Learning texton models for real–time scene context

Alex Flint, Ian Reid and David Murray
Active Vision Laboratory
Oxford University, UK
{alexf,ian,dwm}@robots.ox.ac.uk

Abstract

We present a new model for scene context based on the distribution of textons within images. Our approach provides continuous, consistent scene gist throughout a video sequence and is suitable for applications in which the camera regularly views uninformative parts of the scene. We show that our model outperforms the state–of–the–art for place recognition. We further show how to deduce the camera orientation from our scene gist and finally show how our system can be applied to active object search.

1. Introduction

Traditional computer vision systems have taken a local approach to image understanding in which visual elements are sought by searching exhaustively over an image. For example, a typical approach to object detection is to invoke a classifier on many windows within an image, inputting for each region some local image evidence and outputting the presence or otherwise of the relevant objects ([16] for example).

Recently, several researchers have taken a different approach in which features generated from the whole image are provided to the inference process alongside the traditional local image evidence [14, 3]. This gives context to the local inference process since the global image information can be used to infer how the local evidence fits into the "bigger picture". For example, the accuracy of object detection improves when using global image features since detections in unlikely places (pedestrians in the sky, cars in trees) can be removed [14, 1].

Many of these approaches are designed to operate on well–posed photographs—the type that humans generate when a camera is oriented and directed at the scene with some care [2, 8]. In contrast this paper is concerned with so–called "egocentric" applications in which the camera is attached to some agent (often a human) that is acting in the world without concern for how the camera's view is affected. Such applications are characterized by frequent frames that contain uninformative views of the scene. For example, consider Figure 1: the top row shows typical images from the dataset of Fei–Fei and Perona [7] while the bottom row shows random frames from an ego–centric sequence. Such sequences also tend to contain more frequent motion blur and poorer illumination conditions as a result of the agent's lack of concern for the camera.

Figure 1. Well–posed photographs (top row) versus frames captured in an egocentric application (bottom row). The latter contain less informative frames, often poorly composed and illuminated.

Applications in the egocentric domain often require the system to process incoming frames at video rate in order to provide timely feedback to the wearer or in order that the system can update its knowledge about the world. Even when the system is not required immediately to report its knowledge, video frames will be arriving continuously and, given an unknown sequence length and limited storage capacity, the system will have to process the video at frame rate or else eventually start dropping frames.

We are particularly interested in indoor environments since these tend to produce particularly challenging images (often containing just a patch of carpet or section of a wall) but are the context in which vision for assistive applications and household robots will be required to operate.

In this paper we present progress towards a new model for scene context that is suitable for egocentric applications. Our approach transforms input frames into a set of textons and then deduces scene context from the spatial layout of the textons. We can examine the spatial layout of textons

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

in an image in order to recognize places or deduce facts about the camera orientation. Alternatively we can model the spatial relationship between textons and some object of interest in order to perform "active search" by using the observed textons to assess the probability of finding the object of interest at locations inside and outside the image.

The remainder of this paper is organized as follows. In section 2 we describe previous work in this area, then in section 3 we present our texton model for scene context. In section 4 and 5 we describe two classification problems to which we have applied our system, place recognition and camera orientation classification. In section 6 we show how our system can be used for active object search. Finally, section 7 contains closing remarks.

2. Previous work

The work of Torralba *et al.* has been very influential in underscoring the importance of context in vision [13]. In [11, 14] Oliva and Torralba proposed the "gist" descriptor to capture holistic image context. Under their approach an image is divided into a 4×4 grid and then passed through a bank of Gabor filters. The average response for each filter in each cell is inserted into a feature vector and then PCA is used to reduce dimensionality. Torralba demonstrates scene category recognition by estimating Gaussian mixtures for each class in gist feature space. The goal of our work is similar to theirs but we model the texture structure of scenes in a more explicit way that allows both improved recognition performance and reduced training data requirements.

Hoiem, Efros, and Hebert (HEH) [2] have shown how to recover basic geometric structure from a single image. Their system is able to segment an image into "ground", "upright", and "sky" components. They learn an affinity metric between superpixel pairs using a range of appearance statistics including colour, texture, location, and vanishing points. They further show how this model can be combined with an object detector to improve both geometry estimates and object detection [3]. Our system is also intended to capture geometric context although it is not as explicitly 3–dimensional as their method. However, our system is intended for video sequences that would be unsuitable for the HEH system for a number of reasons: apart from containing indoor environments (they concentrate on outdoor scenes) our dataset contains frequent uninformative images (see Figure 1) as well as images captured at odd camera orientations. Unlike the HEH approach, our system runs in real–time and does not require ground–truth segmentations for training.

Heitz and Koller have demonstrated an object detection system in which spatial context is introduced by learning relationships between objects and the appearance of image patches that are likely to appear nearby [1]. They cluster superpixels based on colour and texture features and then apply structural EM to learn the object/region relationships. Our work is distinct to theirs because (1) our approach is not inherently tied to object detection, (2) we use textons instead of superpixels, (3) we model continuous relationships between scene parts, whereas they select from a pool of discrete relationships ("left of", "below", etc), and (4) our system operates in real–time.

Within the robotics domain, context has been used by Mozos *et al.* [10] to deduce place categories from laser range data. They compute a number of geometric features from the range scans and use AdaBoost to recognise rooms, doorways, and corridors.

The first account of textons was given by Julesz [5], who introduced the idea to the neuroscience community over 25 years ago. The texton notion that we adopt in the present work was first formulated in the computer vision domain by Malik *et al.* [9]. They run a bank of linear filters over the image and then vector quantize using K–means clustering, with the resultant cluster centres becoming texton exemplars.

The use of textons for material recognition was also explored by Varma and Zisserman [15], who model material appearances in terms of a histogram over texton frequencies. Our notion of textons follows this work closely, although we apply it to a completely different problem. Two alternative definitions of textons were discussed in [17], but neither of these is appropriate for our application due to training data requirements and efficiency.

3. Textons for Scene Context

In this section we describe our model for scene context. For each image we run a bank of Gabor filters containing n_o orientations and n_s scales. Each pixel is then assigned a $(n_o \times n_s + 3)$–dimensional feature vector consisting of the the filter responses at that point as well as the pixel's coordinates in HSV colour space. In our experiments we set $n_o = 4$ and $n_s = 3$ but in practice we found that varying these parameters had little effect on the performance of our algorithm. This agrees with the finding in [14] that even using entirely different filters does not significantly affect overall performance (their comment was with regard to Gabor filters versus steerable pyramids).

Using K–means we cluster the feature vectors for all pixels in all images. The output cluster centres become the texton codebook. We then return to the original images and label each pixel by the index of the texton that best matches (in the Euclidean sense) its feature vector. This reduces the dimensionality of the pixel data substantially, and makes our model tractable in terms of both time and training data requirements. In our experiments we set $K = 25$ but similar results were obtained with values between 15 and 50. For $K < 15$ there are too few cluster centres to capture important scene structure and for $K > 50$ the system selects

outliers as many of the cluster centres, which has negligible effect on classification performance but increases the computational complexity of the model.

Our system is predicated on the hypothesis that the layout of textons in images is correlated with scene structure and can be used for inference about visual context. This correlation can be seen explicitly in Figure 7: the bottom two rows depict textons that the system has selected to exemplify edges that are roughly vertical (3rd row) and roughly horizontal (4th row). The three rightmost columns show an average over the occupancy maps for all images in our "corridor" dataset. These are instructive because they show that the texton locations encode some of the geometric structure of the environment. For example, the "vertical edge" texton (3rd row) tends to be stratified such that vanishing points are below the image centre when the camera is facing downwards and above it when the camera is facing upwards, as expected. Hence the system has identified without supervision some of the image elements and geometric constraints that are sought explicitly in other vision systems (e.g. [6]).

The textons generated for a typical indoors training set are as follows. The first 20% or so of the textons represent image regions that are essentially textureless (i.e. the Gabor responses are all close to zero). The next 60% or so represent edges at different orientations and scales. The remaining textons typically represent more exotic image elements such as blobs and ridges.

We wish to model image categories according to the locations in which textons appear. One popular approach is the bag–of–features model [4] but this would tie us to the image frame (and hence to a particular camera orientation) since pixel locations are represented relative to the image origin. Instead we propose a new bag–of–texton–pairs model in which an image is represented as a collection of observed texton pairs $\{(t_i, t_j, s_{i,j})\}$ where t_i and t_j are the texton labels and $s_{i,j}$ is the displacement between the image locations at which they were observed. By considering only displacements and not absolute pixel locations in our model we gain some robustness to camera orientation.

For an image I containing N pixels there are N^2 such pairwise observations. We model the likelihood given class c as

$$p(I \mid c) = \prod_{i=0}^{N} \prod_{j=0}^{N} p(t_i\ t_j\ s_{i,j} \mid c) \tag{1}$$

where we have assumed independence between observations for tractability. We compute the likelihood (1) by estimating the full joint $p(t_i, t_j, s_{i,j}, c)$ using a histogram. We could have used Parzen windowing [12] for the continuous variable $s_{i,j}$ but due to the very large number of samples we obtain (even from a modest number of training images) we found this to be unnecessary.

For images of reasonable size the cost of enumerating all N^2 texton pairs is prohibitively expensive. We overcome this by overlaying a $M \times M$ grid on the image and counting the occurrences of each texton within each grid cell. We then enumerate all pairs of grid cells and evaluate the texton pairs in aggregate. Hence for grid cells C_a and C_b containing n_i^a and n_j^b instances of texton t_i and t_j respectively, we evaluate $n_i^a n_j^b$ instances of the observation $(t_i, t_j, s_{a,b})$ where $s_{a,b}$ is the distance between the centres of the grid cells. We have lost some precision in the texton locations since each texton is effectively moved to the centre of the grid cell containing it, but our experiments show that we are still able to capture sufficient salient information.

During training we evaluate these aggregated observations by multiplying the entry we make in the histogram by $n_i^a n_j^b$, and when classifying some input image I the aggregate observations correspond to multiplications in the class–conditional log likelihood:

$$\log p(I \mid c) \tag{2}$$

$$= \sum_{a=0}^{M^2} \sum_{b=0}^{M^2} \sum_{i=0}^{K} \sum_{j=0}^{K} n_i^a n_j^b \log p(t_i\ t_j\ s_{a,b} \mid c) \tag{3}$$

In both cases the aggregated observations can be evaluated in a single step so the complexity is reduced from $O(N^2)$ to $O(M^4 K^2)$. In practice we found that setting M=8, K=25 was sufficient to capture much of the salient image information while allowing our system to run at video frame rate.

4. Place Recognition

We applied our system to the problem of place recognition. Our data set consisted of several video sequences captured in a hostel using a low–quality camera with a resolution of 320×240, which moved rapidly with the user's upper body. The sequences involved frequent motion blur and rapid variations in camera orientation.

We labelled each frame with the place that it was captured in. There were five labels: bedroom, kitchen, common room, garden, and corridor. As an added challenge we gave all frames captured in corridors the same label (there were four different corridors in the sequence with considerable variations in appearance).

This experiment does not correspond to place *category* recognition since most of the labels included frames from only one place instance. However, it is harder than strict landmark–style localization because, as shown in Figure 2, many images with the same label contain non–overlapping views of the room they were captured in, yet the system is expected to recognize all of them as belonging to the same place.

We compared our system with the gist descriptor of Torralba *et al.* and a K–nearest–neighbours baseline. For the gist descriptor we used the same Gabor filter bank that

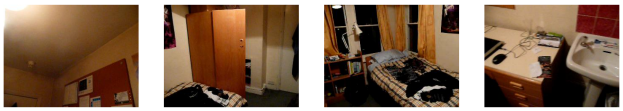

Figure 2. Four frames with the "bedroom" label. There are almost no overlapping scene parts but the system is required to (and did successfully) recognize each of them as part of the same place.

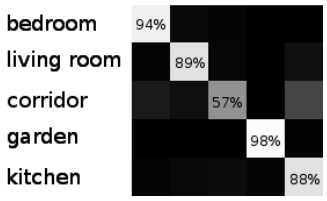

Figure 3. Confusion matrix for place recognition.

we used in our own system and we estimated the class–conditional likelihood in feature space by building Gaussian mixture models with the Gaussians constrained to be spherical, exactly as described in [14].

Initially we used 230 frames for training and 490 frames for evaluation (our training and evaluation sets were taken from separate sequences). The results from this experiment are shown in the middle row of Table 1 and in Figure 3. Our system outperformed Torralba's by a large margin. We suspected that the poor performance of Torralba's system was due to the training data not sufficiently populating the 32–dimensional feature space. This exemplifies one of the important advantages of our system — namely the ability to learn from limited training data. However, to show that this is not the *only* advantage of our system we ran auxiliary experiments with larger and smaller training sets. When the training set was enlarged our system outperformed Torralba's by a significant but smaller margin, and when the training set was decreased our system's performance diminished only slightly, whereas we were unable to estimate the Gaussian mixture for Torralba's approach due to the sparsity of training samples.

Figures 4 and 5 show positive and negative results from our system respectively. Note how our system recognises images containing disjoint views of a room as belonging to the same place.

# train frames	Our system	Torralba *et al.*	KNN
103	**81%**	—	45%
230	**83%**	62%	52%
565	**85%**	70%	55%

Table 1. Place recognition results with varying numbers of training frames (total for all labels). For the experiment with 103 training frames (top row) we were unable to estimate the Gaussian mixtures required for Torralba's system due to the sparsity of the training examples in feature space.

5. Camera Orientation Classification

In this section we show how our system can deduce a coarse camera orientation from a single image. We are interested only in the tilt of the camera with respect to the ground plane. Our intention is to rapidly make a coarse estimate of camera orientation such as might be provided as a prior to a SLAM system or similar. We pose the problem as one of classification with three possible labels: "up", "straight", and "down" (see Figure 6). The "straight" label represents images taken with the camera axis parallel to the ground plane, plus or minus $22.5°$, and the "up" and "down" labels represent all orientations facing further upwards or downwards respectively.

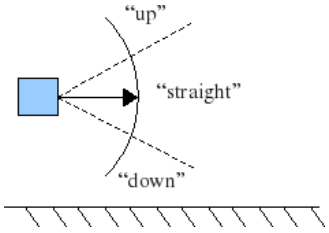

Figure 6. Labels for camera orientation classification.

We captured three sequences in which the camera orientation was fixed within the one of the above orientation ranges. We included footage from five different places (the same ones used in the previous section) but we labelled the frames according to orientation only. We then trained our system to distinguish between the three orientation categories as in the previous section. This represents a difficult classification task because the system must learn properties that correlate with camera orientation but are not tied to the appearance of a particular place.

We again compared with the "gist" of Torralba *et al.* and a KNN baseline. We ran auxiliary experiments with an enlarged training set as in the previous section. The results of these experiments are shown in Table 2 and Figure 8. Our system again outperformed both other classifiers by a significant margin. Some example frames for which our system correctly identified the camera orientation are shown in Figure 9. Of particular interest is our system's ability to generalize across images taken with the same camera orientation at several different locations.

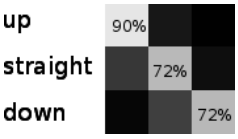

Figure 8. Confusion matrix for camera orientation classification.

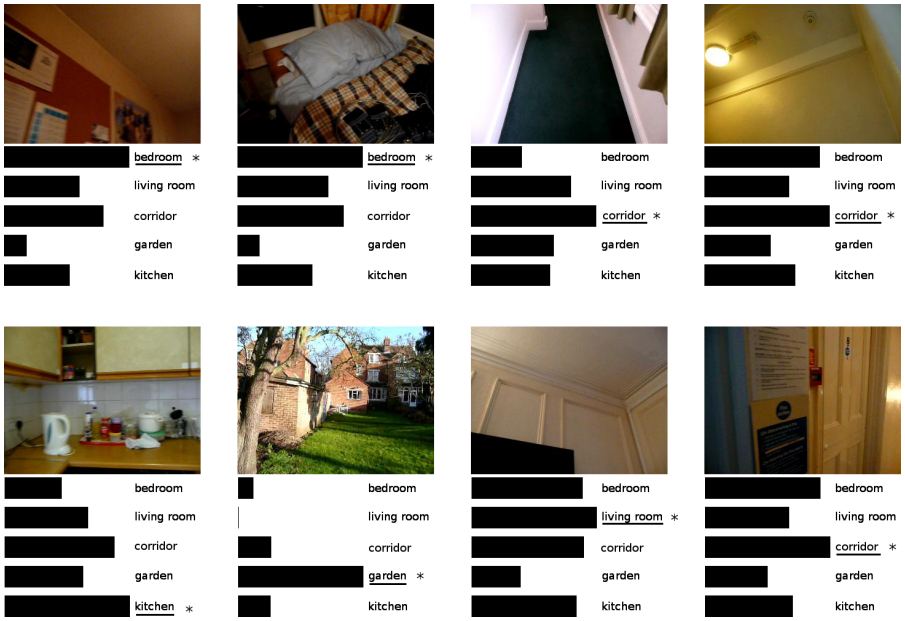

Figure 4. Example frames for which our classifier succeeded. The ground truth label is underlined and the output from our system is starred. We show the log likelihoods not the actual posterior because the large number of terms in (1) causes the posterior to always be sharply peaked and hence the log likelihood is more informative for visualisation. Note the variation between frames with the same label, and the poverty of the information contained in many frames.

Figure 5. Example frames for which our classifier failed. See caption of Figure 4.

# train frames	Our system	Torralba *et al.*	KNN
88	**70%**	61%	59%
728	**79%**	63%	59%

Table 2. Camera orientation classification results using small and large training sets. We were able to estimate the Gaussians for Torralba's system using only 88 training examples because there were fewer labels than in the place recognition problem.

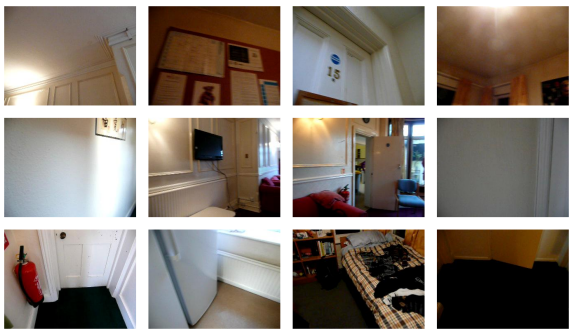

Figure 9. Twelve frames for which our system correctly identified the camera orientation. From the top to bottom the rows contain images from the "up", "straight", and "down" classes.

6. Active Search

To demonstrate the powerful contextual information provided by our system we applied it to the problem of active search. The scope of this problem varies considerably across the literature: in our case we are interested in determining where a particular object is likely to appear relative to the current camera view, and importantly we consider locations both inside and outside the image. That is, given the

Figure 7. Four example textons generated unsupervised for the camera orientation classification problem. From top to bottom the textons represent roughly "wall or ceiling", "floor", "vertical edge", and "horizontal edge". The six columns on the left show examples of where the texton was found. The three columns on the right show the average occupancy map over our dataset for images taken from an upwards–facing, horizontal, and downwards–facing camera (from left to right in the figure). The layout of the textons correlate strongly with camera orientation, which illustrates how our system is able to distinguish between camera orientations based on texton layout.

current image evidence, our system can answer questions about how likely the object is to appear at locations within the image *and* how likely we are to find the object if we move the camera in various directions. This differs from the standard formulation in which only locations within the image are considered [14, 1, 3]. This crucial difference stems from our ultimate goal of "steering" an active camera in order to find an object that may not currently be within the camera's field of view (although we do not use an active camera in the present work), whereas the traditional formulation is motivated by the desire to improve the accuracy and/or efficiency of an object detector [14, 1, 3].

We wish to evaluate the posterior

$$p(\boldsymbol{x} \mid I) \tag{4}$$

that the object is present at location $\boldsymbol{x}$ in image I. The location $\boldsymbol{x}$ is measured in pixel coordinates but may be outside the image bounds, in which case (4) is to be interpreted as the probability of observing the object at $\boldsymbol{x}$ were the camera centre centre to be translated orthogonal to the optical axis such that the location $\boldsymbol{x}$ became visible in the camera frame.

Following our approach to classification we transform an input image I into a collection of textons $D = \{(t_i, \boldsymbol{y_i})\}$ where t_i is the i^{th}texton and $\boldsymbol{y_i}$ is its location in the image. We model the posterior (4) in terms of the observed textons. For tractability we assume that the textons are conditionally independent of each other given the object location $\boldsymbol{x}$. We have

$$p(\boldsymbol{x} \mid D) = \frac{p(\boldsymbol{x}\, t_1 \cdots t_N\, \boldsymbol{y_1} \cdots \boldsymbol{y_N})}{p(t_1 \cdots t_N\, \boldsymbol{y_1} \cdots \boldsymbol{y_N})} \tag{5}$$

$$= \frac{\prod p(\boldsymbol{x}\, t_i\, y_i)}{\prod p(t_i\, y_i)} \tag{6}$$

$$\log p(\boldsymbol{x} \mid D) = \sum \log p(\boldsymbol{x}\, t_i\, \boldsymbol{y_i}) - \sum \log p(t_i\, \boldsymbol{y_i}) \tag{7}$$

The first term in (7) describes the relationship between texton positions and object positions. We model this distribution in terms of the the object–texton displacement:

$$p(\boldsymbol{x}, t_i, \boldsymbol{y_i}) = f(\boldsymbol{x} - \boldsymbol{y_i}, t_i) \tag{8}$$

By assuming that only the displacement between objects and textons is important we allow our system to reason about locations outside the image boundary, and simultaneously gain some independence from camera orientation.

We learn the function f in (8) and the second term in (7) from a training set by building histograms. As in the previous section, we could have used Parzen windowing but found it to be unnecessary due to the large number of textons present in each image.

6.1. Experiments

We trained our system on several common objects appearing in our indoor sequences. In each case the training set consisted of a set of frames with the object location

$\boldsymbol{y}$ marked in each. As described above, the textons were learned unsupervised.

In one experiment we trained our system on 86 frames of 3 different fire extinguishers. The intention was that our system would learn that fire extinguishers are found near the intersection of the wall and floor. Figure 11 shows the marginal $p(\boldsymbol{y} \mid D)$ representing the probability of finding a fire extinguisher at various heights relative to the camera centre. Figure 12 shows the probability $p(\boldsymbol{x} \mid D)$ of finding a fire extinguisher at locations outside the image. Our system is able to generalize well from the training data and produces sensible results even in the presence of uninformative camera views.

In separate experiments we trained our system to find doorknobs and kettles. In the former case we obtained results that were equally as encouraging as the fire extinguisher sequence, but in the latter case our system was not able to generalize well from a single training sequence containing only one kettle instance. This suggests that our system is best suited for objects that occur frequently in the environment rather than objects for which there is only one instance present.

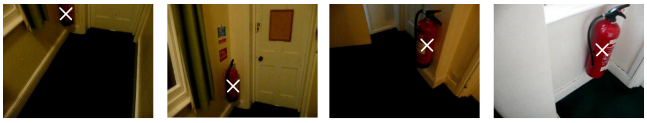

Figure 10. Selected training images for the fire extinguisher active search experiment.

7. Conclusion

We have shown how scene context can be deduced from the spatial layout of textons in an image. We have reported the performance of our system when applied to place recognition, camera orientation classification, and active search in indoor environments. Our system outperforms the state–of–the–art for the two classification problems and obtains encouraging results when applied to active search. Results are particularly impressive in situations where only a limited section of the environment is visible, such as frames containing only a section of the floor or walls.

Our system's most frequent failure case occurs for images containing scene parts that are common to several places. This suggests introducing temporal integration to represent the dependency between camera locations in consecutive frames. In order to focus exclusively on recognition performance we did not include such experiments in the present work.

This paper represents preliminary results and we intend to test our system within a wider class of environments and with more object types. In particular we intend to compare our findings with those obtained by others on standard datasets.

A major advantage of our system is its video-rate performance. Most contextual vision systems proposed in the past have been targeted at object recognition, and speed of computation has not been a priority. Our system has been designed expressly for live use, and represents one of the first applications of contextual vision techniques to this domain.

References

[1] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *Proc 10th European Conf on Computer Vision*, pages 30–43, 2008.

[2] D. Hoiem, A. A. Efros, and M. Hébert. Geometric context from a single image. In *Proc 10th IEEE Int Conf on Computer Vision*, pages 654–661, 2005.

[3] D. Hoiem, A. A. Efros, and M. Hébert. Putting objects in perspective. In *Proc 24th IEEE Conf on Computer Vision and Pattern Recognition*, pages 2137–2144, 2006.

[4] T. Jebara. Images as bags of pixels. In *Proc 9th IEEE Int Conf on Computer Vision*, pages 265–272, 2003.

[5] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–97, Mar. 1981.

[6] J. Koseckà and W. Zhang. Video compass. In *Proc 7th European Conf on Computer Vision*, volume 2353 of *Lecture Notes in Computer Science*, pages 4: 476–490. Springer, 2002.

[7] R. F. L. Fei-Fei and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *Proc CVPR Workshop on Generative-Model Based Vision*, pages 2169–2178, 2004.

[8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc 24th IEEE Conf on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[9] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: cue integration in image segmentation. In *Proc 7th IEEE Int Conf on Computer Vision*, volume 2, pages 918–925, 1999.

[10] O. Mozos, C. Stachniss, and W. Burgard. Supervised learning of places from range data using adaboost. In *Proc 2005 IEEE Int Conf on Robotics and Automation*, pages 1730–1735, 2005.

[11] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[12] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[13] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proc 10th IEEE Int Conf on Computer Vision*, pages 1331–1338, 2005.

[14] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.

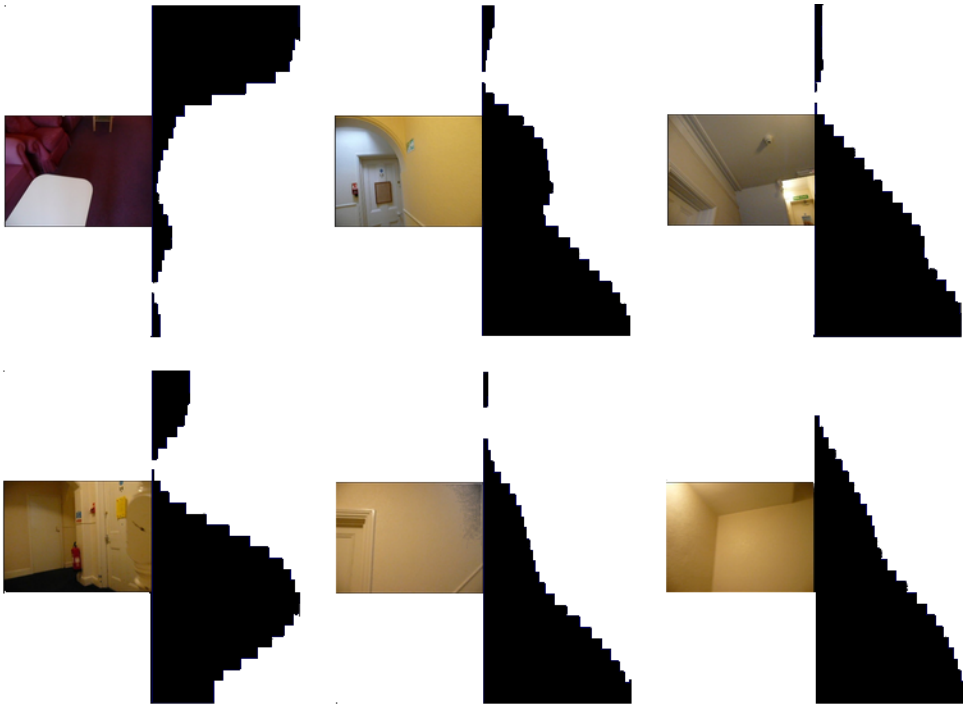

Figure 11. Marginal for presence of fire extinguishers at y positions relative to image centre.

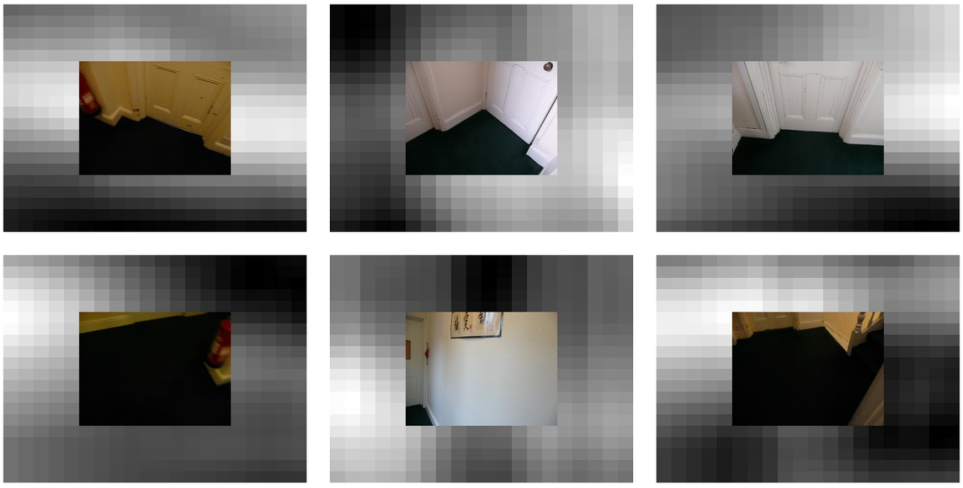

Figure 12. Marginal for presence of fire extinguishers at locations $\boldsymbol{x}$ outside the image. White pixels represent regions with greatest probability.

[15] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81, 2005.

[16] P. A. Viola and M. J. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[17] S. Zhu, C. Guo, Y. Wang, and Z. Xu. What are textons? *IJCV*, pages 121–143, 2005.