Object recognition from a large set of visual features and 3D range data

P. Espinace, D. Mery and A. Soto
Computer Science Department
Pontificia Universidad Católica de Chile
Santiago, Chile
Email: [pespinac,dmery,asoto]@ing.puc.cl

***Abstract*—In Mobile Robotics, recognizing different objects in a given environment is a key requirement for several high level tasks, such as scene understanding, object manipulation, and human-robot interaction. To date, several object recognition approaches have been developed, however, most state-of-the-art methods do not take advantage of a huge pool of visual features and the embedding of the objects in 3-D space. In this paper, we present our approach to build category-level object detectors that optimize the use of the available information by using a two steps procedure: i) Offline training to extract a very large number of visual features from a given training set and to select the most relevant ones to build a particular object model, and ii) Online object detection to use the resulting object model in order to search for objects in images acquired by a mobile robot. Our detectors use a standard AdaBoost method and, similar to previous work, a cascade of weak classifiers is used to focus on discriminative features and quickly discard unlikely image areas. In order to take advantage of the three-dimensional embedding of the searched objects, we augment the visual features with physical properties such as object size, height and internal disparity, acquired using a 3D range sensor on the robot and standard projective geometry. Results show that our method is able to optimize the used features, achieving a performance that is comparable to state of the art approaches using very few features in each object model, and that adding 3D based physical object properties increases performance with respect to a purely 2D visual approach.**

I. Introduction

The human visual system is extremely robust in recognizing and amazing amount of different objects on several different types of scenes, despite of lightning conditions, view point, occlusion, among many other difficulties. In the case of computer based systems, such as a mobile robot, this task has proven to be extremely challenging. State of the art in this field can be measured in contests such as the *Semantic Robot Vision Challenge*, that poses the challenge of downloading images from the web *on-the-fly*, learning models of the objects, and then driving around the environment finding them. Most work in this competition focuses on filtering downloaded images and finding salient regions of the visual field and environment [1]. Also, the *Pascal Visual Object Classes Challenge* poses the goal of recognizing objects from different classes in natural scenes. Some of the most successful results in previous versions of this contest were achieved by the parts based model of Felzenszwalb et. al. [2], and the two stages detector of Harzallah et. al. [3].

Despite good results achieved so far, most state of the art methods do not use the information available from the environment in a proper way. One of the main reasons for this is that traditionally it is recommended to extract a low number of features in order to avoid high computational cost, which leads to ignoring or poorly using a big amount of features that can potentially be useful. This weakness is shared by most pattern recognition methods [4] [5]. Although it is true that using a small number of features is essential for building efficient online algorithms, using today's computational capabilities we are able to extract a very large number of features in an off-line process in order to investigate which features are really relevant for a given task, and use these features in an online procedure to classify new samples. To formalize this idea, we developed a highly general pattern recognition methodology applied to image analysis, which was presented in [6] with encouraging results. In this work, we apply the developed methodology to build category-level object detectors that choose the best group from a big set of available features, in order to build optimal object models. Our models are augmented by using a pyramid of features, similar to the one used in [7], to obtain global and local object features instead of global features only.

A second weakness of traditional object recognition methods is that most of them rely in two-dimensional image data only, ignoring the embedding of the objects in three-dimensional space. Although there are notable exceptions to this statement, such as the work by Hoiem et. al. [8], the use of 3D information is largely ignored, despite the fact that currently available sensors provide a very reliable depth estimation for a given scene. Considering this, we augment the visual features given by a 2D image with physical properties such as object size, height, and internal disparity, acquired using a 3D range sensor and standard projective geometry. We use these physical properties as priors when searching for different objects within an image, allowing us to increase the performance and efficiency of our method.

Accordingly, the main contributions of this work are: i) Applying the methodology presented in [6] to build category-level object detectors that select a suitable set of discriminative visual features from a huge pool of potential features, and ii) Using 3D information to estimate physical object properties that allows us to use semantic information to boost detector performance.

The rest of this paper is organized as follows. Section II presents our problem formulation, showing how we integrate an object recognition method with geometric based priors. Section III describes our method for building category-level object detectors based on the methodology presented in [6]. Section IV shows how we can use 3D information to compute each of the priors included in our model. Section V shows the results for our approach. Finally, Section VI presents the conclusions for our work and future avenues of research.

II. Problem Formulation

In this section we present our mathematical model to solve the problem of recognizing objects in a single image I, using the information that can be extracted from the image itself and from a depth map D given by a 3D range sensor. The formulation of the problem is based on recognizing objects in a given set and is based on a probabilistic framework.

Considering an image may contain multiple objects, and multiple instances of each particular object, we decompose each image into a set of windows. These windows are assumed to be independent and we can evaluate each of them using a sliding windows procedure at different image scales, thus, our problem is to compute

$$p(w_{i,o}|vi, di) \tag{1}$$

that represents the likelihood that object o is present in window i, given the visual information vi that we can extract from window i of image I, and the depth information di that we can extract from window i of depth map D. Using Bayes rule, we can transform our problem into:

$$p(w_{i,o}|vi, di) = \frac{p(vi|w_{i,o}, di) * p(w_{i,o}|di)}{p(vi|di)} \tag{2}$$

As both information sources are given by independent sensors, the information contained in one of them does not influence the content of the other one, thus, $p(vi|di)$ can be transformed into $p(vi)$, which can be assumed to be a flat prior, and $p(vi|w_{i,o}, di)$ can be transformed into $p(vi|w_{i,o})$, that represents the likelihood of extracting information vi in window i of image I given that object o is present in that window. Section III gives details on how we compute this term by building category-level object detectors for each of the objects in a given set.

The remaining term in our equation, $p(w_{i,o}|di)$, represents the likelihood that object o is present in window i given only the information that can be extracted from window i in depth map D. Here, we could directly use the depth values inside the given window and estimate how likely are those values given that object o is present on it, however, we can also use this depth information to infer higher level properties of the object contained in the window, using the sensor parameters and standard projective geometry, and calculate how likely is that window to contain object o given this high level properties. As we will show later, this kind of properties are much more useful to improve the efficiency and performance of the object detection task. In our case, the properties we estimate are object size s, object height h, and object internal disparity d, all measured in meters. Thus, we transform the term $p(w_{i,o}|di)$ in $p(w_{i,o}|s, h, d)$. Note that s, h and d are estimations for the size, height, and internal disparity not for object o, but for any potential object that would be contained inside window i, thus, we are measuring the likelihood that object o is present in window i given that any object contained in window i should have the calculated values for the used properties.

Using the Bayes rule once again, we can transform $p(w_{i,o}|s, h, d)$ into:

$$p(w_{i,o}|s, h, d) = \frac{p(s|w_{i,o}, h, d) * p(w_{i,o}|h, d)}{p(s|h, d)} \tag{3}$$

Assuming that, once calculated, the properties s, h, and d are independent, we can transform $p(s|h, d)$ into $p(s)$, which we can assume as a flat prior (thus it becomes a constant multiplication term), and transform $p(s|w_{i,o}, h, d)$ into $p(s|w_{i,o})$. Analogously, we can transform $p(w_{i,o}|h, d)$ into:

$$p(w_{i,o}|h, d) = \frac{p(h|w_{i,o}, d) * p(w_{i,o}|d)}{p(h|d)} \tag{4}$$

And we can transform $p(h|d)$ into $p(h)$, assuming a flat prior, and transform $p(h|w_{i,o}, d)$ into $p(h|w_{i,o})$. Finally, we can transform the term $p(w_{i,o}|d)$ into:

$$p(w_{i,o}|d) = \frac{p(d|w_{i,o}) * p(w_{i,o})}{p(d)} \tag{5}$$

And we can assume $p(w_{i,o})$ and $p(d)$ as flat priors. Using all this process we have transformed equation 3 into:

$$p(w_{i,o}|s, h, d) = \alpha * p(s|w_{i,o}) * p(h|w_{i,o}) * p(d|w_{i,o}) \tag{6}$$

Continuing with a similar procedure can turn the probability of $w_{i,o}$ given any number of properties into the multiplication of a constant term times the product of the probabilities of each of the properties given $w_{i,o}$. Each of these terms represent the probability of a certain property to have the estimated value given that object o is present in window i. Section IV shows how we calculate these terms in our method.

Given our analysis, we have transformed our model in equation 1 into:

$$p(w_{i,o}|vi, di = (s, h, d)) = \alpha * p(vi|w_{i,o}) * p(s|w_{i,o}) * ... \\ p(h|w_{i,o}) * p(d|w_{i,o}) \tag{7}$$

Note that the analysis and final model is similar to that presented by Torralba in [9] applied to contextual priming for object detection. In our case, we use priors based on physical object properties, which for the case of a mobile robot can be estimated using a 3D range sensor, however, the methodology is highly general and can incorporate context or any other kind of prior.

Using the proposed model, we can estimate the probability that any object in a given set is present in any of the windows. As we mentioned earlier, a sliding window approach can be used to evaluate every window and find the objects that are present in a given image, however, we should decide when the method says that an object is present according to the calculated probabilities. For this purpose, we compute suitable detection thresholds according to training data, as will be shown later. Additionally, some windows may overlap, producing overlapping detections. In cases where two highly overlapping detections occur, we assume the detection with the higher probability to be the correct one.

III. Category-Level Object Detection

In this section we present our approach to category-level object detection, based in the methodology presented in [6], and show how we compute the term $p(vi|w_{i,o})$ of our model. We base our description on the detection of a generic object o, which can be instantiated to any object in a given set.

The key idea behind the methodology presented in [6] is to use an offline procedure to extract a very large number of features from a training set built for the classes we want to separate, and select only those features that are relevant for the separation of the classes. Then, in an online procedure, we can use only the selected features in order to classify new samples. Our claim is that the more features extracted during this process, the more alternatives a selection method has for building distinctive class models. In our case, we model object detection as a two classes problem: object and background. The background class should represent everything that is not the object we are searching for, thus, it must be very diverse. If several classes are available, representing all possible objects in the environment, we can use a one-vs-all procedure for building each of the object classifiers.

In order to increase the prediction power of the described methodology, we use a pyramid of features, similar to the one used in [7], that allows us to obtain the same base feature for different patches within a single object instance, for different number of partitions. This method not only considerably increases the total number of available features, but also allows distinctive internal parts of an object to gain importance, allowing us to extract global and local image features. We use a three level pyramid, obtaining a total of 21 image patches, as shown in the example of Figure 1 for an instance of the PC Monitor class. For each of the patches in the pyramid, we extract 3 groups of features:

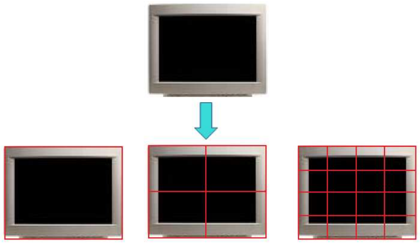


Figure 1. Three levels image pyramid built for extracting features on different patches of an object instance. Example for one instance of the PC Monitor class

1. Grayscale features, providing the mean and standard deviation of the intensity value within patch.
2. Gabor features based on 2D Gabor functions, i.e., Gaussian-shaped bandpass filters, with dyadic treatment of the radial spatial frequency range and multiple orientations, which represent an appropriate choice for tasks requiring simultaneous measurement in both space and frequency domains. We use 8 different scales and 8 different orientations and calculate the mean and standard deviation of the convolved region.
3. Histogram of Oriented Gradients [10], that provide a measure of the magnitude of the gradients of a patch pointing in different directions. We use 4 different number of beans in the histograms.

In total, we use 2 Grayscale based features, 128 Gabor based features, and 66 Histogram of Gradient based features, each of which is calculated for the 21 available patches, obtaining 4116 features for each object instance.

As a selection method, our approach learns different object models using AdaBoost, starting from 4116 weak classifiers (one for each feature), and increasingly adding one weak classifier at a time until a required performance is met for the final strong classifier. Original AdaBoost procedure uses a strong classifier that adds the votes of each weak classifier, which are either -1 or1, and the presence or absence of the object is given by the sign of the final output. Using this method we would obtain a binary output, however, our approach needs a continuous value between zero and one as it is based on a probabilistic framework. Considering this, instead of using the sign, we perform a min-max normalization over the above function output. The minimum and maximum values that we can obtain from the method are:

$$\arg\min(\sum_{t=1}^{T}\alpha_t * h_t(x)) = -\sum_{t=1}^{T}\alpha_t \quad (8)$$

and

$$\arg\max(\sum_{t=1}^{T}\alpha_t * h_t(x)) = \sum_{t=1}^{T}\alpha_t \quad (9)$$

that would be the cases of all weak classifiers voting that the object is not present (every weak classifier output is -1), and that the object is present (every weak classifier output is 1). The min-max normalization value is given by:

$$H2(x) = \frac{\sum_{t=1}^{T}\alpha_t * h_t(x) - (-\sum_{t=1}^{T}\alpha_t)}{\sum_{t=1}^{T}\alpha_t - (-\sum_{t=1}^{T}\alpha_t)} \quad (10)$$

which can be transformed into:

$$H2(x) = \frac{\sum_{t=1}^{T}\alpha_t * h_t(x) + \sum_{t=1}^{T}\alpha_t}{2 * \sum_{t=1}^{T}\alpha_t} \quad (11)$$

The output of this new function has values between zero and one, and is higher if there are more weighted votes supporting the presence of the object and lower if there are less weighted votes supporting the presence of the object, thus, we consider this new function a distribution of the probability of an object instance to belong to a given class. As the new samples we want to classify are the pieces of image inside each of the windows we broke our original image into, and they are represented by the visual information (selected features) we extract from them, the obtained distribution represents the term $p(vi|w_{i,o})$ of our model.

It is important to remark that, while the number of available features is in the order of thousands, all our built classifiers selected less than 100 features, proving the robustness of the methodology presented in [6] in terms of selecting only relevant features for each object model. Also, similarly to other approaches [11], we arrange the voting scheme in a cascade that only uses each further weak classifier if the performance until the previous one is above a threshold chosen during training. This cascade allows us to discard unlikely image places quickly and fits perfectly with further procedures incorporated in our approach, allowing them to increase performance even more, as we will see in section IV. The final decision about the presence of the object is made according to the threshold estimated from training data using all the selected features, if the cascade is able to reach its deeper level.

IV. Geometric Priors

No matter how good an object detector works, in real world applications there are always situations where an object will not be found, either because of point of view, poor lightning conditions, occlusion, etc. For such cases, a possible solution approach would be lowering the detection threshold for the object, however, this carries the cost of an increasing number of false positives. In these situations, using additional sources of information can be crucial for helping in avoiding the new false positives, allowing to lower the threshold for finding more object instances at, ideally, no additional cost. In our approach, we provide such capabilities by using a 3D range finder that provides depth maps associated to the images we already have.

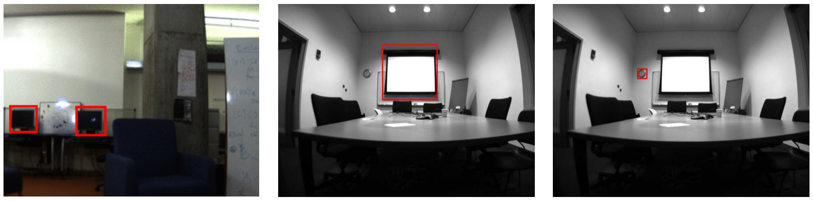
Given an image and its associated depth map, we can use the camera parameters and standard projective geometry in order to calculate how many centimeters does each pixel cover. Knowing this, we estimate three different sources of information about the objects contained in a given window: i) object size, i.e, width and height in centimeters, ii) object height, i.e, distance from the floor to the object in centimeters, and iii) internal disparity, i.e, standard deviation of the distances inside the object in centimeters. Each of these sources of information can provide a prior for the probability of a window containing the object, by evaluating them over Gaussian distributions estimated from training data for each of the objects. Consequently, they provide the geometrical priors $p(s|w_{i,o})$, $p(h|w_{i,o})$, and $p(d|w_{i,o})$ of our model.

The obtained geometrical priors not only increase the performance of our method, as we will show in section V, but also increase the methods computational efficiency by taking advantage of the cascade nature of the resulting classifiers, applying the priors at each level of the cascade. By doing this, several windows that have probability values above a given threshold may be discarded at early stages of the cascade if the priors show that according to size, height, or disparity, this probability is below the threshold.

V. Results

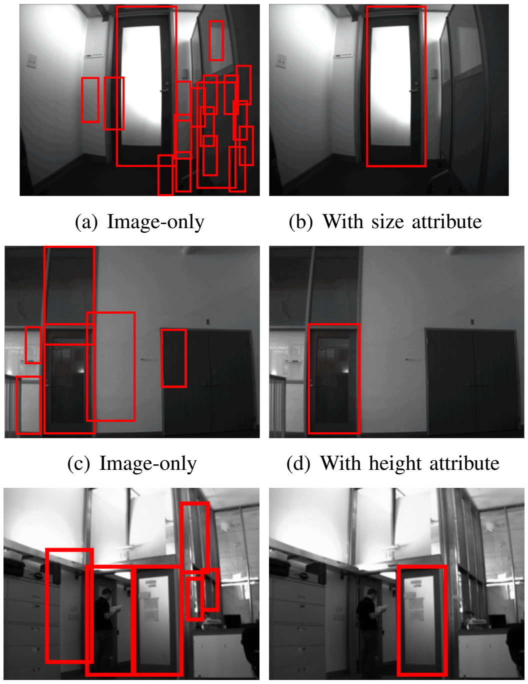
In this section we present experimental results for our method. We made tests with seven different object classes: PC monitor, door, railing, clock, screen, soap dispenser, and urinal. Figure 2 shows detection results for our method with three of these objects. We can see that we can find two instances of the same object, PC monitor, in a single image, big objects such as screen, and small objects such as clock. The screen and clock where both found on the same image by running a single object detector at a time, however, if we run both of them together, both objects would be found.

Figure 3 shows how each physical property of objects helps in the object detection task. In figure 3(a), we see an example image where a door would normally be detected in many places by using only image information, however, because doors usually have a predictable size, the classifier was able to rule out a number of these hypotheses (figure 3(b)) to find the only true positive. Similarly, we see in figure 3(c) and 3(d) that a number of false detections were avoided by using the estimated height prior. Finally, we see



(a) PC Monitor (b) Screen (c) Clock

Figure 2. Object detections for three different objects.



(a) Image-only (b) With size attribute

(c) Image-only (d) With height attribute

(e) Image-only (f) With disparity attribute

Figure 3. In (a) and (b) is an example of the influence of the size prior in the object detection when searching an instance of the door class. In (c) and (d) is an example of the influence of the height prior in the object detection when searching an instance of the door class. In (e) and (f) is an example of the influence of the disparity prior in the object detection when searching an instance of the door class.

in figures 3(e) and 3(f) that false door detections were discarded due to the fact that the depth of doors does not usually have high variance (disparity estimated prior). In addition, since we were using a cascade of classifiers, many windows were discarded at early levels of the cascade because they were unlikely in terms of height, size, or disparity. Thus, we were both able to increase performance using the physical attributes of objects and at the same time decrease the computational demands.

In addition, we evaluated the classification performance of our method using different combinations of the physical properties by running the classifiers over several images corresponding to an office environment dataset. The comparison metric is the area under the Precision-Recall curve for each of the tested objects (screens and clocks were not present in this dataset). The best classification performance is obtained using all physical properties, as can be seen in the following table:

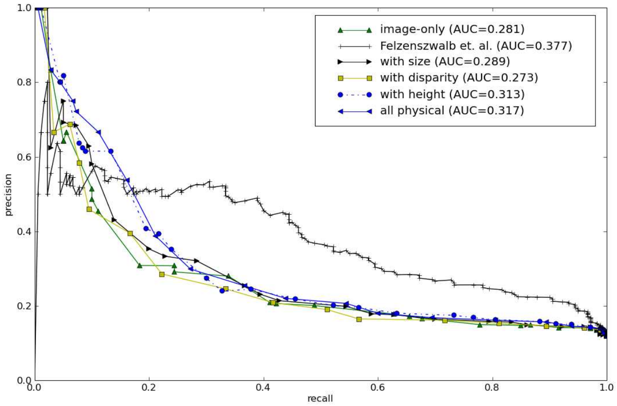


Figure 4. Precision-Recall curve comparison for our method and the method by Felzenszwalb et al.

Area under the P-R curve

Obj.	**Im**	**Ht.**	**Sz.**	**Disp.**	**All**
door	0.474	0.494	0.496	0.469	0.503
railing	0.556	0.648	0.557	0.573	0.671
monitor	0.281	0.313	0.289	0.273	0.317
soap	0.017	0.028	0.018	0.017	0.123
urinal	0.008	0.011	0.011	0.010	0.02

Finally, Figure 4 shows a comparison between our classification performance and the approach of Felzenszwalb et al. [2] for the monitor class. We can see that the area under the curve shows better results for Felzenszwalb approach, however, this fact is greatly influenced by a very wide area where his approach is much better than ours, which are the low threshold areas, while our approach performs better in high threshold areas, which are the one the method selects for our object detection.

VI. Conclusions And Future Work

In this paper, we have shown that, by applying the methodology in [6] and a standard AdaBoost classification procedure, we can build category-level object detectors that have good results for a wide variety of classes. The performance of the method is increased by using physical object properties obtained using a 3D sensor and standard projective geometry.

Our mathematical formulation allows more physical properties or any other source of prior information to be easily included in the model, thus, part of our future research will be focused in including some of these information sources. Particularly, work is already being done for incorporating higher level context, including spatial, temporal and object to object relations that can help in new object detections, such as the ones shown in [12]. Additionally, we are working towards incorporating planning strategies that can help in increasing the efficiency and performance of current object detection. Finally, our goal is to use object detection as a tool for developing higher level robot tasks, such as human robot interaction and office delivery applications.

References

[1] P.-E. Forssen, D. Meger, K. Lai, S. Helmer, J. J. Little, and D. G. Lowe, "Informed visual search: Combining attention and object recognition." 2008, pp. 935–942.

[2] P. Felzenszwalb, D. Mcallester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Anchorage, Alaska, June 2008.*, June 2008. [Online]. Available: http://www.ics.uci.edu/~dramanan/papers/latent.pdf

[3] H. Harzallah, C. Schmid, F. Jurie, and A. Gaidon, "Classification aided two stage localization," oct 2008, pASCAL Visual Object Classes Challenge Workshop, in conjunction with ECCV. [Online]. Available: http://lear.inrialpes.fr/pubs/2008/HSJG08

[4] K. Fukunaga, *Introduction to statistical pattern recognition*, 2nd ed. San Diego: Academic Press Inc., 1990.

[5] A. Webb, *Statistical Pattern Recognition*. England: Wiley, 2005.

[6] D. Mery and A. Soto, "Features: The more the better," in *International Conference on Signal Processing, Computational Geometry and Artificial Vision (ISCGAV)*, 2008.

[7] A. Bosch, A. Zisserman, and X. Muñoz, "Image classification using random forests and ferns," in *IEEE International Conference on Computer Vision*, 2007.

[8] D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006.

[9] A. Torralba, "Contextual priming for object detection," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 169–191, 2003.

[10] T. B. Dalal, N., "Histograms of oriented gradients for human detection," in *European Conference on Computer Vision*, 2005.

[11] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 37–154, 2004.

[12] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things." in *International Conference on Robotics and Automation (ICRA)*, 2009.