Detecting planes and estimating their orientation from a single image

Osian Haines
www.cs.bris.ac.uk/~haines
Andrew Calway
www.cs.bris.ac.uk/~andrew
Visual Information Laboratory
University of Bristol

Abstract

We propose an algorithm to detect planes in a single image of an outdoor urban scene, capable of identifying multiple distinct planes, and estimating their orientation. Using machine learning techniques, we learn the relationship between appearance and structure from a large set of labelled examples. Plane detection is achieved by classifying multiple overlapping image regions, in order to obtain an initial estimate of planarity for a set of points, which are segmented into planar and non-planar regions using a sequence of Markov random fields. This differs from previous methods in that it does not rely on line detection, and is able to predict an actual orientation for planes. We show that the method is able to reliably extract planes in a variety of scenes, and compares favourably with existing methods.

1 Introduction

Planar surfaces are ubiquitous in man-made environments, and are important in automated analysis since they constrain the scene structure and allow a simplified geometric representation [2]. Planes are especially useful when only a single image is available, for estimating camera placement [13], wide-baseline matching [16] and creating simple 3D reconstructions [11], for example. Single image plane detection has the advantage that there is no need to accumulate parallax over many frames, so it can potentially be faster and more efficient, and it can be used when further information cannot be obtained – such as with photographs.

We present an algorithm to detect planar regions in a single image. This divides an image into planar and non-planar regions, and furnishes each plane with an orientation with respect to the camera – see figure 1. The objective is to group a set of salient points into distinct planar regions (i.e. we are not proposing a pixel-level segmentation), using texture and colour information. Our algorithm does not rely on geometric reasoning or assumptions of orthogonality, nor does it require the potentially difficult extraction of intermediate features, such as vanishing lines or rectilinear structures.

We are motivated by humans' apparent ability to interpret scenes easily, without explicit reference to geometric information, based on our prior visual experience. To make use of prior knowledge we use a machine learning approach, trained on data gathered in an urban environment. In this respect, we are inspired by recent methods which are able to solve similar single-image problems – for example surface layout detection by Hoiem *et al*. [11], and Saxena *et al*.'s depth map estimation [18].

http://dx.doi.org/10.5244/C.26.31

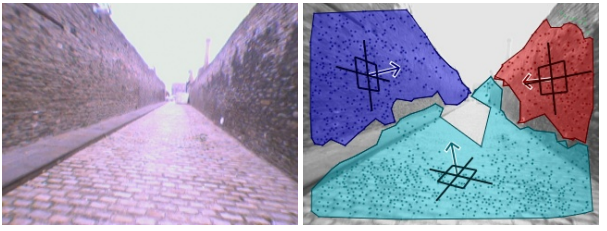


Figure 1: An example result of our algorithm. From such an image as this, we can detect planes and predict their 3D orientation, as shown on the right.

We make use of an algorithm which can classify planar image regions, and estimate their 3D orientation [8]. This is applied at multiple overlapping image locations, allowing estimates of local planarity to be calculated at each of a set of salient points in the image. These local plane estimates are used to segment the image into planar and non-planar segments, using a Markov random field (MRF) framework, resulting in a set of plane regions, each with an orientation estimate. To our knowledge, such detection of planar regions with associated orientation estimates, from a single image without explicit use of geometric information, has not been demonstrated before.

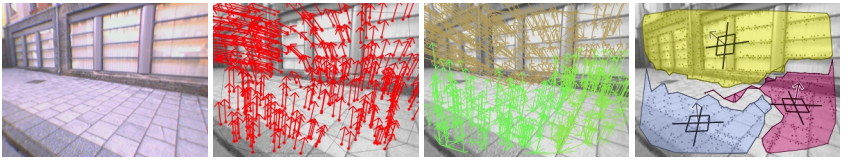
The paper is structured as follows. In the next section, we describe related work; section 3 gives a brief overview of our whole method, followed by a detailed description of the plane estimation method for individual regions in section 4, and the plane detection method in section 5. The results in section 6 show that the method performs well in a variety of environments, achieving a classification accuracy of 88% and a mean plane orientation error of 18.3°, and compares favourably with existing methods. Section 7 concludes the paper.

2 Background

One way to obtain planes from single images is to make use of geometric structures, such as vanishing points and images of rectangles [9]. Košecká and Zhang [13], for example, use line detection to find vanishing points, which are used to identify orthogonal planes. Similarly, Micušık et al. [16] show how grouping lines into quadrilaterals can be used to find planes in both indoor and outdoor scenes. These methods illustrate how such plane detection can be useful, for estimating camera pose and performing wide baseline stereo; but while they are ideal when strong lines are visible and the Manhattan world assumption is valid, they are not generally applicable.

An alternative class of approach is to make use of shape from texture methods, in which the statistical properties of surfaces are used to infer orientation [1]. Gårding [7], for example, estimates the tilt and slant of planar surfaces using the distribution of line segments; however, such methods rely on specific assumptions about the properties of the texture, and are difficult to apply to more general scenes.

Other methods use machine learning, and are more similar to our approach. The most prominent example is the surface layout recovery method of Hoiem *et al.* [11], which is capable of recovering the basic geometric layout of a scene, and creating simple 'pop-up' reconstructions; this shows that it is possible to obtain good information about structure by learning from monocular cues. Orientation is represented by classification into geometric classes, which gives a coarse estimate of plane orientation (quantised to four directions,



(a) (b) (c) (d)

Figure 2: Our method: from the input image (a), we sweep a classifier over the image to obtain a point-wise local plane estimate (b). This is segmented into distinct regions (c), from which the final plane detections are derived (d).

namely left, right, forward and supporting surfaces) – so while this gives impressive performance in terms of geometric labelling, it is not explicitly intending to detect planes, and is unable to give precise orientation.

3 Overview

We first provide an outline of the whole method; further details can be found in subsequent sections. Our task is to group the salient points of an image into regions corresponding to planar surfaces, complete with an orientation estimate, or into non-planar regions. To achieve this, we employ the method described by Haines and Calway [8], which, given a region of an image, classifies it as a plane or not, and predicts its orientation – we refer to this as 'plane estimation'. However, this in itself is not sufficient to detect planes, since we do not know the boundaries between different regions. Our solution is to sweep a window over the image, running the plane estimation about each salient point, in order to estimate the local planarity (class and orientation) at each point, producing what we will refer to as the local plane estimate (figure 2(b)).

While this appears to reflect the underlying structure, it is still insufficient: first, it does not actually tell us about planar surfaces, since there is no notion of connectivity. Secondly, the estimates at each point are only an average of all planes on which it might lie, as opposed to a decision of the surface it belongs to. The next stage is therefore to separate the points, first into planar and non-planar areas, and then to segment the planar regions into distinct planes according to their orientation. This is done using the local plane estimate at each point plus smoothness constraints, implemented within a MRF framework, to produce spatially coherent regions, as illustrated in figure 2(c).

These segmented regions are then passed back to the plane estimation algorithm, to ensure they are truly planar, and to obtain an updated estimate of their orientation. These regions constitute the final output of the plane detection algorithm, as shown in figure 2(d).

4 Plane estimation

In this section we describe the plane estimation step, in which individual regions are classified and given an orientation estimate. A full description of the method can be found in [8] – here we give an overview and describe how it is adapted for use during plane detection. We emphasise that this method was developed to only estimate the planarity and orientation of

a given, pre-segmented region, and is not able to locate the planes in an image – that is the focus of the current work, which we describe in section 5.

To summarise: this method proceeds by creating descriptors for points in an image region, which are quantised using a pre-computed codebook to form a bag of words; the word vectors are projected into a latent topic space to reduce their dimensionality, and combined with spatial distribution information from the 2D points by making use of spatiograms [4]. Such descriptions of training regions are used to train a classifier and regressor, which are then used to predict the planarity and orientation of new test regions.

4.1 Training data

To acquire training data which is well suited to the plane detection task, we sweep a window over marked-up ground truth images (as we do when creating the local plane estimate – see section 5.1). We use images taken from a video sequence, collected in an urban environment, hand-segmented into planar and non-planar regions (so that every pixel is covered), and labelled according to their class (plane or non-plane). To specify the ground-truth orientation for a region we mark by hand four points corresponding to the corners of a rectangle lying on the plane in 3D; extending these mutually orthogonal worldlines to vanishing points allows calculation of the normal to the plane. Where an extracted region lies over the boundaries between ground-truth regions, we set its class and orientation to be the modal class and mean orientation, respectively, of the underlying ground-truth regions.

4.2 Features

To describe the image regions, we first detect salient points – to reduce the amount of data required, and focus only on regions with interesting appearance. For this we use the Difference of Gaussians detector [15], which gives the location and scale of the detected feature. Descriptors are created for each point, using the detection scale to set the patch size.

Two complementary feature descriptors are used: we represent texture with histograms of gradient orientation (following [8]), and colour with RGB histograms. Gradient orientation histograms are created by building histograms of estimated edge orientation, each with 12 angular bins, from the four quadrants surrounding the patch centre – these are concatenated to form a 48 dimensional gradient descriptor.

Colour descriptors are created by concatenating intensity histograms of the red, green and blue channels of the image patch, each comprising 20 bins, forming a 60-D descriptor. The importance of colour was demonstrated by Hoiem et al. [11], and we found it gave better classification accuracy than the gradient histograms. As we hoped, combining both gives performance superior to either in isolation.

However, colour is not likely to be beneficial for estimating orientation – indeed, we found that using colour histograms alone gave poor orientation accuracy, and combining both descriptors conferred no improvement. For this reason, we maintain separate topic spaces for classification and regression, the former using gradient and colour descriptors and the latter with gradient only, meaning that each region will have two spatiograms.

4.3 Region description

We use a visual bag of words [19] to combine the information from features in a region, built by clustering representative feature vectors to form two codebooks (one for the gradient

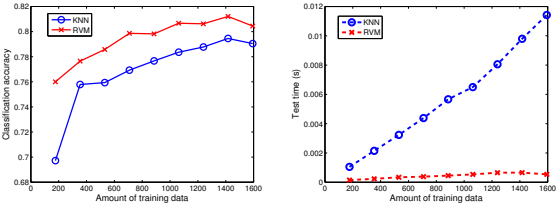
histograms and one for the and colour histograms); new descriptors are quantised with these, to form visual words. To combine the two vocabularies, and reduce their dimensionality, we derive a latent topic space [10] from the concatenation of the two term-document matrices, using Orthogonal Non-negative Matrix Factorisation (ONMF) [6] (chosen because it allows simple projection into the topic space, and since non-negative coefficients are essential for creating the spatiograms described next).

From the topics in each image region, we create a spatiogram [4] – this is a histogram augmented with the mean and covariance of point locations contributing to each bin – where each bin corresponds to a topic. This encodes the spatial distribution of the topics, which as demonstrated in [8] provides improved performance over a standard bag of words model, which discards all location information. Given two regions, we assess their similarity by comparing their spatiograms, $\mathbf{S}^A$ and $\mathbf{S}^B$, using the method proposed by Ó Conaire [17], denoted by $\rho(\mathbf{S}^A, \mathbf{S}^B)$. This forms the basis of classifying planar regions and estimating their orientation based on training regions, as described in the following section.

4.4 Classification and regression

Regions are classified using a Relevance Vector Machine (RVM) [21] – since this is grounded in Bayesian statistics, it gives a probability of classification, which we find useful. Crucially – because it must be used hundreds of times per image – classification with the RVM is very fast, as once it has been trained only a small proportion of the data are retained. We verified, through cross-validation, that it is indeed much faster than the K-Nearest Neighbour (KNN) classifier used in [8], and scales well to larger training sets (as shown in figure 3) – which is important since accuracy improves with more training data.

The two-class plane/non-plane decision is handled by the standard RVM, but in order to regress orientation, we need the multivariate regression RVM [20]. This is able to simultaneously predict all three dimensions of the normal vector using the same set of relevance vectors. For both RVMs, we use a kernel function K_S based on a polynomial function of the spatiogram similarity measure ρ, since our experiments showed it to be superior to other alternatives: thus $K_S(\mathbf{S}^A, \mathbf{S}^B) = \sum_{q=1}^{Q} \rho(\mathbf{S}^A, \mathbf{S}^B)^q$, where the maximum power Q is set to 4.



(a) Classification accuracy (b) Classification time

Figure 3: Comparison of the KNN and RVM classifiers. Accuracy improves with more training data for both, but the RVM is much faster.

5 Detection

5.1 Sweeping

To detect the individual planar regions, we sweep a window over the image, sampling overlapping locations to identify potential planes. This is done by using each salient point in turn as the centre of a window, defined by including all points within some radius (50 pixels in

our experiments) – plane estimation is applied to each, discarding those whose classification certainty (from the RVM) is below a threshold, to retain only the most confident estimates. We empirically set the threshold to 0.8 in our experiments, which is generally sufficient to avoid using ambiguous regions during calculation of the local planarity estimate.

The plane estimates from these overlapping windows are accumulated at the points: each salient point i is given an estimate of its probability of belonging to the plane class, denoted by r_i, and of its orientation n_i, by combining the values for each sweep window in which it lies. We use the median for this (geometric median for vectors in $\mathbb{R}^3$) to give a degree of robustness to outliers. The process, as illustrated in figure 4, results in an estimate of the planarity and orientation at each point – the local plane estimate, figure 4(e) – which will be used to segment the points into distinct regions.

(a) (b) (c) (d) (e)

Figure 4: The window sweeping process: from the input (a) we take a small window to classify (b); this is swept across the image, covering it with overlapping windows (c) and ignoring those with low probability (d). Points are given a class and normal derived from all the windows in which they lie, resulting in a class and orientation estimate at each point (e).

5.2 Segmentation

We segment these points in two stages, first to separate planes from non-planes, then into planes of different orientations. These are formulated as Markov random fields (MRF) [14], on a graph formed from the salient points and the edges of a Delaunay triangulation.

To segment planes from non-planes, let p represent a configuration of the field, where each node $p_i \in \{0,1\}$ represents the class of point i (1 and 0 being plane and non-plane respectively). The goal of optimising the MRF is to find the optimal configuration p^*, defined as $p^* = \arg\min_p U(p)$, where $U(p)$ denotes the posterior energy of the plane-class MRF:

$$U(p) = \sum_{i \in S} V_1(p_i, r_i) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} V_2(p_i, p_j) \tag{1}$$

where r_i is the observation at point i (the estimated probability of belonging to a plane, being the median of all sweep estimates, as above), S is the set of all nodes, and $\mathcal{N}_i$ are the neighbours of node i. The functions V_1 and V_2 are the single site and pair site clique potentials respectively, defined as

$$V_1(p_i, r_i) = (p_i - r_i)^2 \tag{2}$$

$$V_2(p_i, p_j) = \delta_{p_i = p_j} \tag{3}$$

where $\delta_{p_i=p_j}$ takes the value 1 if p_i and p_j are equal, 0 otherwise. We initialise each p_i to the value in $\{0,1\}$ which is closest to r_i. The MRF is optimised using Iterative Conditional Modes (ICM) [3], which generally converges within a few iterations. In short, the purpose of optimising this MRF is to set each node to its most likely value (plane or not), given the smoothness constraint implied by its neighbours.

The plane-class MRF uses only two classes (plane and non-plane), making optimisation straightforward; however, the MRF for plane orientation must deal with a continuous range of normals. We make the assumption that the points belong to a finite number of planar surfaces, corresponding to neighbourhoods of points with the same normal. The orientation of these planes correspond to the modes in a density estimate of all the observed normals d_i (the geometric median of the sweep estimates for each point), which we find using mean shift [5] with a Gaussian kernel. Thus, plane segmentation is the process of deciding to which of these discrete planes each point should belong, given its observation and its neighbours.

The configuration of this MRF is represented by n, where $n_i \in \mathbb{R}^3$ is the 3D normal at node i; each n_i is initialised to the closest orientation mode, given its observation d_i. We wish to obtain $n^* = \arg\min_n E(n)$, where the posterior energy $E(n)$ is

$$E(n) = \sum_{i \in S'} F_1(n_i, d_i) + \sum_{i \in S'} \sum_{j \in \mathcal{N}_i} F_2(n_i, n_j) \tag{4}$$

and, using the function $\theta(\cdot,\cdot)$ to calculate the angle between two vectors in $\mathbb{R}^3$, the clique potentials are

$$F_1(n_i, d_i) = \theta(n_i, d_i) \tag{5}$$

$$F_2(n_i, n_j) = \theta(n_i, n_j) \tag{6}$$

The final step, after extracting the planar segments, is to run them through the plane estimation algorithm once more, to get a planarity and orientation estimate based only on the image area in question. However, the shapes of these segments are unlikely to be similar to the windows used for sweeping, and so a second pair of RVMs is trained for this task, obtaining appropriate training data by running the full plane detection algorithm on the original ground truth images. We find that using a separate classifier improves orientation accuracy by several degrees in the final output (obviously, the segmentation is unaffected).

6 Results

To evaluate the method, we collected a second set of ground truth images, from a different urban location, consisting of 63 manually segmented images – labelled as described in section 4.1. To evaluate classification accuracy, we consider the percentage of points in the image which are grouped into regions of the correct class. To assess orientation estimation, we take the angle between the normal of each detected plane and the mean orientation in the corresponding ground truth region (since detected regions will not necessarily lie entirely inside a single planar region). These values are displayed for all examples shown.

After running the plane detection algorithm on these images, we obtain a mean point classification accuracy of 88% (excluding the 6% of points which were not in any regions); the mean orientation error for all regions was 18.3°, which falls to 16.5° when considering only those regions which *do* lie entirely inside true planar regions (around half of them), indicating how well the algorithm can perform when segmentation is good. These results show that the method can perform well, giving a correct classification for most points. We believe the mean orientation accuracy of 18.3° to be very reasonable considering the difficulty of the task, and that the method makes no explicit use of geometric information. It also compares favourably with the value of 17.5° reported in [8], where all regions were known to belong to a single planar surface.

We show some typical examples of these test images in figure 5. Figure 5(b) illustrates successful disambiguation of non-coplanar surfaces, while 5(d) shows separation of planes

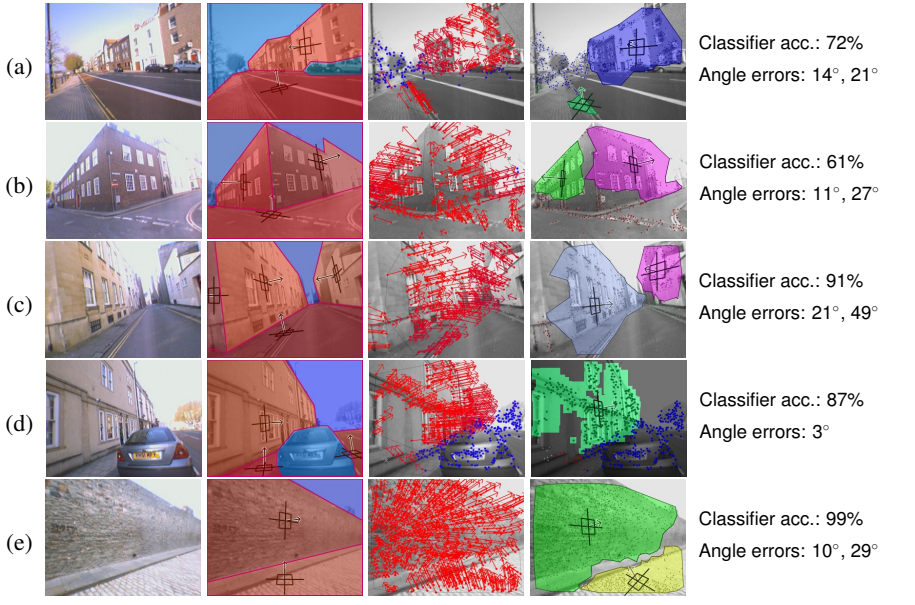


Figure 5: Example result of our algorithm. The four columns show, respectively, the input image, the hand-labelled ground truth, the estimates at each point calculated from the sweeping, and the final plane detection. Groups of planar points are enclosed in coloured regions, displaying their orientation; individual coloured points have been deemed non-planar. Classification accuracy over all points, and the orientation error for each region, are displayed.

from non-planes. Example 5(e) is important, since it shows plane orientation can be recovered in the absence of any obvious geometric structure. Again, we emphasise what is being achieved: these planes are being detected, with an estimate of their extent and 3D orientation, from just one image, without recourse to explicit geometric calculation. This is the novel contribution of our work.

Examples where our method performs badly are shown in figure 6. A common error is an inability to detect small regions, since fine details are difficult to observe using a region-based classifier. We also observed a tendency to group disparate planes together (e.g. figure 6(b)), or to over-segment the image (6(c)), when sparse or inaccurate orientation estimates are used for segmentation .

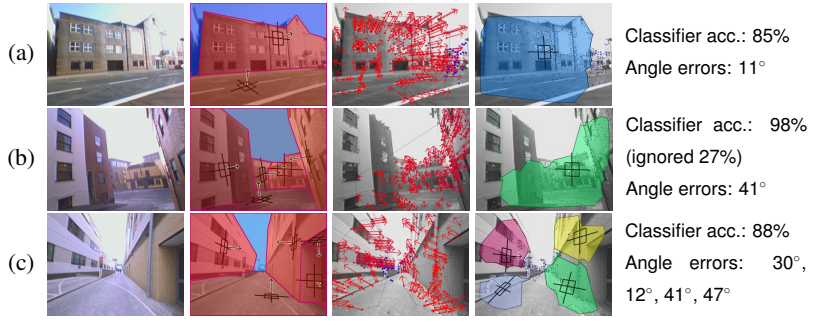


Figure 6: Examples where our method fails. Columns are as in fig. 5.

6.1 Comparison to prior work

The closest work to ours is Hoiem *et al.*'s algorithm for recovering surface layout [□] (which we refer to here as HSL). As we describe in section 2, this segments an image into geometric classes, where left, right, forward, and supporting surfaces are considered to be planar, and the others non-planar. Since we predict actual orientation, rather than discrete classes, we quantise our orientation estimates to these classes in order to compare the two – thus orientation error is now expressed as a classification accuracy. As above, we evaluate both methods by using the set of salient points (although these are not actually used as part of HSL).

We stress that HSL was not developed explicitly for plane detection – however, our intention here is to determine, if we use it to group points into planar regions, how well it performs. We tested this using code made available by the authors, which appeared to give reasonable results when applied to our dataset. The results are as follows: for classification accuracy, 71% of points are classified correctly, compared to 84% for our method; the mean orientation class accuracy for HSL is 68%, while our algorithm achieved 73%. As anticipated, given that our method is geared specifically toward plane detection, and uses a training set gathered solely for this purpose, it gives better results than HSL.

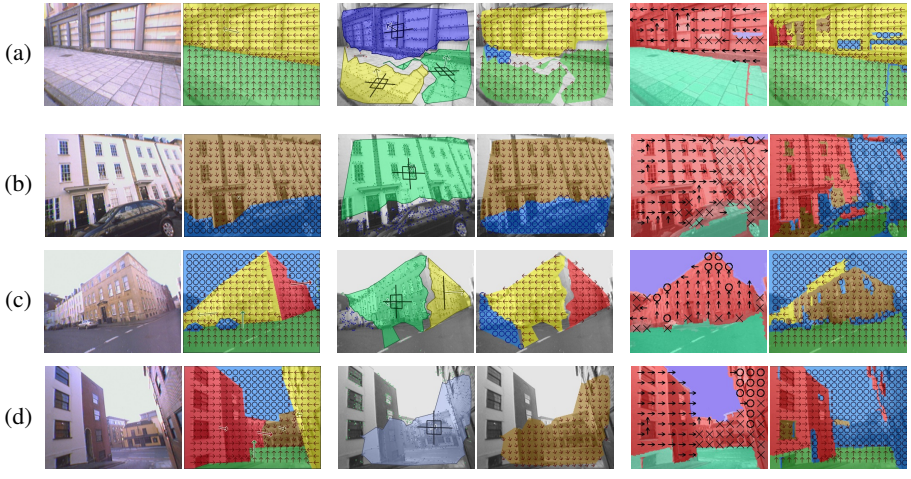


Figure 7: Comparison to the surface layout method of Hoiem *et al.* (HSL). Columns are: input image, ground truth, our method, our method drawn to show orientation classes, original output of HSL, and HSL drawn to show orientation classes.

Figure 7 shows some example results. We show the output of our method as before, and the standard output images of HSL (as described in [□]). To illustrate how we repurpose HSL for plane detection, all non-planar regions are shown in blue (circle symbol), and left, right, forward and supporting orientations are shown in yellow, red, brown and green respectively, with arrow symbols denoting orientation class. In this manner we draw the output of HSL, the quantised versions of our algorithm, and ground truth.

These images show that our algorithm performs similarly on some examples (such as figure 7(a)), and is occasionally superior – like 7(c) where we successfully disambiguate the two faces of a building; although here, as in many cases, we fail to detect the relatively featureless road, whereas HSL is very good at detecting road surfaces since it uses different types of feature. On others images, for example 7(d), HSL gives the better segmentation, since it is more able to perceive fine detail.

7 Conclusions

We have presented a new algorithm to detect planes, and estimate their 3D orientation, from a single image. Unlike existing methods, we do not rely upon rectilinear structure or vanishing lines, so that it is applicable to a wider range of scenes. We show that it is able to detect planes with good accuracy compared to labelled ground truth, and compares well to [11], the most similar existing work to ours.

Our method works by repeated sampling of windows from an image to recover individual planes; however, we have found that this makes the method unable to deal with small planes. An avenue of future work, therefore, would be to incorporate edge or discontinuity information, which has been shown to be beneficial in scene layout estimation [12]. It would be interesting to use a similar technique for relative depth estimation, which is possible from monocular cues [18, 22] – either to improve the fidelity of plane detection, or together with plane detection for more sophisticated interpretation of images.

Acknowledgements

We would like to thank David Hanwell and José Martínez-Carranza for useful discussions and comments. This work was funded by UK EPSRC.

References

[1] J. Aloimonos. Shape from texture. *Biological cybernetics*, 58(5):345–360, 1988.

[2] A. Bartoli and P. Sturm. Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *International Journal of Computer Vision*, 52(1): 45–64, 2003.

[3] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302, 1986.

[4] S.T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *Computer Vision and Pattern Recognitionn*, volume 2, pages 1158–1163, 2005.

[5] Y. Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799, 1995.

[6] S. Choi. Algorithms for orthogonal nonnegative matrix factorization. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1828–1832. IEEE, 2008.

[7] J. Gårding. Direct estimation of shape from texture. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1202–1208, 1993.

[8] O. Haines and A. Calway. Estimating planar structure in single images by learning from examples. *International Conference on Pattern Recognition Applications and Methods*, February 2012.

[9] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[10] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAIâĂŹ99*, page 21. Citeseer, 1999.

[11] D. Hoiem, A.A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[12] D. Hoiem, A.N. Stein, A.A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[13] J. Košecká and W. Zhang. Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding*, 100(3):274–293, 2005.

[14] S.Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York Inc, 2009.

[15] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.

[16] B. Micušık, H. Wildenauer, and M. Vincze. Toweards detection of orthogonal planes in monocular images of indoor environments. In *International Conference on Robotics and Automation*, May 2008.

[17] C. Ó Conaire, N.E. O'Connor, and A.F. Smeaton. An improved spatiogram similarity measure for robust object localisation. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages I–1069. IEEE, 2007.

[18] A. Saxena, M. Sun, and A.Y. Ng. Make3d: learning 3d scene structure from a single still image. *Transactions on Pattern Analysis and Machine Intelligence*, pages 824–840, 2008.

[19] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, 2003.

[20] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and R. Cipolla. Multivariate relevance vector machines for tracking. *Computer Vision–ECCV 2006*, pages 124–138, 2006.

[21] M.E. Tipping. Sparse bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

[22] A. Torralba and A. Oliva. Depth estimation from image structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1226–1238, 2002.