order to find the consensus partition. This optimization procedure also takes a long time. The consensus partition results in 2000 clusters, so the codebook or vocabulary obtained have 2000 visual words.

All images were represented by histograms using the obtained vocabulary. The division on train and test set was done as establish in [11] and a Support vector machines classifier was trained. The results reached working will all the 51 categories were not good. After some analysis, we find that the problem presented in the modified SIFT features (i.e., extracting very few keypoints in some kind of images) is influencing a lot the results, because there are some categories for which the number of extracted features is dramatically small. We decided then, to removed those categories and made the experiments with the remaining ones that do not suffer so much from this problem.

The experiments were done with 12 categories and the classification performance obtained was 93%. Of course, with this simplification, it is difficult to draw conclusions and comparisons with the state-of-the-art algorithms can't be done. But, at least, on the tested categories, the preliminary results are aceptable. The classification accuracy reached using only the standard SIFT descriptors was 77%.

6 Conclusions and future work

This report summarizes the preliminary results achieved in an investigation that is in its beginning. Thus, rather than concluding the work, we will focus on the issues that must be improved as well as the lines of future work for improving the results.

This investigation has focused on new methods for object recognition using RGB-D images. A modification to the SIFT algorithm for RGB-D images was proposed. Some of its drawbacks were discussed in section 2 and they constitute lines of future work. Specifically, we intend to improve the SIFT algorithm by making more use of the geometric information given by the Kinect sensor. Besides, we intend to evaluate the use of other geometric descriptors like the spin images [6].

The Bag of Features approach was used for object recognition. We proposed the use of cluster ensemble for the construction of the codebook. This is an advantageous solution over using any particular cluster algorithm. However, there are several aspects of this methodology worth of further study. We need

to explore cluster algorithms prepared for large datasets. This is crucial for obtaining good members in the ensemble. Different distances can be used inside those cluster algorithms (not only the Euclidean distance). Once having the consensus partition, variants more robust than just taking the mean should be studied for the selection of prototypes. Also information about spatial relation can be incorporated in the Bag of Features approach. In the bag of words model, the visual words are seen as independent and orderless. Different approaches have been introduced to incorporated spatial relation (e.g., represent an image as a histogram of pairs of visual words which co-occur within a local spatial neighborhood). This constitute another line of future work in our investigation.

The experiments must be repeated including all categories in the classification and an exhaustive comparisons with the state-of-the-art algorithms. Also it is necessary to design experiments that allows to evaluate the influence that each stage (i.e., feature extraction, bag of visual model, classifier) is having in the final results. Instance recognition will be also included as well as object recognition from video which is the final goal of this research.

References

[1] Microsoft Kinect. http://www.xbox.com/en-us/kinect.

[2] B. Steder, G. Grisetti, M. Van Loock and W. Burgard. Robust on-line model-based object detection from range images. In *IEEE Intl. Proc. on Intelligent Robots and Systems (IROS)*, 2009.

[3] B. Steder, G. Grisetti and W. Burgard. Robust place recognition for 3D range data based on point features. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2010.

[4] R. Rusu, Z. Marton, N. Blodow and M. Beetz. Persistent point feature histograms for 3d point clouds. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS-10)*, 2008.

[5] R. Rusu, N. Blodow and M. Beetz. Fast point feature histograms (fpfh) for 3D registration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2009.

[6] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 21, 433-449, 1999.

[7] B. Steder, R. Rusu, K. Konolige and W. Burgard. Point feature extraction on 3d range scans taking into account object boundaries. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91-110, 2004.

[9] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding* 110, 346-359, 2008.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[11] K.Lai, L. Bo, X. Ren and D. Fox. A large-scale hierarchical multiview RGB-D object dataset. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2011.

[12] L. Bo and C. Sminchisescu. Efficient Match Kernel between Sets of Features for Visual Recognition. In *Advances in Neural Information Processing Systems*, 2009.

[13] L. Bo, X. Ren, and D. Fox. Kernel Descriptors for Visual Recognition. In *Advances in Neural Information Processing Systems*, 2010.

[14] L. Bo, K. Lai, X. Ren and D. Fox. Object Recognition with Hierarchical Kernel Descriptors. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011.

[15] L. Bo, X. Ren and D. Fox. Depth Kernel Descriptors for Object Recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[16] K.Lai, L. Bo, X. Ren and D. Fox. Sparse Distance Learning for Object Recognition Combining RGB and Depth Information. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2011.

[17] M. Blum, J. Springenberg, J. Wulfing and M. Riedmiller. On the Applicability of Unsupervised Feature Learning for Object Recognition in RGB-D Data. In

[18] I. Dhillon, Y. Guan and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. *KDD*, 551-556, 2004.

[19] R. Duda, P. Hart, D. Stork. Pattern Classification. *Pattern Classification and Scene Analysis: Pattern Classification*, Wiley, 2001

[20] P. Barthlemy and B. Leclerc. The median procedure for partitions. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 19, 3-34, 1995.

[21] S. Vega-Pons, J. Correa-Morris and J. Ruiz-Shulcloper. Weighted partition consensus via kernels. *Pattern Recognition* 43, 2712-2724, 2010.

[22] P. Barthlemy and B. Leclerc. The median procedure for partitions. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 19, 3-34, 1995.

[23] N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. John Shawe-Taylor, Royal Holloway, University of London, 2000.

[24] K. Grauman and T. Darrel. The pyramid match kernel: discriminative classification with sets of image features. *In ICCV*, 2005.

[25] J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid. Local features and hernels for classification of texture and object categories: A comprenhensive study. *IJCV* 73, 213-238, 2007.