Estimating planar structure in single images by learning from examples

Osian Haines and Andrew Calway
Department of Computer Science, University of Bristol, UK
{*haines, andrew*}*@cs.bris.ac.uk*

Abstract: Outdoor urban scenes typically contain many planar surfaces, which are useful in tasks such as scene reconstruction, object recognition and navigation. Planar constraints are especially useful when only a single image is available, though the lack of 3D information makes finding them difficult; but a number of cues – such as rectangular shapes, edges, and appearance – can make this possible. We develop a method to determine if regions in an image are planar and find their orientation; motivated by how humans use their prior knowledge to help interpret new scenes, this is done by learning from a training set of examples. In contrast to previous methods which often rely on rectangular structure, this allows our method to generalise to a variety of outdoor environments, without relying on restrictive assumptions such as a Manhattan-like world or a camera aligned with the ground plane. From only one image, our method is able to reliably distinguish planes from non-planes, and estimate their orientation accurately; this is fast and efficient, with application to a real-time system in mind.

1 INTRODUCTION

We address the problem of detecting planes in a single image, and estimating their 3D orientation. Planar structures are useful because man-made urban environments tend to be dominated by them, allowing compact representation of 3D scenes (Bartoli, 2007) and more efficient robot navigation (Gee et al., 2008). The ability to discover planes from only a single image would be beneficial in tasks including image understanding (Saxena et al., 2008), reconstructing 3D models (Sturm and Maybank, 1999), and object recognition (Hoiem et al., 2007); or could be complementary to multi-view information in situations such as wide baseline matching (Mičušík et al., 2008), and in robot mapping where estimating geometry in real-time is difficult (Martínez-Carranza and Calway, 2010).

The lack of depth information makes understanding single images challenging, since any number of 3D scenes could have produced the image; fortunately, monocular cues exist which constrain the possible 3D configurations. Use of vanishing lines (Košecká and Zhang, 2005) and rectangular structures (Mičušík et al., 2008) are popular approaches, where the presence of regular, orthogonal structure (such as window frames) enables the geometry to be inferred. However, this can only work in scenes which contain such regularity; and even then, requires accurate detection of lines, which can be difficult to do reliably in general scenes.

Motivated by the way humans appear able to understand scenes from a single view without explicit geometric reasoning (Howe et al., 2006), we take an approach which exploits prior knowledge by learning from a training set (figure 1). This is inspired by techniques such as (Hoiem et al., 2007) and (Saxena et al., 2008), which take a machine-learning approach to interpreting single images, and so do not depend on explicit geometric properties. Our intention is to learn the relationship between plane structure and appearance, so that regions of the image are classified as planar if they correspond to structures such as building façades, pavements, and even stone walls, and an estimate is made of their 3D orientation relative to the camera; or classified as being non-planar for images of foliage, vehicles, and so on. We do this using basic feature descriptors in a bag of words representation, modified to include spatial information, so that we may find relevant training examples. Unlike (Hoiem et al., 2007) we assign an actual orientation estimate, rather than a coarse classification, and do not require assumptions about the camera pose.

We show that the method is able to accurately separate planes from non-planes, making a sufficiently confident decision in around 90% of cases, of which the correct class is predicted with 90% accuracy. For those regions correctly deemed to be planar, their orientation is predicted with a mean error of around 14° (between the estimated and true normal vectors). Since we do not rely on vanishing lines or rectangular structure, the method is applicable to a wider range of scenes. We also show that the method performs well in new environments by testing on an independent data set. The method is efficient and fast, being

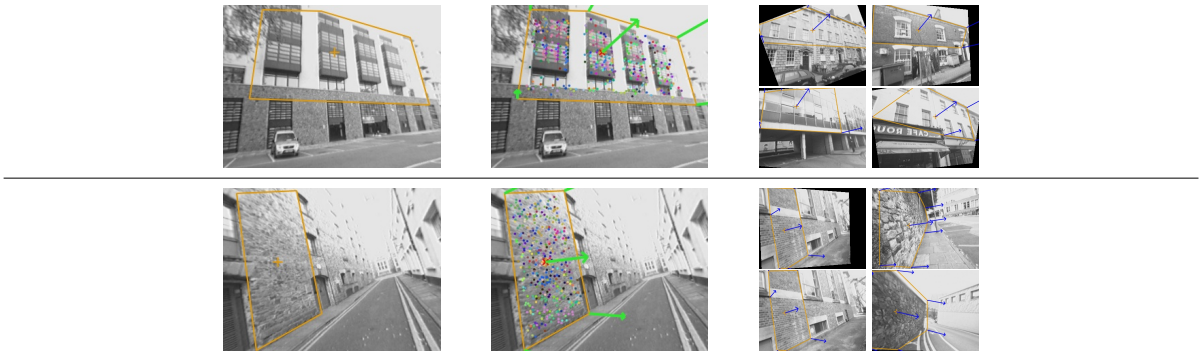


Figure 1: For a given image region (left) our algorithm classifies them as planes and estimates their orientation (centre) by finding training examples with similar orientation (right).

able to process and make a decision for a new region in much less than one second, though no code optimisation has been attempted. Presently we consider only the classification and orientation of regions, and assume a suitable area of the image is given as input; we leave automatic segmentation of an image for future work.

1.1 Overview

Our approach is summarised as follows. To learn the relationship between appearance and structure, we gather a database of manually labelled training examples. While anything can be considered planar at some level, for our purposes a planar region is one which is clearly planar on the scale of the image in question, so that it would be useful for the tasks described above. To classify regions (plane or non-plane), and estimate orientation (normal vector), we use a K-Nearest Neighbour classifier to select the most 'similar' examples. To achieve this, we must decide how to represent image regions, and how to measure the distance between them.

We base our representation on local image descriptors, corresponding to histograms of oriented gradients extracted at salient points across a region. These are not informative enough to predict the orientation of the underlying plane, so we accumulate information from across the region using a bag of words approach, where individual words represent basic visual primitives, and it is their combination which allows us to do classification. In order to deal with different appearance corresponding to the same orientation, we use a variant of Latent Semantic Analysis (Deerwester et al., 1990), which also serves to reduce the dimensionality of the representation. This takes the word histogram and creates a more compact representation in terms of 'topics'.

The occurrence of topics can discriminate planes from non planes and estimate orientation, but performance is improved by also considering their spatial configuration in the image. We represent spatial information by using a spatiogram (Birchfield and Rangarajan, 2005), which is a higher-order generalisation of a histogram, adding a mean and covariance for each bin. Comparison of regions using spatiograms significantly improves performance by retrieving more relevant neighbours for a test region. As far as we aware, using a spatiogram with a bag of words representation is novel.

Section 2 discusses related work concerned with finding planes in a single image. In section 3 we discuss the details of the method, including an explanation in section 3.4 of how spatiograms are created and compared. Our results in section 4 discusses the effect of different parameter choices, and shows that the method distinguishes planes from non-planes and reliably predicts their orientation in a number of different situations. We conclude in section 5 and outline directions of future work.

2 RELATED WORK

In general, planes have proven to be useful for 3D reconstructions (Bartoli, 2007), interactive modelling (Sturm and Maybank, 1999), and in visual mapping, for example to simplify the scene structure (Gee et al., 2008) and perform higher-level reasoning (Wangsiripitak and Murray, 2010). We describe a few important examples of single-image methods for finding planar structure, which use appearance information from the image as opposed to any 3D or multi-view information.

2.1 Single Image Geometry

A standard way to obtain geometry from a single image is the use of vanishing points (Criminisi et al.,

2000). One prominent example is (Košecká and Zhang, 2005), which relies on the orthogonality of planes to robustly group lines into three orthogonal directions in outdoor images. Rectangle hypotheses are formed from which the pose of the camera can be recovered, and used to make a simple reconstruction of the scene. Related is (Mičušík et al., 2008), which treats rectangle detection as a labelling problem on vertices and edges, and apply this to wide baseline matching by using the planes' orientation to rectify images before comparison.

(Barinova et al., 2008) use these ideas to create visually pleasing reconstructions of urban scenes; using vanishing points and the horizon, a polygonal line segment separating the ground from vertical structures is hypothesised, which completely describes a simplified scene geometry. Interestingly, a machine learning approach is taken to identify which line segments are valid, based on learned examples.

2.2 Learning From Images

Another cue which may be exploited is the distinctive appearance of certain parts of images. (Torralba and Oliva, 2002) use the fact that certain types of structure tend to appear at particular distances, in order to estimate the overall depth of an image. However, this method would not be able to estimate accurate depth for individual sections of an image.

An approach more similar to our own is that of (Hoiem et al., 2007), in which the orientation of planes are estimated without requiring obvious rectangular structure. This works by classifying 'superpixels' obtained by image over-segmentation into geometric classes, which coarsely describe the scene orientation for each region. These classes limit planes to being a 'support' surface (horizontal), or a left, right, or front facing vertical surface. A robust multiple-segmentation algorithm, using features including colour, texture, line length and explicit vanishing point information, allows homogeneous regions to be created from the initial super-pixels. The result is a segmentation of the general layout of a scene, which has been used to create simple 'pop-up' 3D models and as a prior for object recognition (Hoiem et al., 2006).

(Saxena et al., 2008) focus on depth estimation instead of plane detection, though the two problems are related. A full depth map for all pixels is estimated, after training the algorithm on range data from a laser scanner. Image regions are described by responses from a set of edge filters and Laws masks; local and global features represent absolute and relative depth cues, then a Markov Random Field is used to find a consistent depth map over the whole image. This has also been used for 3D model building, both from a single image and from a sparse set of images, and to guide a model car by avoiding obstacles (Michels et al., 2005).

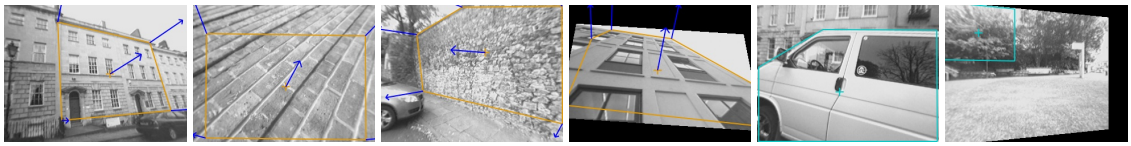
While the above methods show considerable progress in understanding single images, a number of issues remain to be addressed. Those which can describe the scene in terms of oriented planes use the assumption of a regular, 'Manhattan' world, which can be quite limiting. In many situations, such structure might not be present; and even if it is, image quality may be poor, and so we cannot rely on good line extraction. On the other hand, methods applicable to more general scenes can only obtain very coarse orientation, or estimate a depth map for every pixel, which is more computation than necessary for only finding planes. Below we propose an alternative that does not need line or rectangle detection or the presence of vanishing points, and can reliably identify planes and find their orientation in a variety of scenes, using the appearance information in the image.

3 METHOD

3.1 Training Data

So that we may learn about planar structure from the appearance of images, we gather a set of training examples consisting of planes and non-planes from a variety of outdoor locations. Images have a resolution of 320×240 pixels, and have been corrected for radial distortion. We obtain these using a hand-held webcam, and by treating this as a standard pin-hole camera we use the results of standard projective geometry: refer to (Hartley and Zisserman, 2003) for more details. The known intrinsic parameters (that is, principal point, focal length, aspect ratio and skew) of the camera are encoded by the 3×3 calibration matrix $\mathbf{K}$.

For each image we mark the region of interest, and assign them to the plane or non-plane class as appropriate. For the plane examples, a ground truth orientation is assigned using an interactive method, which requires marking corners of a quadrilateral in the image, corresponding to a real rectangle. This defines two orthogonal sets of parallel lines, whose intersections define vanishing points $\mathbf{v}_1$ and $\mathbf{v}_2$ (expressed as homogeneous 3-vectors), from which the vanishing line $\mathbf{l}$ of the plane is calculated: $\mathbf{l} = \mathbf{v}_1 \times \mathbf{v}_2$. This line defines a plane through the camera centre parallel to the scene plane, and whose normal can be obtained by $\mathbf{n} = \mathbf{K}^{\mathrm{T}}\mathbf{l}$. These regions are the training and testing



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

Figure 2: Examples of the training data we use, showing the manually selected region of interest. The orientation of the planar regions (a)-(d) is shown by superimposing a normal vector; examples (d) and (f) were obtained by warping the original images.

data for our method, and some examples are shown in figure 2.

3.1.1 Training Set Extension

To obtain a larger set of examples, we generate more from our initial marked-up set, by applying various geometric transformations. First, since the mirror image of a scene is equally plausible, we double our training set by reflecting all examples about the vertical axis; this has the additional advantage of removing any bias for left or right facing regions. Furthermore, using the known pose of the planar regions, we can render a new view from a different location, giving us many more examples from different directions. For a new viewpoint, represented by a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ relative to the original view, we find the homography $\mathbf{H}$ which relates the two views:

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{t}\mathbf{n}^{\mathrm{T}}}{d} \qquad (1)$$

where d is the perpendicular distance to the plane (all defined up to scale). We can use this to warp the original image $\mathcal{I}$ to create a new image $\mathcal{I}^w$, which approximates the plane as seen from the new viewpoint. This is achieved by setting the intensity of each pixel $\mathbf{v}^w$ in $\mathcal{I}^w$ to the intensity of the corresponding pixel $\mathbf{v} = \mathbf{H}^{-1}\mathbf{v}^w$ in $\mathcal{I}$. The boundary vertices are warped by multiplying by $\mathbf{H}$, and the normal $\mathbf{n}^w$ of the warped region is obtained by $\mathbf{n}^w = \mathbf{R}\mathbf{n}$. In practice, while almost any new orientation can be obtained in this way, this is limited by the image resolution, so we ensure that we do not create new views where the region is too stretched or foreshortened.

3.2 Features

Before creating a bag-of-words representation, we need to extract a set of basic feature descriptors. While (Saxena et al., 2008) and (Hoiem et al., 2007) use banks of filters and features such as colour, we instead follow more typical object recognition approaches and use descriptors that describe local orientations and gradients, which seems an important predictor of planarity and plane orientation.

3.2.1 Feature Descriptor

We use histograms of gradient orientations, calculated over local patches of the image; this is the basis of the SIFT (Lowe, 2004) and HOG (Dalal and Triggs, 2005) descriptors, which have found much success in tasks such as object recognition. However, we emphasise the difference between what we intend to achieve and typical object recognition or tracking: one of the benefits of SIFT is that it is invariant to a wide range of deformations, enabling recognition from a range of viewpoints; but our aim is specifically to determine orientation, irrespective of the identity of the plane, so use of SIFT or its variants is not appropriate.

For each patch, we create four gradient histograms, from each quadrant. Each of these has 12 angular bins (weighted by the gradient magnitude, and interpolating between bins) and is normalised to sum to 1, giving a descriptor of 48 dimensions (see figure 3). We do this in order to capture some local structure information and build a richer descriptor.

3.2.2 Feature Scale Selection

In real images, features occur at a variety of scales, and this is important here since the change in scale across a surface is a useful cue for its orientation. To deal with this, we use the Difference of Gaussians detector (DoG), commonly used as the first stage in SIFT, which also finds the appropriate scale of points. This gives an (x,y) location for each point, as well as a scale parameter σ, which we use to set the

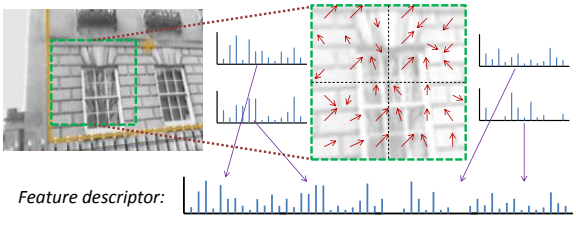


Figure 3: Creating a descriptor: for an image region (left) we find the gradient angle and magnitude for each pixel, and create a histogram for each quadrant (right), which are concatenated to create the final descriptor (bottom).

width ω of the patch to create the descriptor, setting $\omega = \max(\sigma, 10)$ and discarding patches where $\sigma \geq 60$, since descriptors this large no longer represent local information.

We compared this to single scale feature detection using FAST (Rosten and Drummond, 2006), and found DoG detection to be consistently superior (see the results section). Interestingly, we also found that DoG detection out-performed FAST for any single fixed scale, suggesting the blob-like features detected by DoG are more appropriate than FAST's corner-like features.

By selecting the scale for each feature, we are making the method invariant to scale, rather than explicitly making use of it. This is advantageous since it ensures the most appropriate scale is being used at each location, though it would be interesting to include such scale information in the representation, to represent scale change across a region; this is left for future work.

3.3 Bag of Words

The gradient descriptors capture information about local areas, but are not sufficient to disambiguate the structure of the scene, and so we accumulate information over the whole region. We use the bag of words model, adapted from the text retrieval literature, where documents are summarised simply by the count of word occurrences, disregarding all syntax and context. When applied to vision, each image is analogous to a document, and a discrete set of 'visual words' are the analogue of words or terms in a text document (Sivic and Zisserman, 2003).

The bag of words representation uses a single vector $\mathbf{x}$ for each image region, where $\mathbf{x} = \{h_n | n = 1...N\}$ is a histogram over N words. These words are found by quantising each of the D descriptor vectors $\mathbf{d}_d$ in the image to a codebook, and so $\mathbf{x}$ is calculated using

$$h_n = \sum_{d=0}^{D} \delta_{dn} \tag{2}$$

where δ_{dn} is 1 iff $\mathbf{d}_d$ quantises to word n.

To create the codebook, we use 66 images of typical outdoor urban scenes, representing both planar and non-planar structures. Descriptors are created (section 3.2), and we use K-means to cluster these to create a codebook (typically N = 300; we discuss the effect of vocabulary size in section 4). Finally, we use term frequency – inverse document frequency weighting (tf-idf) to create the weighted word vector $\mathbf{x}'$, so that common words are down-weighted and all documents are scaled according to their size (Sivic and Zisserman, 2003).

3.3.1 Topic Discovery

With a large number of possible words, but a limited number of feature points per image region, the word vector is both high dimensional and sparse. Moreover, word synonymy (different words characterising similar structure) cannot be captured with only a word vector. One way to overcome this is to use Latent Semantic Analysis (LSA) (Deerwester et al., 1990). This is essentially a dimensionality reduction technique, which finds hidden semantic meaning amongst the words and reduces the word histogram to a vector of topic weights. Instead of the standard LSA formulation we use Orthogonal Nonnegative Matrix Factorisation (ONMF), because this gives us a topic vector which has only non-negative components (which will be important later), and since it allows us to easily obtain new topic vectors by projection of word vectors.

Like LSA, ONMF performs a factorisation of the term-document matrix $\mathbf{X} = [\mathbf{x}'_0, \mathbf{x}'_1 .. \mathbf{x}'_M]$, in which the element $\mathbf{X}_{nj}$ is the (weighted) number of occurrences of word n in document j; $\mathbf{X}$ is decomposed as $\mathbf{X} \approx \mathbf{WH}$, where $\mathbf{W}$ contains the basis vectors of the latent topic space, and $\mathbf{H}$ contains the topic weights for each document. For M documents and N words, $\mathbf{X}$ is N by M, $\mathbf{W}$ is N by T and $\mathbf{H}$ is T by M, where T is the number of topics. Each word vector from $\mathbf{X}$ is approximated by $\mathbf{x}'_i \approx \mathbf{W}\mathbf{h}_i$, where $\mathbf{h}_i$, a column of $\mathbf{H}$, is a topic vector, the lower dimensional representation of $\mathbf{x}'$. To find the topic vector $\mathbf{h}_q$ for new word vector during testing, we make use of the orthogonality of $\mathbf{W}$ and rearrange the above to obtain $\mathbf{h}_q = \mathbf{W}^{\mathrm{T}}\mathbf{x}_q$.

ONMF factorisation has no closed form solution, so we use an iterative method, which is guaranteed to monotonically decrease the reconstruction error $\|\mathbf{X} - \mathbf{WH}\|$ and converge to a local minimum (Lee and Seung, 2001). We use the multiplicative update algorithm proposed by (Choi, 2008), and re-normalise the columns of $\mathbf{W}$ to unit norm after each iteration:

$$\mathbf{W}_{nt} \longleftarrow \mathbf{W}_{nt} \frac{(\mathbf{X}\mathbf{H}^{\mathrm{T}})_{nt}}{(\mathbf{W}\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{W})_{nt}} \tag{3}$$

$$\mathbf{H}_{tm} \longleftarrow \mathbf{H}_{tm} \frac{(\mathbf{W}^{\mathrm{T}}\mathbf{X})_{tm}}{(\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{H})_{tm}} \tag{4}$$

3.4 Spatiograms

In order to improve the performance of the method outlined above, we include spatial information about the distribution of topics, which helps to disambiguate plane orientations. While the constellation and star models (Fergus et al., 2005) allow spatial information

to be taken into account, these are very computationally expensive and do not scale well to large numbers of parts. Instead we use the spatiogram (Birchfield and Rangarajan, 2005), which is a higher-order generalisation of the histogram – instead of just a count for each bin, a spatiogram also represents the mean position and covariance matrix for the points contributing to each bin; they have been shown to outperform histograms in tasks such as object tracking (Birchfield and Rangarajan, 2005) and object detection (Ó Conaire et al., 2007).

So to replace the word histogram $\mathbf{x}$, we create a word spatiogram $\mathbf{S}^{\text{word}}$, comprised of a set of triplets

$$\mathbf{S}^{\text{word}} = \{s_n | n = 1 \dots N\} \tag{5}$$

$$s_n = \langle h_n, \mu_n, \Sigma_n \rangle \tag{6}$$

where h_n is the histogram component as above, and μ_n and Σ_n are respectively the mean and covariance matrices of the 2D coordinates for points contributing to the histogram bins. We modify the original equations for calculating the spatiogram to obtain the weighted unbiased estimate for the covariance:

$$\mu_n = \frac{1}{\alpha} \sum_{d=1}^{D} \mathbf{v}_d \delta_{dn} \tag{7}$$

$$\Sigma_n = \frac{\alpha}{\alpha^2 - \beta} \sum_{d=1}^{D} (\mathbf{v}_d - \mu_n)(\mathbf{v}_d - \mu_n)^{\mathrm{T}} \delta_{dn} \tag{8}$$

where $\mathbf{v}_d$ is the 2D point at which descriptor $\mathbf{d}_d$ is created, and $\alpha = \sum_{d=1}^{D} \delta_{dn}$, $\beta = \sum_{d=1}^{D} \delta_{dn}^2$. This allows us to calculate spatiograms for the tf-idf weighted word histogram $\mathbf{x}'$ or reduced-dimension topic vector $\mathbf{h}$, denoted by $\mathbf{S}^{\text{word}\prime}$ and $\mathbf{S}^{\text{topic}}$ respectively, by reweighting the contribution of each descriptor $\mathbf{d}_d$ to each bin. To create $\mathbf{S}^{\text{word}\prime}$, replace δ_{dn} above with the quantity

$$\delta'_{dn} = \frac{x'_n}{x_n} \tag{9}$$

where x_n and x'_n are the values for word n in $\mathbf{x}$ and $\mathbf{x}'$ respectively, i.e. δ'_{dn} is the weighting for one occurrence of word n in the image. When using topics, instead of contributing to one word only, each descriptor makes a contribution to each topic, and so we calculate $\mathbf{S}^{\text{topic}}$ (having length T) by replacing δ_{dn} with ξ_{dt}, which is the contribution of each descriptor $\mathbf{d}_d$ to topic t:

$$\xi_{dt} = \frac{x'_n}{x_n} \mathbf{W}_{nt} \tag{10}$$

where n is the word to which descriptor $\mathbf{d}_d$ quantises, and $\mathbf{W}_{nt}$ is the component of the basis vector for topic t relating to word n. Note that the requirement that all weights be positive is the reason we use ONMF instead of LSA.

To use these for classification, we use a distance metric on the space of spatiograms proposed by (Ó Conaire et al., 2007). For two general spatiograms $\mathbf{S}$ and $\mathbf{S}'$ of dimension P, this similarity is defined as

$$\rho = \sum_{p=1}^{P} \sqrt{h_p h'_p} 8\pi |\Sigma_p \Sigma'_p|^{\frac{1}{4}} \mathcal{N}\left(\mu_p; \mu'_p, 2(\Sigma_p + \Sigma'_p)\right) \tag{11}$$

We found that using spatiograms significantly improved the performance for both classification and orientation estimation, suggesting that the relative position of features in the image is relevant for plane recognition. We tried spatiograms representing the image position both with absolute position in the image (thus, like (Hoiem et al., 2007), using position in the image as a feature), and with respect to the region position, such that the representation is translation invariant; we found absolute location to be slightly beneficial but not essential. It may be thought that the benefit of spatiograms is simply because they represent the shape of the pre-segmented region; we tested this by constraining all regions to be the same shape, but found that spatiograms still out-perform histograms.

3.5 Classification

Finally, once we obtain the spatiograms for the image regions, we can use this to classify them and estimate an orientation. We opt for a relatively simple means of classification – the K-Nearest Neighbour classifier (KNN). We do this to verify that the method is making use of the features and spatial information described above, and is not reliant on a sophisticated classifier; and also because it is easy to interpret the decisions of the KNN, by looking at which neighbours were chosen (figures 1, 8). The two tasks – classification to plane or non-plane and orientation estimation – can be performed simultaneously, by finding the K nearest neighbours: the class is assigned to the majority class of these, and the orientation of planes is the mean of the orientation, expressed as a 3D unit normal vector. This also allows us to place a confidence value on the classification, corresponding to the proportion of neighbours in the majority class; we can then use a threshold on this to reject regions with uncertain classification.

We might alternatively separate the two tasks, using separate classifiers for each, since extracting topics on the planes and non-planes together does not necessarily find the best set of topics for distinguishing plane orientation. However, when testing without

non-planar examples, we saw no significant change in orientation accuracy.

4 RESULTS

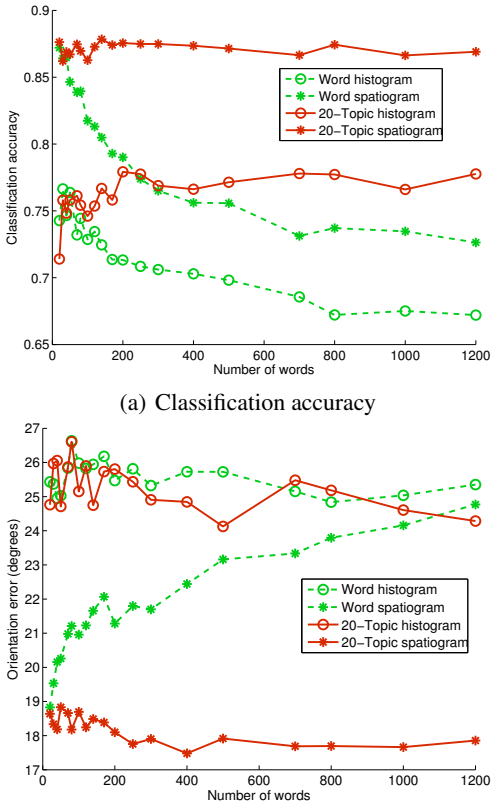
The data set we use for evaluating the method was obtained from an urban area, as described in section 3.1, totalling 556 regions. The dataset is first reflected, to give 1112 basic regions, then warped, giving us a final total of 7752. Five-fold cross validation is used, with the regions grouped to ensure that a test example can never be matched to a warped version of itself; we do not test on warped regions since they are not necessarily realistic test images.

First, we analyse the performance using topic discovery and spatial information as compared to the basic bag of words model. To do this, we ran the algorithm using the (weighted) word histograms $\mathbf{x}'$ only, on word-spatiograms $\mathbf{S}^{\text{word}\prime}$, on topic vectors $\mathbf{h}$ only, and on topic spatiograms $\mathbf{S}^{\text{topic}}$ (the full method), for varying vocabulary size (histograms are compared using the Bhattacharyya coefficient). Figure 4 shows both classification accuracy and orientation error – in general, using topic discovery out-performs using words directly, and as the vocabulary size increases, performance using word histograms decreases as they become increasingly sparse; but performance using topics remains almost constant, showing that it is able to extract meaningful information from high dimensional word vectors (we found similar performance when using varying numbers of topics, from 20 to 100).

Interestingly, the results suggest it may be possible to use words directly, without topic discovery, with a very small vocabulary (i.e. 20 words). However, this makes the method less flexible by constraining it to use only small vocabularies; and it seems less likely that small vocabularies would generalise well to new data sets. We tested this on our independent data set (see below) and found that in this case, using words instead of topics decreased mean orientation accuracy from 17.5° (with standard deviation 15.9°) to 20.5° (standard deviation 18.1°).

The graphs also clearly show the benefit of using spatial information, which outperform histograms in all cases, and appears more robust to change in number of words.

Next, we show the increase in performance gained by adding more warped training data (figure 5), while changing the number of nearest neighbours used. Using warped data does not improve performance (in terms of orientation accuracy) when using only topic histograms; but for spatiograms, using warped regions improves accuracy by several degrees on average. This graph also confirms that performance is improved by using spatial information; and that performance is fairly stable for different K.



(a) Classification accuracy

(b) Orientation error

Figure 4: Comparison of words and topics for different vocabulary sizes.

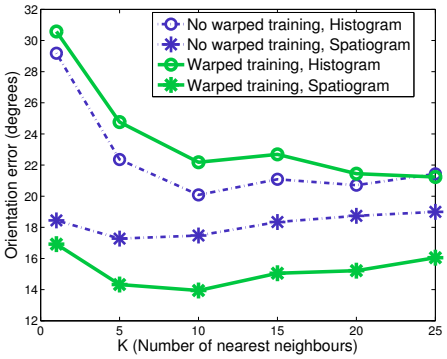


Figure 5: Orientation estimation for varying K, showing the difference in performance due to using more (warped) training regions, when using histograms or spatiograms.

We also ran an experiment to verify that using scale selection for the features is important. To ensure that no one scale was the best with scale selection simply choosing this occasionally, we tested scale se-

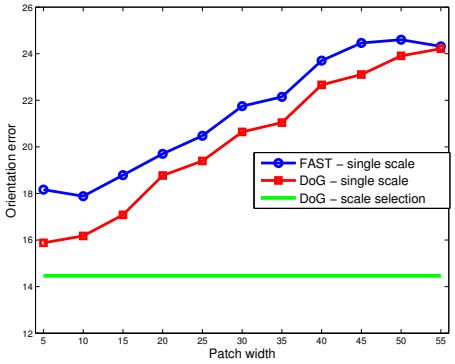


Figure 6: Using the Difference of Gaussians detector to choose the scale at which descriptors are built outperforms any single fixed scale, detected with either DoG or FAST.

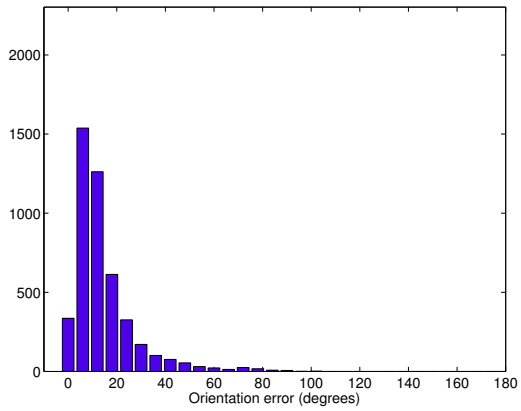
lection (using the DoG detector) against fixed patch sizes in the range $\omega = 5$ to 55, detected with both FAST and DoG; as figure 6 shows, scale selection is always better than any one scale.

For the remaining tests, we decide upon using a value of $K = 5$, with spatiograms using absolute position information, feature scale selection, and warped training examples, on a vocabulary of 300 words, and we discard regions with a confidence below 0.7. The results we obtain for this situation is a recall (percentage of regions above the confidence threshold)of 91%, classification accuracy of 90%, and a mean orientation error of 14°. Figure 7(a) shows a histogram for orientation estimation, clearly showing that for the majority of regions (81%), the error is in the region of 0° to 20°.

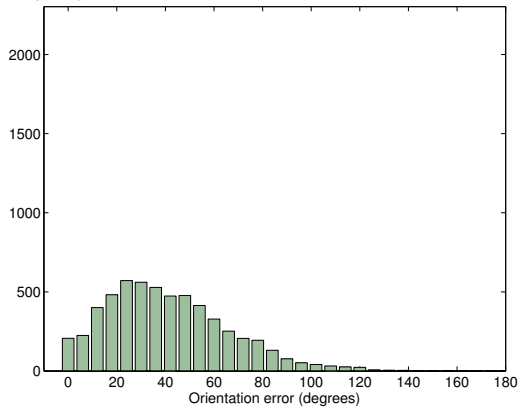
To get some perspective on what an error of 14° actually means, we ran an experiment where the neighbours are selected randomly. This is a useful validation of the method, as it shows that our algorithm is finding appropriate features in the appearance of the images in order to find relevant neighbours, as opposed to exploiting an artefact of how the data are distributed. Comparing to the histogram in figure 7(b), our method is clearly performing better than would be expected if it were choosing randomly, where the mean error is above 40°.

4.1 Independent Data

We also tested the algorithm on an independent data set, collected from a different urban area, comprising 538 regions. The data set used above was used for training (including all warped and reflected regions), with the same parameters. We achieved similar performance – a recall of 91%, classification accuracy of 87%, and mean orientation error of 17.5°. This set included some challenging examples, such as images



(a) Distribution of errors for our method, showing the majority of errors are small.



(b) Performance when using random neighbours.

Figure 7: Distribution of errors for estimating orientation.

of pavements and roads, since we were careful to include *no* images of the ground in the training set (figures 8 bottom, 9(f),9(g)); and difficult regions such as textured walls (figures 1,9(d)). Figure 8 shows some example results of orientation estimation, alongside their nearest neighbours: while these are quite different in appearance, they have a similar orientation, a quality which allows the method to generalise to new data. Figure 9 shows further examples, including regions correctly deemed to be non-planar. In these images, blue (thin) arrows indicate ground truth, and green (thicker) arrows are the estimated normal, with cyan circles denoting non-planar classification.

Figure 10 shows cases where the method performs poorly, both in terms of inaccurate orientation and misclassification. 10(a) and 10(b) are examples of the (comparatively rare) case where all the matching planes have very different orientation, something which requires further investigation. Figure 10(d) shows one of the more difficult examples, being very different from anything in our training set. Figure 10(g) may be confused by the railings, vertical trees

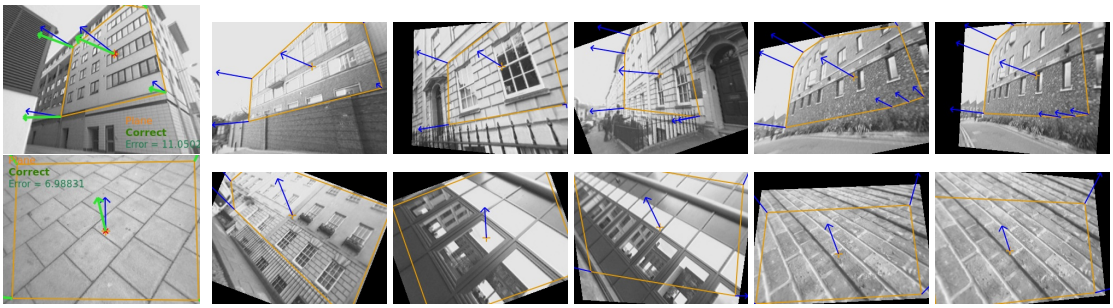


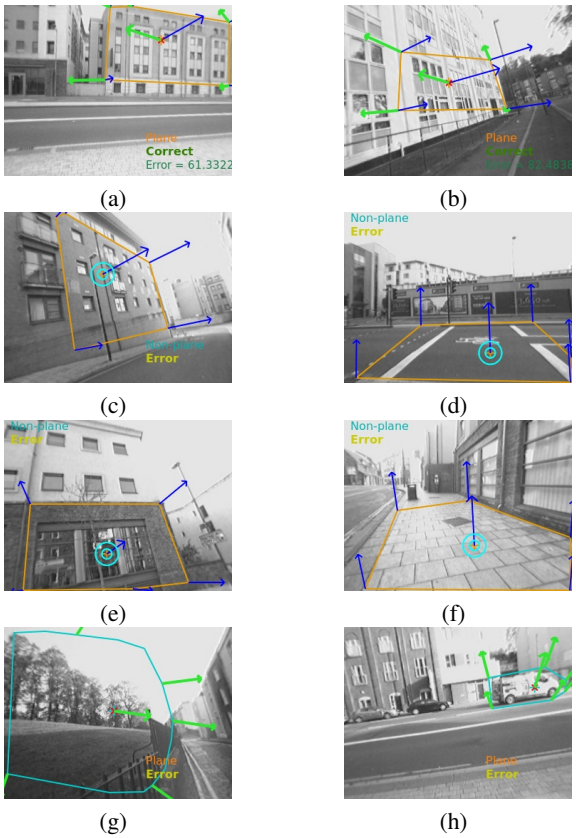
Figure 8: Examples of test planes (far left) and their 5 nearest neighbours. Top: a typical building front, matching to neighbours of rather different appearance. Bottom: correct orientation found for a pavement, even though no such images are in the training set.

and strong horizon line; and it is interesting to note that 10(h) is incorrectly determined to be a plane, when the side of a van could arguably be considered planar.

5 CONCLUSIONS

We have shown that we can reliably determine the planarity of regions – a classification to a plane or a non-plane class – and their orientation with respect to the viewpoint. This is achieved using only the information from one image, in the form of gradient-based descriptors and accumulating information over the region using the bag of words representation. Improved performance is achieved by incorporating information about the spatial distribution of topics across the image region, using a spatiogram. Using a KNN classifier, we demonstrate that the algorithm is capable of classifying a wide variety of plane and non-plane images, generalising well to new data; and to be able to accurately estimate plane orientation, even in examples devoid of typical structure such as lines, vanishing points and images of rectangles.

Future work will involve improving the classifier, both to give more accurate results, and also to be more scalable to larger training sets. Now that we have shown this kind of single-image plane identification to be possible, we intend to use our algorithm to automatically segment planar regions from images. However, since we operate on whole regions as opposed to using local colour or edge information, this will require a different approach to typical image segmentation.



(a) (b) (c) (d) (e) (f) (g) (h)

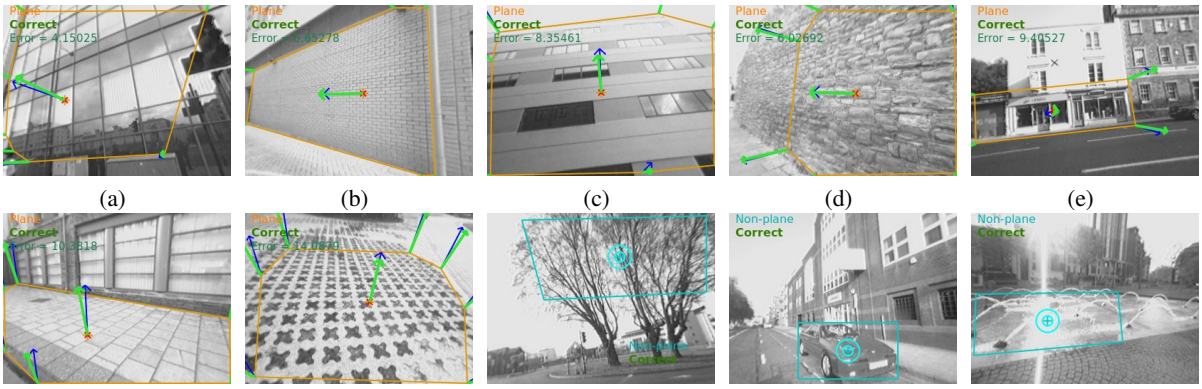
Figure 10: Where the method fails: (a)-(b) show planes with incorrect orientation estimate, whereas (c)-(f) are false negatives and (g)-(h) are false positives for plane detection.

ACKNOWLEDGEMENTS

This work was funded by UK EPSRC. With thanks to José Martínez-Carranza and Sion Hannuna for useful discussions and advice.

REFERENCES

Barinova, O., Konushin, V., Yakubenko, A., Lee, K., Lim, H., and Konushin, A. (2008). Fast automatic single-view 3-d reconstruction of urban scenes. In *European Conference on Computer Vision*.



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

(f) (g) (h) (i) (j)

Figure 9: Examples of (a)-(g) planes with good orientation estimates and (h)-(j) correctly classified non-planes.

Bartoli, A. (2007). A random sampling strategy for piecewise planar scene segmentation. *Computer Vision and Image Understanding*, 105(1).

Birchfield, S. and Rangarajan, S. (2005). Spatiograms versus histograms for region-based tracking. In *Computer Vision and Pattern Recognitionn*, volume 2.

Choi, S. (2008). Algorithms for orthogonal nonnegative matrix factorization. In *International Joint Conference on Neural Networks*. IEEE.

Criminisi, A., Reid, I., and Zisserman, A. (2000). Single view metrology. *International Journal of Computer Vision*, 40(2).

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, volume 1.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for Information Science*, 41(6).

Fergus, R., Perona, P., and Zisserman, A. (2005). A sparse object category model for efficient learning and exhaustive recognition. In *Computer Vision and Pattern Recognition*, volume 1.

Gee, A., Chekhlov, D., Calway, C., and Mayol-Cuevas, W. (2008). Discovering higher level structure in visual slam. *Transactions on Robotics*, 24.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press.

Hoiem, D., Efros, A., and Hebert, M. (2006). Putting objects in perspective. In *Computer Vision and Pattern Recognition*, volume 2, pages 2137–2144.

Hoiem, D., Efros, A., and Hebert, M. (2007). Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1).

Howe, C., Beau Lotto, R., and Purves, D. (2006). Comparison of bayesian and empirical ranking approaches to visual perception. *Journal of Theoretical Biology*, 241(4).

Košecká, J. and Zhang, W. (2005). Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding*, 100(3).

Lee, D. and Seung, H. (2001). Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*.

Martínez-Carranza, J. and Calway, A. (2010). Unifying planar and point mapping in monocular slam. In *British Machine Vision Conference*.

Michels, J., Saxena, A., and Ng, A. (2005). High speed obstacle avoidance using monocular vision and reinforcement learning. In *International Conference on Machine learning*.

Mičušík, B., Wildenauer, H., and Košecká, J. (2008). Detection and matching of rectilinear structures. In *Computer Vision and Pattern Recognition*.

Ó Conaire, C., O'Connor, N. E., and Smeaton, A. F. (2007). An improved spatiogram similarity measure for robust object localisation. In *International Conference on Acoustics, Speech and Signal Processing*, volume 1.

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. *Lecture Notes in Computer Science*, 3951.

Saxena, A., Sun, M., and Ng, A. (2008). Make3d: learning 3d scene structure from a single still image. *Transactions on Pattern Analysis and Machine Intelligence*.

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

Sturm, P. and Maybank, S. (1999). A method for interactive 3d reconstruction of piecewise planar objects from single images. In *British Machine Vision Conference*.

Torralba, A. and Oliva, A. (2002). Depth estimation from image structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9).

Wangsiripitak, S. and Murray, D. (2010). Reducing mismatching under time-pressure by reasoning about visibility and occlusion. *British Machine Vision Conference*.