ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Extraction and Recognition of Planes in a Single Image Using Matlab

T.Ravi Chandra Babu
Associate Professor & HOD,
Department of ECE,
Krishnamurthy Institute of Technology and Engineering.

Jarupula Ramesh
PG Scholar-SSP,
Department of ECE,
Krishnamurthy Institute of Technology and Engineering.

Abstract:

We present a novel method to recognize planar structures in a single image and estimate their 3D orientation. This is done by exploiting the relationship between image appearance and 3D structure, using machine learning methods with supervised training data. As such, the method does not require specific features or use geometric cues, such as vanishing points. We employ general feature representations based on spatiograms of gradients and color, coupled with relevance vector machines for classification and regression. We first show that using hand-labeled training data, we are able to classify pre-segmented regions as being planar or not, and estimate their 3D orientation. We then incorporate the method into a segmentation algorithm to detect multiple planar structures from a previously unseen image.

INTRODUCTION:

This project is concerned with the automatic extraction of 3D structure from single images. While the creation of 3D models of real-world scenes from image data has been a topic of interest for a long time, it is usual for this to involve either multiple views of a scene or video data, exploiting parallax to obtain information about scene depth. Inferring depth from only a single image is much more challenging. However, previous works have shown that a number of image cues can be exploited to extract information about depth, shape, or other 3D structure.Two prominent existing methods are those of Saxena et al. [32] and Hoiem et al. [19]. The former is able to recover an approximate depth map for an image having learned the relationship between image appearance and ground truth depth maps derived from laser scanning.

RELATED WORK:

Here we discuss examples of prior work on extracting planar structure from single images. This can be divided into two main categories: methods which explicitly use geometric properties, such as parallel lines and texture; and work which aims to recognize structure based on learning from training examples. Given two or more sets of parallel lines lying on a plane, their respective vanishing points uniquely define the plane’s 3D orientation [17]. Thus, detecting such line features in an image enables the extraction of structure, providing they lie on a common plane. One approach is to detect rectangular structures, such as windows or doors, which provide orthogonal parallel lines in the same plane [22], [27], which can be used for basic camera pose recovery and wide baseline matching. However, these methods rely on orthogonal, Manhattan-like structure, and reliable line detection, hence limiting applicability.

Machine Learning Methods:

More recent work has looked at techniques for learning the relationship between appearance and structure. A good example of such a technique is by Torralba and Oliva [35], who estimate overall depth using knowledge that certain types of structure tend to appear at particular distances. However, this work focuses on global scene properties, which is a level of understanding too coarse for most interesting applications. Saxena et al. [32] go further than this by estimating whole-image depth maps based on training images labeled with absolute depth.

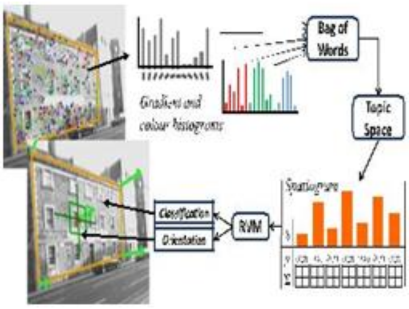


Fig.2: Main components of the plane recognition algorithm.

They can create basic 3D models of the scene, built from locally planar facets, and their comparison to ground truth depth shows good accuracy. However, the resulting models do not explicitly represent higher-level structures—The super-pixel segments are all assumed to be locally planar, and accuracy of planar facet orientations is not reported. Rather, the focus of the work is to produce visually plausible renderings of the scene, which are assessed by human subjects. This is in contrast to the work we present here, where we explicitly aim to find large-scale planes in the image, and to assign them an accurate orientation.

PLANE RECOGNITION:

In this section we describe the plane recognition algorithm, which classifies image regions as being planar or not, and for the former provides a 3D orientation estimate. We emphasize that this works on individual, pre-segmented image regions only, and does not apply to the image as a whole; marking the relevant planar or non-planar image region is part of the data acquisition process. The main components are shown in Fig. 2.

Training Set:

The plane normal can then be obtained from n ¼K^Tl, where K is the 3 _ 3 intrinsic camera calibration matrixes [17]. It is this relationship between camera parameters and plane orientation that makes it possible to recover accurate plane orientations for new data, because the relationship between image appearance and orientation is constant.

Fig: Examples of hand-segmented regions and their ground truth orientation (rightmost image is obtained by warping)

This requires a consistent and known camera calibration for all images used, which is consistent with our intended application area of SLAM and 3D reconstruction, but makes running our algorithm on other existing datasets problematic. To further increase the size of training set, we synthetically generate new variations from the marked-up set, first by reflecting all the regions about the vertical axis; then we generate examples of planes with different orientations by warping the regions—effectively simulating the view as seen by a camera in different poses [17]. Examples of training data are shown in Fig. 3.

Salient Points:

An image region will generally contain a large amount of visual information, as well as potentially less informative blank regions. To create a more compact representation, and focus on parts of the image which are more likely to be useful, we select a subset of salient points in the image around which to concentrate further processing. This is achieved with the difference of Gaussians (DoG) saliency detector [24], which selects blob-like regions in the image.

Image Descriptors:

Image descriptors are created in the region about each salient point, where the region size is dictated by the scale returned by the saliency operator. We use two complementary feature descriptors: the first is gradient orientation histograms to describe texture, which consist of histograms of edge orientation, computed by applying edge filters to the image. We create four histograms per patch, one for each quadrant, comprised of 12 angular bins covering the range ½0; pÞ, and concatenated to give a 48D descriptor. Secondly, we represent color using RGB histograms, created by concatenating intensity histograms from the red, green and blue channels of the patch. Each has 20 bins, giving a 60D descriptor. The importance of color for classifying structure was demonstrated by [19], and as we hoped, combining both types of descriptor gives superior performance to either in isolation (see Section 6.1.1). However, color is not beneficial for estimating orientation, and so we maintain separate representations for classification and regression, the former comprising gradient and color information, the latter with gradient only.

Bag of Words:

To further reduce dimensionality, we represent the distribution of descriptors in a region using a bag of words approach [25]. We identify clusters in descriptor space and their centers then form 'visual words', creating a 'vocabulary' codebook. Two separate vocabularies are created, to represent the gradient and color descriptor spaces—created by running K-means on a set of 100 representative images. Regions are then compactly represented by a pair of word histograms, expressing the occurrence ofgradient and color words from the respective vocabularies.

PLANE DETECTION:

In this section, we describe the plane detection algorithm, which identifies planar regions in images and estimates their orientation. As illustrated in Fig. 4, we do this by applying the plane recognition algorithm at different locations over the image, and use this to estimate planarity at individual points, and after clustering and smoothing we are able to extract individual planar structures. Example results from each stage are shown in Fig. 5. The recognition algorithm, as explained above, uses a discrete set of points, and so the density of such points determines the segmentation resolution that we can expect. In effect, we aim to group these points into planar and non-planar regions based on their spatial adjacency and compatibility in terms of planar characteristics; since these points are the cancroids of the image patches used to describe the image, our point-based segmentation is effectively segmenting the image itself.

Location Sampling:

The first stage of the algorithm is to apply plane recognition (Section 4) at multiple overlapping locations in the image, to sample possible locations where planes might be.

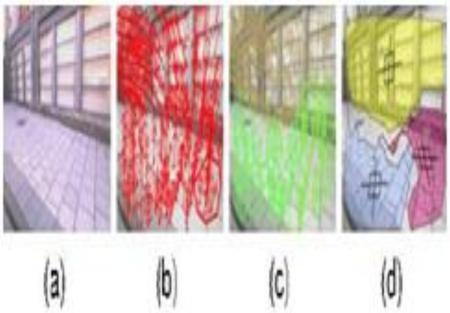


Fig. Outputs from plane detection: from the input image (a), we apply plane recognition over the image to obtain a point-wise estimate of orientation (b). This is segmented into distinct regions (c), from which the final plane detections are derived (d).a set of up to 100 regions per image, centered over a subset of points. These regions are circular with a fixed radius (50 pixels in the experiments) and all points within such a region are used as input to one invocation of PR, giving planar/non-planar classifications (and orientation estimates) at these locations.

Segmentation:

The goal of the segmentation stage is to take the points in the local plane estimate (above), and separate them into distinct planar or non-planar regions. This is achieved in three steps: first, to cluster the labels assigned to points to obtain a discrete set of assignable labels; assign each point its most likely label; and then to extract connected regions of points with the same label. While a number of segmentation algorithms could be employed to achieve this, the problem is

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

naturally expressed as a simple MRF on a graph connecting the points. The segmentation of planes from non-planes, and into planes of different orientations, is done separately, since different criteria are used for the two stages (classification probabilities and orientation estimates, respectively), and they act on different sub-graphs of the point set (orientations are usually not defined for regions deemed non-planar). A joint segmentation should be possible, but we present details of the simpler model here.Subdividing planar regions based on their orientation estimates is a little more complicated, as the labels belong to the continuous range of normal vectors, rather than simply two classes.

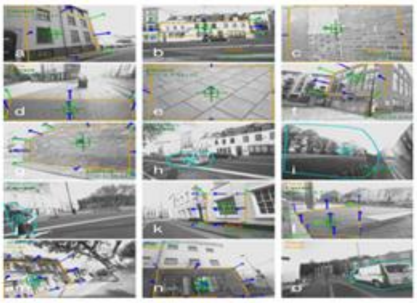
RESULTS:

This section presents results of experiments to evaluate the proposed algorithms. First, we look at performance of the plane recognition algorithm on individual image regions; before showing the results of experiments to evaluate the full plane detection method, both against our own ground truth data, and by comparing with prior work.

Plane Recognition:

The data we used for evaluating the plane recognition algorithm consist of regions extracted manually from images (we are not using the whole image), labeled with the true class (plane or non-plane) and orientation (normal vector), as described above. We used two datasets, for training and testing.

The training set was used for cross validation, to evaluate the accuracy and consistency of the method, and to investigate performance using different representations and parameter values, before training the full algorithm. This consisted of 556 regions captured by a 320 _ 240 pixel calibrated webcam.



Fg: a-j

Independent Data:

Example outputs of plane recognition, showing correct classification (a-j) and good orientation estimation (a-g), plus some failure cases: poor orientation estimate (k,l), misclassification as non-plane (m,n), and misclassification as planar (o). Orange/cyan boundaries denote ground-truth plane/non-plane respectively; those classified as planes have green arrows (estimated orientation), ground-truth orientation is drawn with blue arrows. Fig:(a-j) shows examples of successful plane recognition, from the independent data. Correctly classified planes and their orientation are indicated by green arrows (ground truth is shown in blue) while correctly identified non-planar regions are indicated by cyan circles, including vehicles, foliage and people. Note in particular the variation in appearance of the planar regions, including both regular and irregular texture, demonstrating the generality of the algorithm.

Multi-resolution Grid:

To compare this with the original version, we evaluated this new algorithm on the independent dataset. It gave a classification accuracy of 81.6 percent, and a median orientation error of 16.3 degrees (histogram of errors in Fig. 9b). These results are a little worse than the DoG method, but we believe this constitutes good performance—especially since this means the detection is now covering almost the entire image (94 percent by area).

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

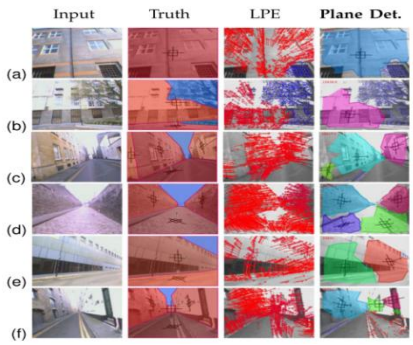


Fig: shows a selection of results, alongside the inter-mediate local plane estimates (Section 5.1) and ground truth. Example 10c shows that it is quite capable of detecting planes in environments where there are dominant vanishing lines; but example 10d is important since it shows it can also cope in the absence of any obvious geometric structure, where such methods would fail. Note also Fig. 10b (and Fig. 1), where non-planar areas are successfully segmented from the planar surfaces.

Smoothing:

We discussed how a MRF can be employed to smooth the assignment of planarity and orientation labels to the points, before extracting regions. The parameters a_P ;a_O control the relative influence of the unary and pair

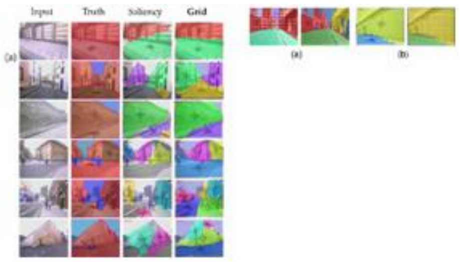


Fig: Examples of using the grid-based whole-image method, com-pared to ground truth and the original saliency-based version.

CONCLUSIONS:

We have shown that it is possible to learn the relationship between appearance and structure in single images, and presented a new algorithm to detect planes, which can for the first time estimate a 3D orientation. The approach comprises a method to estimate the planarity and orientation of individual regions, based on learning from a training set. This is then used in a plane detection algorithm that does not require a priori region segmentation or knowledge of plane boundaries. Our algorithm can detect planes with good accuracy compared to labeled ground truth, and gives comparable segmentations to the most similar work [19].

The plane detection works by repeated sampling of windows to recover individual planes; however, this makes it unable to deal with small planar regions. An avenue of future work, therefore, would be to incorporate edge or con-tour information, which can be beneficial in scene layout estimation [20]. A similar technique could also be applied to relative depth estimation [32]—to improve the fidelity of plane detection, or to use alongside plane detection for more sophisticated interpretation of images.

REFERENCES:

[1] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, and A. Konushin, “Fast automatic single-view 3-d reconstruction of urban scenes,” in Proc. Eur. Conf. Comput. Vis., 2008, pp. 100–113.

[2] J. Besag, “On the statistical analysis of dirty pictures,” J. Roy. Stat. Soc. B, vol. 48, no. 3, pp. 259–302, 1986.

[3] S. Birchfield and S. Rangarajan, “Spatiograms versus histograms for region-based tracking,” in Proc. IEEE Comput. Soc. Conf. Com-put. Vis. Pattern Recognit., 2005, pp. 1158–1163.

[4] D. Blei, A. Ng, and M. Jordan, “Latent Dirichlet allocation,” J. Mach. Learn. Res., vol. 3, pp. 993–1022, 2003.

[5] S. Choi, “Algorithms for orthogonal nonnegative matrix factorization,” in Proc. IEEE Int. Joint Conf. Neural Netw., 2008, pp. 1828–1832.

[6] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” IEEE

Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 603–619, May. 2002.

[7] A. Criminisi, I. Reid, and A. Zisserman, “Single view metrology,” Int. J. Comput. Vis., vol. 40, no. 2, pp. 123–148, 2000.

[8] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” J. Amer. Soc. Inform. Sci., vol. 41, no. 6, pp. 391–407, 1990.

[9] R. Fergus, P. Perona, and A. Zisserman, “A sparse object category model for efficient learning and exhaustive recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2005, pp. 380–387.

[10]J. Garding, “Direct estimation of shape from texture,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 15, no. 11, pp. 1202–1208, Nov. 1993.

[11]R. L. Gregory, “Knowledge in perception and illusion,” Philos. Trans. Roy. Soc. London. Series B: Biol. Sci., vol. 352, no. 1358, pp. 1121–1127, 1997.

[12]F. Guo and R. Chellappa, “Video metrology using a single cam-era,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 7, pp. 1329–1335, Jul. 2010.

[13]O. Haines, “Plane detection from single images,” Ph.D. disserta-tion, Univ. Bristol, Bristol, U.K., 2013.

[14]O. Haines and A. Calway, “Detecting planes and estimating their orientation from a single image,” in Proc. Brit. Mach. Vis. Conf., 2012, pp. 31.1–31.11.

[15]O. Haines and A. Calway, “Estimating planar structure in single images by learning from examples,” in Proc. Int. Conf. Pattern Rec-ognit. Appl. Methods, 2012, vol. 2, pp. 289–294.

[16]O. Haines, J. Mart_ınez-Carranza, and A. Calway, “Visual mapping using learned structural priors,” in Proc. IEEE Int. Conf. Robot. Autom., 2013, pp. 2227–2232.

[17]R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[18]D. Hoiem, A. Efros, and M. Hebert, “Putting objects in perspective,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2006, pp. 2137–2144.

[19]D. Hoiem, A. Efros, and M. Hebert, “Recovering surface layout from an image,” Int. J. Comput. Vis., vol. 75, no. 1, pp. 151–172, 2007.

[20]D. Hoiem, A. Stein, A. Efros, and M. Hebert, “Recovering occlu-sion boundaries from a single image,” in Proc. Int. Conf. Comput. Vis., 2007, pp. 1–8.

[21]P. Kohli and P. Torr, “Dynamic graph cuts for efficient inference in Markov random fields,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 12, pp. 2079–2088, Dec. 2007.

[22]J. Ko_secka_ and W. Zhang, “Extraction, matching, and pose recov-ery based on dominant rectangular structures,” Comput. Vis. Image Understanding, vol. 100, no. 3, pp. 274–293, 2005.

[23]S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene catego-ries,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2006, vol. 2, pp. 2169–2178.

[24]D. G. Lowe, “Distinctive image features from scale-invariant key-points,” Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004.

[25]C. D. Manning, H. Raghavan, and P. Schtze, Introduction to Infor-mation Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[26]J. Mart_ınez-Carranza and A. Calway, “Efficient visual odometry using a structure-driven temporal map,” in Proc. IEEE Int. Conf. Robot. Autom., 2012, pp. 5210–5215.

[27]B. Mi_cu_s_ık, H. Wildenauer, and J. Ko_secka,_ “Detection and match-ing of rectilinear structures,” in Proc. IEEE Conf. Comput. Vis. Pat- tern Recognit., 2008, pp. 1–7.

[28]C. O Conaire, N. O’Connor, and A. Smeaton, “An improved spatio-gram similarity measure for robust object localisation,” in Proc. IEEE Int. Conf. Acoustics, Speech Signal Process., 2007,

pp. 1-1069–1-1072.

[29]G. Orban,_ J. Fiser, R. Aslin, and M. Lengyel, “Bayesian model learning in human visual perception,” in Proc. Adv. Neural Inform. Process. Syst., 2006, pp. 1043–1050.

[30]E. Prados and O. Faugeras, “Shape from shading,” in Handbook of Mathematical Models in Computer Vision. New York, NY, USA: Springer, 2006, pp. 375–388.

[31]S. Ramalingam and M. Brand, “Lifting 3d manhattan lines from a single image,” in Proc. IEEE Int. Conf. Comput. Vis., 2013, pp. 497–504.

[32]A. Saxena, M. Sun, and A. Ng, “Make3D: learning 3D scene struc-ture from a single still image,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 5, pp. 824–840, May. 2009.

[33]A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and R. Cipolla, “Multivariate relevance vector machines for tracking,” in Proc. Eur. Conf. Comput. Vis., 2006, pp. 124–138.

[34]M. Tipping, “Sparse Bayesian learning and the relevance vector machine,” J. Mach. Learn.Res., vol. 1, pp. 211–244, 2001.

[35] A. Torralba and A. Oliva, “Depth estimation from image structure,”