Integrating Visual and Range Data for Robotic Object Detection

Stephen Gould, Paul Baumstarck, Morgan Quigley, Andrew Y. Ng, Daphne Koller

Integrating Visual and Range Data for Robotic Object Detection

Stephen Gould* Paul Baumstarck* Morgan Quigley†
Andrew Y. Ng† Daphne Koller†

*Department of Elec. Eng.
Stanford University

†Department of Comp. Sc.
Stanford University

Abstract. The problem of object detection and recognition is a notoriously difficult one, and one that has been the focus of much work in the computer vision and robotics communities. Most work has concentrated on systems that operate purely on visual inputs (i.e., images) and largely ignores other sensor modalities. However, despite the great progress made down this track, the goal of high accuracy object detection for robotic platforms in cluttered real-world environments remains elusive.
Instead of relying on information from the image alone, we present a method that exploits the multiple sensor modalities available on a robotic platform. In particular, our method augments a 2-d object detector with 3-d information from a depth sensor to produce a "multi-modal object detector." We demonstrate our method on a working robotic system and evaluate its performance on a number of common household/office objects.

1 Introduction

Endowing autonomous robots with the ability to perceive objects in their environment is a notoriously difficult problem in computer vision. One standard solution is to equip the robot with a video camera and run state-of-the-art, image-based object detection algorithms on each frame. However, there are a number of difficulties with this approach.

Simply finding each object—let alone classifying it—is a non-trivial task in cluttered real-world environments. Many state-of-the-art object detectors combine the process of object detection (localization) and object recognition (e.g., the sliding-window approach), and only compute features local to the region being considered. While this approach works well for uncluttered scenes with well-framed objects, it fails when applied to real-world images as seen by an autonomous robot.[1]

An autonomous robot needs to operate in a world of small, occluded objects that are often ill-framed. Many visual percepts will not have been encountered before, and portions of the scene may contain confusing or conflicting visual cues when taken out of context. It has been shown that contextual information greatly improves the performance of vision-based object detectors especially when the objects of interest are small [2–4]. However, even when this information is taken into account, the context extracted from image features is inherently 2-dimensional and can be unreliable in natural scenes with significant clutter.

[1] The best performing methods in recent PASCAL Visual Object Classes Challenge [1] (which contains images of difficult natural scenes) achieved less than 50% average precision for most object classes.

Sensor specifications:

- high resolution video camera (704×480 images at 15 frames per second);
- laser range scanner (75 vertical scanlines per second at $0.5°$ resolution);
- laser pan/tilt unit ($37.5°/s$ horizontal pan);
- odometry from navigation laser and wheel encoders.

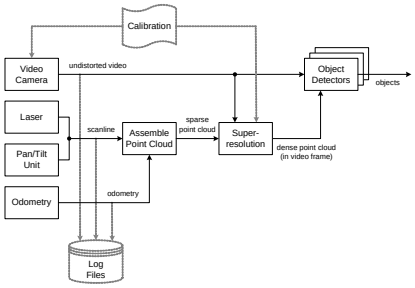


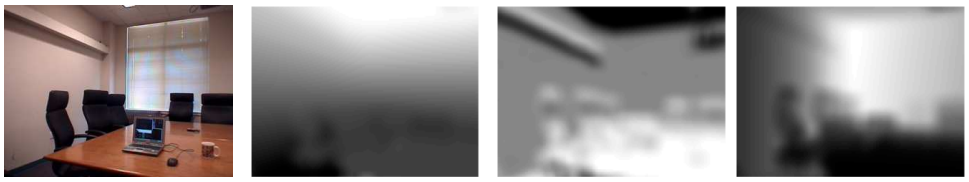
Fig. 1. Our robotic platform, hardware specifications, and data flow: a sparse point cloud is reconstructed in software by knowing the relative pose between the pan/tilt unit and the video camera, and correcting for robot motion using odometry. The depth of every pixel is then inferred, and features from the 2-d image and the 3-d point cloud are combined by our probabilistic object detectors.

Exacerbating the above difficulties are the real-time processing constraints imposed by robotic applications. This, in general, requires that the vision system be simple, and, in particular, rules out the option of going to higher resolution unless a sophisticated attentional mechanism is employed.

The shortcomings of image-based object detection can be addressed on a robotic platform by exploiting the multiple sensor modalities available to the robot rather than relying on a single monocular or stereo-vision sensor mode. Since the robotic system is situated in its environment it can directly measure features that would help detection instead of inferring these quantities from 2-d image cues. For example, measuring temperature can help in the detection of people, and finding large supporting surfaces can help locate small objects. With the availability of cheaper and easier-to-use sensors, such as infrared cameras and lasers, it makes sense to leverage these different sensor modalities for vision.

In this paper, we propose to augment state-of-the-art robotic visual systems with input from a laser depth scanner and to combine the 2-d and 3-d sensor modalities to enhance object detection in cluttered real-world environments. We fuse the depth and visual data so that, for each pixel in the robot's field-of-view, we have color/intensity, depth (location in 3-d space), and surface normal information. This allows us to directly extract contextual cues (such as height above the ground) and 3-d features (such as object size). Figure 2 depicts these features for a simple office scene.

Our method consists of a number of stages which can easily be pipelined for real-time processing on a robotic platform. (See Figure 1 for a schematic of our dataflow). First, using super-resolution techniques, we combine frames from a video camera with a low resolution, or *sparse*, point cloud generated from a laser scanner to produce a high resolution, or *dense*, point cloud in the video camera's frame of reference. Next, we apply a sliding-window object detector at multiple image scales: for each location ((x, y)-position and scale) in the image, we compute local 2-d patch features and corresponding 3-d contextual features. The patch features are provided as input to a trained boosted classifier which, like standard image-only object detectors, estimates the probability of an object being at the location given



(a) Image of scene (b) Height above ground (c) Surface normal (d) Object size

Fig. 2. Example 3-d feature responses for a window size of 40×40 pixels (approximately the size of the coffee mug in this scene). Light areas indicate stronger feature values. For example, the surface normal feature can be used for identifying supporting surfaces such as the table with horizontal support or the walls with vertical support.

just the image features. Finally, we apply a learned logistic classifier for each object to a feature vector comprising this image-only probability *and* the computed 3-d features. A distinct advantage of this approach is that the image-only classifier can be trained using standard state-of-the-art techniques (and on large datasets) while our "multi-modal" classifier still takes advantage of 3-d contextual information.

We demonstrate our method on a number of small household/office objects in natural environments. Importantly, we show that by fusing information from the depth sensor we are able to significantly improve object detection rates over state-of-the-art image-only techniques in cluttered real-world environments.

2 Background and related work

Sliding-window object detection is a popular technique for identifying and localizing objects in an image. The method has been very successfully applied to face detection [5] and can be highly efficient when combined with a cascade of boosted ensembles (CoBE) classifier and simple Haar-wavelet features [6]. However, Haar-wavelet-like features tend to perform poorly on other object classes, and researchers have developed other, more sophisticated discriminative features (e.g., histogram of oriented gradients [7], biologically inspired (visual cortex) features [8], or patch-based features [9]) at the cost of processing speed. When applied to multiple object classes, features can be shared amongst the detectors, thereby amortizing the cost of feature extraction [10].

A number of researchers have shown that context can significantly improve object detection accuracy, especially when the objects are small [2, 3, 9, 11, 4]. However, little work has focused on 3-d context. Notable exceptions are the innovative works of Hoiem et al. [4] and Leibe et al. [12] who infer the camera location and scene geometry from a single 2-d image or stereo video stream, respectively. These works reconstruct rough geometry of street scenes (pedestrians and cars) and cannot, for example, be used for estimating 3-d features of small objects. Instead of trying to infer 3-d structure, we propose to measure it *directly* using a laser.

Some novel works use 3-d point clouds from depth sensors for detecting objects. Nuchter et al. [13] use separate reflectance and depth information from a single sensor to detect known objects. Other works focus on discovering geometric primitives [14], or detecting large novel objects but without recognition [15]. Unlike these works, we detect and recognize small objects from sparse depth information in real-time.

3 Sensor fusion and scene representation

In this section, we describe how we process the raw data from the robot's sensors (Figure 1) into a representation that unifies the modalities of the camera and the sparse depth sensor. In particular, we describe how the sparse laser measurements are used to infer depth at the (much higher) spatial resolution of the camera.

Ideally we would like to measure the depth (and surface normal) at every pixel in the image so that we can make use of 3-d geometric cues in our object detector. Unfortunately, the laser range scanner does not support the same resolution as the video camera.[2] Thus we need to resort to super-resolution techniques in order to *infer* the depth at every pixel. The resulting high resolution depthmap can then be used to estimate surface normals. State-of-the-art methods for super-resolution include MAP inference on a pairwise Markov random field (MRF) [16, 17] and bilateral filtering [18], and are based on the intuition that depth discontinuities usually coincide with color discontinuities in the image.

Our method is similar to the MRF model of Diebel and Thrun [16].[3] However, instead of encoding a preference for fronto-parallel planes (implicit in their formulation) we allow for arbitrarily sloped planar surfaces. Thus our method can be thought of as reconstructing a first-order approximation to each surface rather than a zeroth-order one. We also use a robust Huber penalty instead of the more commonly used ℓ_2 penalty.[4] A quantitative comparison measuring mean-square reconstruction error on a hold-out set of points showed that, on average, our model performed better than that of Diebel and Thrun [16] on our office scenes—details omitted due to space constraints.

Concretely, let the image pixel intensities be $\{x_{i,j} \mid (i,j) \in \mathcal{I}\}$, the laser depth measurements be $\{z_{i,j} \mid (i,j) \in \mathcal{L}\}$ and the reconstructed/inferred depth for every pixel be $\{y_{i,j} \mid (i,j) \in \mathcal{I}\}$ where $\mathcal{I}$ indexes the image pixels and $\mathcal{L} \subseteq \mathcal{I}$ indexes the laser measurements (projected onto the image plane). Two MRF potential functions are defined—the first penalizes discrepancy between measured and reconstructed depths, while the second encodes a preference for smoothness:

$$\Phi_{ij}(\mathbf{y}, \mathbf{z}) = h(y_{i,j} - z_{i,j}; \lambda) \tag{1}$$

$$\Psi_{ij}(\mathbf{x}, \mathbf{y}) = w^v_{ij} h(2y_{i,j} - y_{i,j-1} - y_{i,j+1}; \lambda) + w^h_{ij} h(2y_{i,j} - y_{i-1,j} - y_{i+1,j}; \lambda) \tag{2}$$

where $h(x; \lambda)$ is the Huber penalty function, and $w^v_{ij} = \exp\{-c\|x_{i,j-1} - x_{i,j+1}\|^2\}$ and $w^h_{ij} = \exp\{-c\|x_{i-1,j} - x_{i+1,j}\|^2\}$ are weighting factors indicating how unwilling we are to allow smoothing to occur across vertical and horizontal edges in the image as in [16].

[2] The video camera on our robot has a base resolution of 0.1° and can optically zoom down to 0.01°. On the other hand, modern (full scene scanning) depth sensors, such as the SwissRanger SR-3000, only have a resolution of approximately 0.5°.

[3] Note that in their setup the camera and laser were axis-aligned and the scene imaged by rotating through 360°. Thus they did not have to deal with calibration issues between the two sensors nor occlusions.

[4] The Huber penalty function, $h(x; \lambda) = x^2$ for $-\lambda \leq x \leq \lambda$ and $\lambda(2|x| - \lambda)$ otherwise, is convex and thus we can still find the MAP solution exactly.

(a) (b) (c)

Fig. 3. Results from our super-resolution MRF on a typical office scene (a). Shown are (b) initial interpolated depth estimates; and (c) final depth estimates at convergence (526 iterations).

(a) (b) (c)

Fig. 4. 3-d views of (a) the sparse point cloud from laser scanlines, (b) reconstructed (dense) point cloud, and (c) reconstructed surface normals (color represents normal direction).

We can now define our super-resolution MRF as

$$p(\mathbf{y} \mid \mathbf{x}, \mathbf{z}) = \frac{1}{\eta(\mathbf{x}, \mathbf{z})} \exp\left\{ -k \sum_{(i,j)\in\mathcal{L}} \Phi_{ij} - \sum_{(i,j)\in\mathcal{I}} \Psi_{ij} \right\} \tag{3}$$

where k specifies the trade-off between measurement reconstruction and smoothness, and $\eta(\mathbf{x}, \mathbf{z})$ is the normalization constant (partition function).

We initialize our depth estimates $\mathbf{y}$ by first projecting measured depths into the camera plane and performing quadratic interpolation to obtain an initial depth estimate at every pixel. We then find the MAP solution by minimizing $k \sum_{(i,j)\in\mathcal{L}} \Phi_{ij} + \sum_{(i,j)\in\mathcal{I}} \Psi_{ij}$ using the L-BFGS algorithm [19]. The process is illustrated in Figure 3. Optimization is fairly quick, taking less than 10 seconds on a 320×240 image. Most of the progress towards the optimal solution is made in the first few iterations and high quality depth estimates can still be obtained by stopping early. Furthermore, in a robotic application where data is continuously being streamed, the algorithm can be initialized from the previous frame (with motion compensation) resulting in much faster convergence.

Once we have the high resolution depthmap we reconstruct the location of each pixel in 3-d space (relative to the camera's reference frame) $\{\hat{X}_{i,j} \in \mathbb{R}^3 \mid (i,j) \in \mathcal{I}\}$ by projecting a ray through the camera plane at each pixel location (i,j) and scaling by the inferred depth. We also infer point normals, $\{\hat{n}_{i,j} \in \mathbb{R}^3 \mid (i,j) \in \mathcal{I}, \|\hat{n}_{i,j}\| = 1\}$, from the local neighborhood around each point.[5] Figure 4 shows sample results from our super-resolution procedure. The tuple $(x_{i,j}, \hat{X}_{i,j}, \hat{n}_{i,j}) \in \mathbb{R}^7$ forms the basis of our 3-d scene representation.

4 Multi-modal object detection

In this section, we describe how we use the unified scene representation of Section 3 and a sliding-window approach to perform multi-modal object detection. Specifically, we discuss 2-d and 3-d feature extraction, and describe our method for combining them into a single probabilistic model.

[5] Here we compute the SVD of the 3×3 covariance matrix of points $\hat{X}_{i'j'}$ near to the point in question $\hat{X}_{ij}$ and take the surface normal estimate $\hat{n}_{ij}$ to be the direction of the singular vector corresponding to the smallest singular value. We resolve ambiguity in the sense of the normal vector by taking the solution that points towards the camera.

4.1 Image-only object detectors

Sliding-window object detection is a simple yet effective approach to simultaneous localization and recognition of objects in images. The approach involves scanning the image with a fixed-size rectangular window and applying a classifier to the sub-image defined by the window. The classifier extracts image features from within the window (sub-image) and returns the probability that the window (tightly) bounds a particular object. The process is repeated on successively scaled copies of the image so that objects can be detected at any size. The (x, y)-location of the window and scale σ of the image implicitly defines the bounding box, or set of pixels, belonging to the candidate object $\mathcal{B}(x, y, \sigma) = \{(i, j)\} \subseteq \mathcal{I}$.

Our image features are similar to Torralba et al. [10] (and other works by the same author). Here a dictionary of image patches is constructed at training time by randomly selecting small patches from a set of training images. Associated with each patch is a spatial mask over which the patch response is valid. The mask is derived by taking a small rectangular region around the patch's original (image) location (see Figure 5(d)). In addition to patches taken from the (intensity) image, we also extract patches from the edge-filtered (gradient) image.[6] Thus our dictionary is comprised of the triplet of patch, spatial mask, and image operator (intensity or gradient). Figure 5 shows example training images and a patch dictionary.

We scan the image at multiple locations and scales. The response for feature f at location (x, y) and scale σ is given by $v^f(x, y, \sigma) = \max_{w_f} |T_f(\mathcal{I}_\sigma) \otimes g_f|$ where $\mathcal{I}_\sigma$ is the image at scale σ, $T_f(\cdot)$ is the image transform associated with feature f (intensity or gradient), g_f is the image patch, w_f is the spatial mask, and $\otimes$ is the normalized cross-correlation operator.

Given the dictionary of patch features and a set of training images we learn a gentle-boost classifier [20] over two-split decision stumps for each object class. At run time, we apply the boosted classifier to each candidate location (x, y) and scale σ to obtain the probability (based on image features alone) of that location containing the given object, $P^{\text{image}}(o \mid x, y, \sigma)$. We then use the log-odds ratio

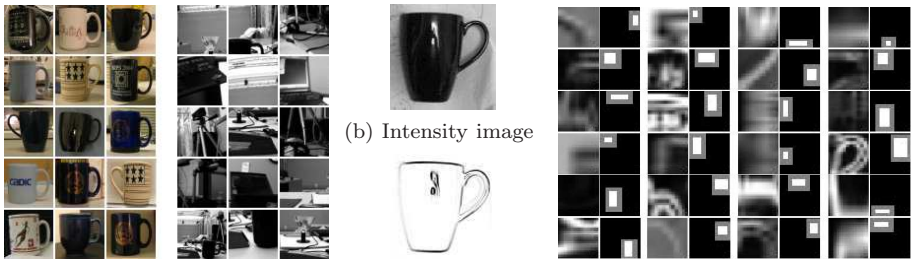
$$f^{\text{2d}}(x, y, \sigma) = \log \left(\frac{P^{\text{image}}(o \mid x, y, \sigma)}{1 - P^{\text{image}}(o \mid x, y, \sigma)} \right) \tag{4}$$

as a feature in our multi-modal object detector.

Note that these detectors do not make explicit use of any 3-d information and therefore can be trained and evaluated using standard state-of-the-art techniques. Briefly, we use 100–200 positive and 20,000 negative training examples for each object class. The positive examples were downloaded from the web or manually collected using a digital camera. The negative examples were collected by randomly snipping rectangles from a five-minute video sequence. All examples were scaled to 32 pixels for the smaller dimension. (See Figure 5.)

We construct our patch dictionary by randomly selecting 10 (intensity and gradient) patches from each positive training sample. The patches vary in size from 4×4 to 16×16 pixels, and we fix the spatial mask w_f to 7×7. We then train the object detectors in two stages for improved efficiency: first we select 2,000

[6] We convolve the intensity image with 3×3 Sobel kernels to obtain horizontal and vertical gradient images $G_h(x, y)$ and $G_v(x, y)$. The edge-filtered image is then given by the gradient magnitude $\mathcal{I}^{\text{edge}}(x, y) = \sqrt{G_h(x, y)^2 + G_v(x, y)^2}$.



(a) Example training images (b) Intensity image (c) Gradient image (d) Example dictionary entries

Fig. 5. Example positive and negative training examples for the "mug" class are shown in (a). An example dictionary for the class "mug" is shown in (d). Adjacent columns represent patch g_f and spatial mask w_f for each dictionary entry. White in the spatial mask indicates size of patch, and gray boundary indicates valid response area.

negative training examples at random and train a boosted classifier for 50 rounds; next, we trim our patch dictionary to remove all patches not used by this classifier; finally, we retrain using all 20,000 negative training examples to obtain our final image-only object detectors. The resulting patch dictionary typically contains 50–75 entries compared to the 500 entries used in [10].

4.2 3-d features

For each candidate location (x, y) and image scale σ, we compute 3-d features by taking the bounding rectangle implicitly defined by the location and scale $\mathcal{B}(x, y, \sigma) = \{(i, j)\} \subseteq \mathcal{I}$ and conceptually projecting that region into the scene as shown in Figure 6. We shrink the set of points enclosed by the bounding box by removing the (i, j) corresponding to points $\hat{X}_{i,j}$ that lie in the outer 5% (in either the x-, y-, or z-directions) of the points in $\mathcal{B}$. This removes outliers and most background points. The resulting shrunken set of points $\mathcal{B}'$ is then used for computing the features.

Let $\{\hat{X}_{i,j} \in \mathbb{R}^3 \mid (i, j) \in \mathcal{I}\}$ be the location of each pixel in 3-d space (relative to the camera's frame of reference), and let $\{\hat{n}_{i,j} \in \mathbb{R}^3 \mid (i, j) \in \mathcal{I}, \|\hat{n}_{i,j}\| = 1\}$ be the estimated surface normal vector for each pixel. The centroid and covariance of the points in $\mathcal{B}'$ are

$$\mu_X = \frac{1}{|\mathcal{B}'|} \sum_{(i,j)\in\mathcal{B}'} \hat{X}_{i,j} \quad \text{and} \quad \Sigma_X = \frac{1}{|\mathcal{B}'|} \sum_{(i,j)\in\mathcal{B}'} (\hat{X}_{i,j} - \mu_X)(\hat{X}_{i,j} - \mu_X)^T \qquad (5)$$

where $|\mathcal{B}'|$ is the number of points in $\mathcal{B}'$. A similar computation gives μ_n and Σ_n, the mean and covariance over surface normal vectors.

We now enumerate the object attributes and contextual cues captured by our 3-d features and provide some insight into why they are useful for improving object detection. Implementation details for how we capture each attribute/contextual cue as a vector-value feature are also described.

Height above ground. Many objects are located at consistent heights above the floor. For example, computer monitors are often found on desks which are a standard height above the floor; door handles are placed at a relatively fixed

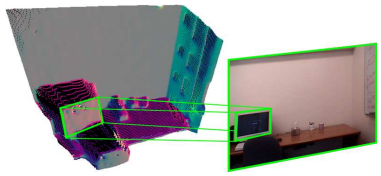


Fig. 6. Illustration of a candidate object location in the image plane being projected into 3-d space.

location on the door where most people find it comfortable to reach; and wall clocks are placed high enough so that they can be seen above other objects in the room. Since we calibrate the point cloud xz-plane to coincide with the floor, our height above ground feature is simply

$$f^{\text{height-above-ground}} = \left[\mu_X^y \quad (\mu_X^y)^2 \right]^T \tag{6}$$

where μ_X^y indicates the y-component of μ_X.

Distance from robot. Objects farther from the robot are harder to see, not only because of their diminished size but also because of lighting effects and depth measurement inaccuracy. Although the probability of an object does not depend its distance from the robot, knowing this distance allows our model to compensate for the influence the above effects have on other features (e.g., the 2-d detectors):

$$f^{\text{distance-from-robot}} = \left[\mu_X^z \quad (\mu_X^z)^2 \right]^T \tag{7}$$

where μ_X^z indicates the z-component of μ_X.

Surface variation and orientation. Most objects have significant surface variation and extend above their supporting surface. Obvious exceptions are flat objects, such as LCD monitors, which appear as a single plane in the 3-d point cloud and so are distinguished by their lack of surface variation. We compute the following vector-valued features to capture these attributes:

$$f^{\text{surf-var}} = \begin{bmatrix} \Sigma_X^{xx} + \Sigma_X^{zz} \\ \Sigma_X^{yy} \\ \Sigma_n^{xx} + \Sigma_n^{zz} \\ \Sigma_n^{yy} \end{bmatrix}, \qquad f^{\text{surf-orientation}} = \begin{bmatrix} \mu_n^y \\ \sqrt{1 - (\mu_n^y)^2} \\ (\mu_n^y)^2 \end{bmatrix} \tag{8}$$

where Σ_X and Σ_n are the covariance matrices defined above.

Object dimensions. One of the most defining attributes of an object is its size. Knowing this information allows a significant number of false candidates to be rejected. The width, height, and (projected) area of the object can be estimated by considering the projection of the bounding rectangle into 3-d space:

$$f^{\text{width}} = \max_{(i,j)\in\mathcal{B}'} \hat{X}_{i,j}^x - \min_{(i,j)\in\mathcal{B}'} \hat{X}_{i,j}^x \tag{9}$$

$$f^{\text{height}} = \max_{(i,j)\in\mathcal{B}'} \hat{X}_{i,j}^y - \min_{(i,j)\in\mathcal{B}'} \hat{X}_{i,j}^y \tag{10}$$

$$f^{\text{area}} = f^{\text{width}} \times f^{\text{height}} \tag{11}$$

where $\hat{X}_{i,j}^x$ and $\hat{X}_{i,j}^y$ denote the x- and y-components of $\hat{X}_{i,j}$, respectively. Our feature vector also includes the square of these terms. A measure of object depth is already captured by our surface variation feature and so is not repeated here.

The above features are assembled into a single 17-dimensional descriptor f^{3d} which captures the 3-d attributes of a possible object.

4.3 Multi-modal object detectors

Our multi-modal object detectors are simple binary logistic classifiers based on the 2-d and 3-d features, $f^{2\mathrm{d}}$ and $f^{3\mathrm{d}}$, defined above. Note that, although this is a simple representation, the inclusion of squared terms for many of the features allows us to learn rich decision boundaries.

The probability for an object o existing at location (x, y) and scale σ in the image plane can be written as

$$P(o \mid x, y, \sigma) = q\left(\theta_{3d}^T f^{3\mathrm{d}} + \theta_{2d}^T f^{2\mathrm{d}} + \theta_{\mathrm{bias}}\right) \tag{12}$$

where $q(s) = \frac{1}{1+e^{-s}}$ is the logistic function and $\boldsymbol{\theta} = \{\theta_{3\mathrm{d}}, \theta_{2\mathrm{d}}, \theta_{\mathrm{bias}}\}$ are the learned parameters of the model. The parameters $\theta_{2\mathrm{d}}$ and $\theta_{3\mathrm{d}}$ trade-off between 2d and 3d features while the bias term θ_{bias} models the prior prevalence of the object.

We learn one multi-modal model per object class on data (images and corresponding point clouds) collected from static scenes. We do this rather than using video sequences to avoid the introduction of bias during training and evaluation due to the high correlation between consecutive video frames. The images are annotated with a bounding box and class label for each object of interest. We construct a training set by first running the image-only object detectors over each image, keeping all detections with a probability above 0.001. Any detection overlapping by more than 50% with a groundtruth annotation is used as a positive example while all other detections are used as negative examples. If for any groundtruth annotation there was no overlapping detection, we further run the image-only object detector on the bounding box for that annotation and add it to our positive examples. Finally, for each training example, we extract the 3-d features as detailed above and learn the parameters $\boldsymbol{\theta}$ of our logistic classifier using Netwon's method so as to maximize the log-likelihood of the training set. We use ℓ_2 regularization to prevent over-fitting, with weight chosen by cross-validation on the training set.

5 Experimental results

We collected 420 static images and corresponding sparse point clouds of cluttered scenes. The scenes contained a number of small objects (coffee mugs, disposable cups, monitors, wall clocks, door handles, and ski boots) which we would like to detect, as well as distractors (see Figure 7). A number of the scenes were extremely challenging and the authors even had trouble identifying objects in some images during groundtruth labeling because of image resolution. We performed k-fold cross-validation and report the aggregate performance over the hold-out sets. On each fold we learned the model parameters and regularization weight using the training set. The data in the hold-out set was only used for testing.

We evaluate our performance by comparing against image-only object detection. Figure 8 shows precision-recall curves for our learned object detectors. The multi-modal detectors (solid red curve) are consistently superior to the image-only detectors (dashed blue curve).

In general, 3-d features significantly help, especially when strong features such as size and location overcome large intra-class appearance variation (door handles and ski boots) or lack of discriminating visual features (computer monitors). The cup class performs badly for both detectors primarily due to its lack of distinguishing

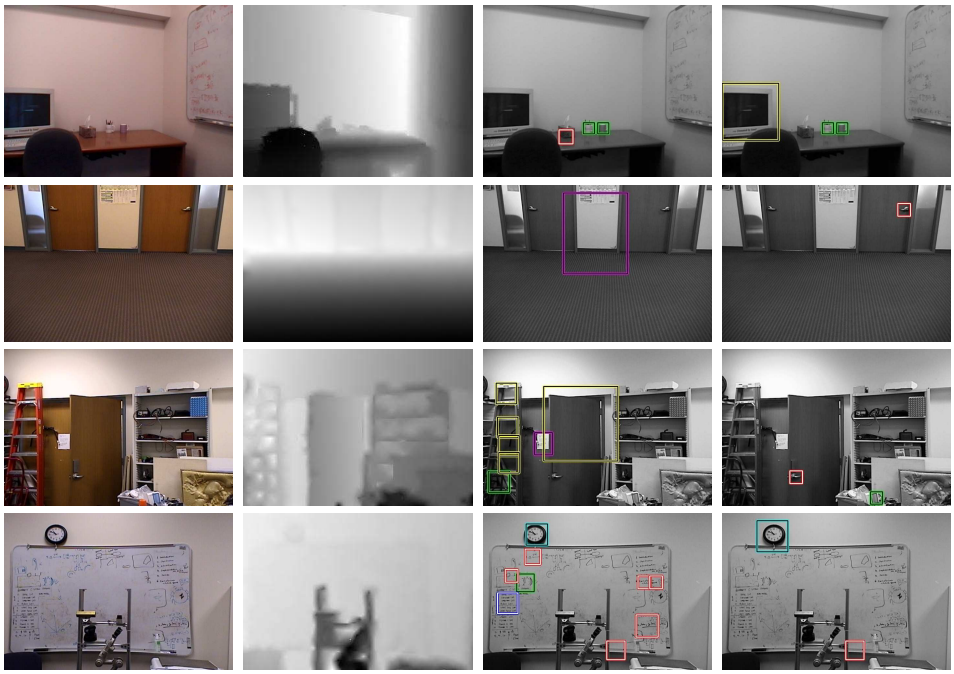
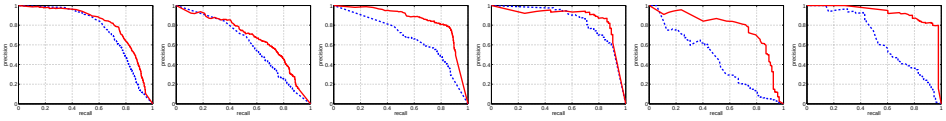


Fig. 7. Representative scenes showing our results from multi-modal object detectors compared against image-only detectors. (Best viewed in color.)

features at small scale. We found a large number of false-positive cups coming from other small objects such as coffee mugs. This suggests that a model which considers the location of other objects may improve accuracy for such classes.

In order to understand the contribution that each of our features makes to improving object detection, we evaluated the performance of the image-only detector augmented with each 3-d feature separately (see Table 9). Here we compare performance by measuring the maximum F_1-score. As expected, object dimensions and height above the ground are the strongest individual features.

Finally, to gain an intuition for where our method works, we provide some representative results in Figure 7. The first row shows that our multiple sensors help when objects are partially occluded. Here the monitor is detected even though the left edge is not visible. With some visual features missing, the image-only detector cannot conclude the existence of the monitor. The second row shows how knowing object size can be used to reject the false-positive disposable cup despite the strong visual resemblance. In this scene our method also successfully detects one of the two door handles. The last two rows show how image-only detectors are easily confused by significant clutter or textured backgrounds. Interestingly, the multi-modal detector incorrectly labels some trash as a coffee mug (bottom of the third scene). A patch on the whiteboard that resembles a door handle is also mistakenly labeled by our detector in the last scene.



(a) Mug (b) Cup (c) Monitor (d) Clock (e) Handle (f) Ski boot

Fig. 8. Precision-recall curves for commonly found household/office objects. Results from 2-d object detectors shown in dashed blue; results from 3-d augmented detectors shown in solid red. Scores are computed by first applying non-maximal neighborhood suppression to remove overlapping detections. A true-positive is counted if any detection overlaps with our hand-labeled groundtruth by more than 50%. Any detection that does not overlap with a groundtruth object (of the correct class) is considered a false-positive.

Detectors	Mug	Cup	Monitor	Clock	Handle	Ski boot
image-only	**0.707**	**0.594**	**0.645**	**0.776**	**0.506**	**0.594**
w/ height	0.737 (0.03)	0.619 (0.03)	0.755 (0.11)	0.760 (-0.02)	0.708 (0.20)	0.676 (0.08)
w/ distance	0.715 (0.01)	0.600 (0.01)	0.673 (0.03)	0.847 (0.07)	0.477 (-0.03)	0.616 (0.02)
w/ surface var.	0.753 (0.05)	0.608 (0.01)	0.695 (0.05)	0.796 (0.02)	0.563 (0.06)	0.537 (-0.06)
w/ obj. dim.	0.735 (0.03)	0.609 (0.02)	0.714 (0.07)	**0.874** (0.10)	0.605 (0.10)	0.763 (0.17)
laser (3d) only†	0.206 (-0.50)	0.241 (-0.35)	0.582 (-0.06)	0.835 (0.06)	0.229 (-0.28)	0.294 (-0.30)
multi-modal (all)	**0.768** (0.06)	**0.650** (0.06)	**0.821** (0.176)	0.865 (0.09)	**0.760** (0.25)	**0.879** (0.29)

Fig. 9. Comparison of maximum F_1-score for image-only detectors augmented with individual 3-d features. The delta over image-only detectors is given in parentheses. The 3-d only results (†) are provided for comparison and are calculated for the same set of candidate rectangles returned by the image-only detector (but the log-odds ratio feature is not used).

6 Discussion

In this paper we proposed a multi-modal object detector for robots situated in their environments. We showed how robots can exploit 3-d data from a low resolution depth sensor—the most common mode of robotic perception after optical sensing—by combining it with 2-d image data for robotic object detection in real-world environments. Our main contributions were two-fold. First, we showed how super-resolution techniques allow us to obtain a high resolution scene representation consisting of pixel intensities, 3-d locations, and surface normals. Second, we showed how 3-d features and contextual cues derived from this scene representation can be combined with a state-of-the-art 2-d object detector to significantly improve detection accuracy. Importantly, our method works with *any* 2-d object detector.

Our experimental results showed that the multi-modal detector improves over a baseline 2-d detector for common household/office objects. The improvement is most striking for objects which lack distinguishing features or those with high intra-class variance. Object size and location (height above the ground) are the strongest 3-d features, both of which are easily derived from our multi-sensor scene representation. An exciting avenue for further work is the exploration of more sophisticated 3-d features while still restricting ourselves to real-time depth sensors which are inherently low resolution.

Our architecture also fits well with the needs of real-time robotics. The data processing pipeline allows for multi-threading and can be scaled up to an arbitrary number of object detectors. The slowest operation is the extraction of the patch-response features used by our 2-d detectors (taking 10–20s per frame per object

class). Performance can be improved by pruning parts of the scene based on 3-d cues (such as size or surface variation) which are quick to compute.

Vision-only object detection systems are plagued by a number of difficulties in real-world scenes, e.g., lighting, texture, occlusion, etc. Many of these difficulties can be overcome by augmenting visual perception with complementary sensor modalities (e.g., depth and infrared) using the methods described in this paper, providing a step towards robust object detection for autonomous robots.

Acknowledgements

Support from the Office of Naval Research under MURI N000140710747 is gratefully acknowledged.

References

1. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results (2007)
2. Fink, M., Perona, P.: Mutual boosting for contextual inference. In: NIPS. (2003)
3. Murphy, K., Torralba, A., Freeman, W.: Using the forest to see the tree: a graphical model relating features, objects and the scenes. In: NIPS. (2003)
4. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. In: CVPR. (2006)
5. Viola, P., Jones, M.: Robust real-time face detection. IJCV (2004)
6. Brubaker, S., Wu, J., Sun, J., Mullin, M., Rehg, J.: On the design of cascades of boosted ensembles for face detection. In: Tech report GIT-GVU-05-28. (2005)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)
8. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: CVPR. (2005)
9. Torralba, A., Murphy, K., Freeman, W.: Contextual models for object detection using boosted random fields. In: NIPS. (2004)
10. Torralba, A., Murphy, K., Freeman, W.: Sharing visual features for multiclass and multiview object detection. PAMI (2007)
11. Torralba, A., Oliva, A., Castelhano, M.S., Henderson, J.M.: Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. Psychol Rev (2006)
12. Leibe, B., Cornelis, N., Cornelis, K., Gool, L.V.: Dynamic 3d scene analysis from a moving vehicle. In: CVPR. (2007)
13. Nuchter, A., Lingemann, K., Hertzberg, J., Surmann, H.: Accurate object localization in 3d laser range scans. In: IEEE. (2005)
14. Rabbani, T., van den Heuvel, F.: Efficient hough transform for automatic detection of cylinders in point clouds. In: ISPRS. (2005)
15. Chen, H., Wulf, O., Wagner, B.: Object detection for a mobile robot using mixed reality. In: VSMM. (2006)
16. Diebel, J., Thrun, S.: An application of markov random fields to range sensing. In: NIPS. (2006)
17. Torres-Mendez, L.A., Dudek, G.: Reconstruction of 3d models from intensity images and partial depth. In: AAAI. (2004)
18. Yang, Q., Yang, R., Davis, J., Nister, D.: Spatial-depth super resolution for range images. In: CVPR. (2007)
19. Liu, D., Nocedal, J.: On the limited memory method for large scale optimization. In: Mathematical Programming B. Volume 45. (1989) 503–528
20. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Technical report, Dept. of Statistics, Stanford University (1998)