Top Down Approach to Detect Multiple Planes from Pair of Images

Prateek Singhal
Robotics Research Center
IIIT Hyderabad, India

Aditya Deshpande
CVIT
IIIT Hyderabad, India

Harit Pandya
Robotics Research Center
IIIT Hyderabad, India

N Dinesh Reddy
Robotics Research Center
IIIT Hyderabad, India

K Madhava Krishna
Robotics Research Center
IIIT Hyderabad, India

ABSTRACT

Detecting multiple planes in images is a challenging problem, but one with many applications. Recent work such as J-Linkage and Ordered Residual Kernels have focussed on developing a domain independent approach to detect multiple structures. These multiple structure detection methods are then used for estimating multiple homographies given feature matches between two images. Features participating in the multiple homographies detected, provide us the multiple scene planes. We show that these methods provide locally optimal results and fail to merge detected planar patches to the true scene planes. These methods use only residues obtained on applying homography of one plane to another as cue for merging. In this paper, we develop additional cues such as local consistency of planes, local normals, texture etc. to perform better classification and merging. We formulate the classification as an MRF problem and use TRWS message passing algorithm to solve non metric energy terms and complex sparse graph structure. We show results on Michigan Indoor Corridor Dataset and our challenging dataset, common in robotics navigation scenarios. Experiments on the datasets demonstrate the accuracy of our plane detection relative to ground truth, with detailed comparisons to prior art.

Keywords

Detection, Markov Random Field, TRWS

1. INTRODUCTION

Detecting multiple planes in images is a challenging problem. If done accurately, it can provide strong cues to efficiently perform many vision tasks. Previously, Kähler and Denzler [6, 7], Zhou et al. [12] demonstrated the use of multiple planes for 3D reconstruction. Zhou et al. also exploit multiple planes for video stabilization [13]. Pham et al. develop an augmented reality application using multiple planar regions of images [9]. Kumar and Jawahar use multiple planes to guide camera positioning for robot manipulators [8]. To find planes, these methods use: (i) manual annotation [7] or, (ii) iterative RANSAC methods [7, 12] or, (iii) fit planes in the 3D reconstructed output (easier than detecting the planes in images) [13]. We leverage recently developed multiple structure detection methods and build a sophisticated approach to identify multiple planes given two images. Our approach gives good results on challenging datasets (few are shown in Figure 1), on which the methods discussed above either fail or work, but use extra information in the form of 3D reconstruction.

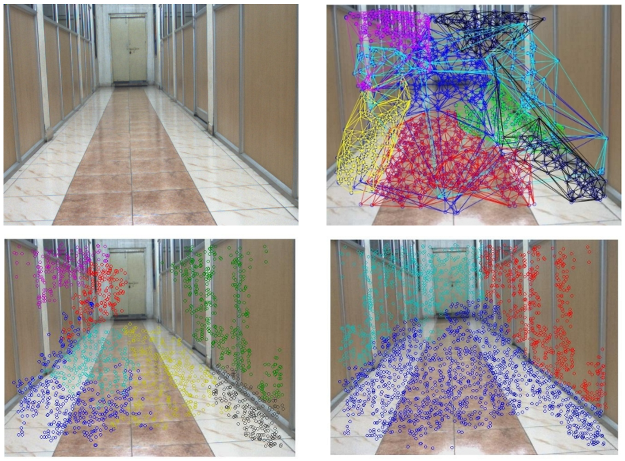


Figure 1: Multiple Plane detection from images. Top Left Initial Image to be segmented. Top Right Initial planar patches detected from ORK. Bottom Left Refined patches after distance based segmentation of planes Bottom Right Multiple planes detected corresponding to scene planes after optimal MRF labelling.

Though detecting multiple planar regions from only a single image is possible [4], it is an ill-posed problem. These methods make strong assumptions such as orthogonality of planes, or depend on tough to find features such as vanishing points, or require large labelled training datasets. As

ICVGIP '14, December 14-18, 2014, Bangalore, India
Copyright 2014 ACM ACM 978-1-4503-3061-9/14/12 ...$10.00.

a result of these limitations, single image plane detection methods cannot have wide applicability. In most of the applications discussed above viz. 3d reconstruction, augmented reality, video stabilization etc. more than one image of the planar scene is available. In this paper, we focus on the problem of detecting multiple planes given two images (i.e. an image pair). Given this setting we can compute stable features such as SIFT, SURF and also find feature matches between the two images. These matches can then be used to compute a *homography*, which encodes the transformation between planar region seen in the image pair. There are robust RANSAC based methods, for estimating a single homography, a survey of these methods can be found in [1]. Estimating a single homography typically gives us only the single most dominant planar region in the image. To find multiple planar regions, we need to estimate multiple homographies from the feature matches.

Initially, multiple homography estimation was performed using iterative methods. These methods eliminated the inliers of homography estimated for current iteration and again performed homography estimation on remaining matches. They typically need a priori knowledge of number of homographies/planes, otherwise we do not know when to terminate the iterative procedure. Also, the errors get compounded if a few wrong inliers are removed in the initial iterations and we end up achieving spurious results. More recently, sophisticated multi-structure detection methods have been developed by Toldo and Fusiello [11], Chin et al. [2], Pham et al. [9] and Jain and Govindu [5]. These methods bootstrap by randomly generating many hypotheses that fit a subset of data. The individual data points (in our case feature matches) are then associated with all the hypothesis (in our case homographies) that they fit well. The data points are then clustered into multiple structures based on the similarity of the set of hypothesis that they match to. The underlying idea is data points belonging to the same structure will show a preference to the same hypotheses from our initial sampled set. These multiple structure detection methods are domain independent and they show remarkable results when applied to problems such as multiple 2d-line fitting, multiple 2d-circle fitting, multiple 3d-plane fitting etc. When we use these methods for our problem of *multiple homography estimation*, we see that these methods at best provide us with multiple small planar patches (after some post processing), far exceeding the number of planes in our image pair. We use realistic scenes for all our experiments. This difference in performance is a result of sampling homographies that fit nearby points and also, there being more ambiguity in the problem of homography estimation as compared to curve fitting.

After detecting multiple structures, typically some heuristic merging methods are used. These methods merge two homographies provided the residual error after merging is small. These methods do not work well in practice, this is demonstrated by the experiments done in Section 4.4. In our work, we develop a novel alternative to merge the planar patches output by multiple structure detection methods. We use homography decomposition to associate estimated homographies of planar patches to their normals in 3d world. At this stage, some planar patches have incorrect normals and cannot be trivially merged. We propose an MRF model using TRWS to achieve the merging. Each feature match is assigned a random variable, which can take labels corresponding to initial patches. We arrive at an optimal labelling of the feature matches by minimizing an energy function defined over these random variables. Using texture and locally computed normals in our energy minimization function, we show that we are able to assign correct labels to feature matches that span large scene planes. This is primarily because we incorporate local normals in our MRF formulation, majority of which are correctly oriented despite the normal of the entire patch being incorrect. Also, our smoothness term in the MRF formulation ensures that labels assigned to a feature match are consistent with its neighboring matches. As discussed in section 5, we show good results on challenging datasets and compare our performance to other state-of-the-art multiple plane detection methods.

2. RELATED WORK

RANSAC based homography estimation methods have been extended to the problem of detecting multiple planes by removing inliers and re-estimating new homographies iteratively. Further, Zuliani et al. [14] developed a multi-RANSAC algorithm that is capable of estimating all homographies simultaneously. These methods do not work well in practice and also, need additional knowledge of number of planes.

2.1 Methods using J-Linkage

Recently, sophisticated algorithms which do not require prior knowledge of number of planes have also been developed. Toldo and Fusiello [11] develop one such algorithm called *J-Linkage*. For homography estimation, J-Linkage starts by generating M homographies from randomly sampled nearby feature matches. For each feature match, a preference set of the homographies (out of the M) that fit to the match within a threshold ϵ is created. A clustering step iteratively merges the feature matches that have similar preference set using the Jaccard Distance measure ($d_J(X,Y) = \frac{|X \cup Y| - |X \cap Y|}{|X \cup Y}$). This clustering step proceeds till the minimum d_J is 1, i.e. the preference set of all clusters have no more overlap.

Fouhey et al. [3] observe that J-linkage uses only nearby feature matches to generate initial homographies. The downside of such an approach is that the homographies output by J-linkage are also locally optimal and do not fit a large scene plane. Fouhey et al. [3] solve this by continuing to cluster the matches (after J-linkage) using the distance measure:

$$d_F(X,Y) = \frac{1}{|X \cup Y|} \sum_{c \epsilon X,Y} err_{H'}(c)$$

2.2 Methods using Ordered Residual Kernel

Another category of multiple structure detection algorithms based on ordered residues have been formulated by Chin et al. [2]. For the problem of multiple homography estimation, again these methods start by randomly sampling M homographies. Residues $(x' - Hx)$ are then computed for each feature match and homographies are ordered on the basis of increasing residue for each feature match.

Thus, given a data point θ_i we obtain ordered homographies λ_1^i (minimum residue) to λ_M^i (maximum residue):

$$\tilde{\theta}_i = \{\lambda_1^i, \lambda_2^i, \lambda_3^i, ..., \lambda_{M-1}^i, \lambda_M^i\}$$

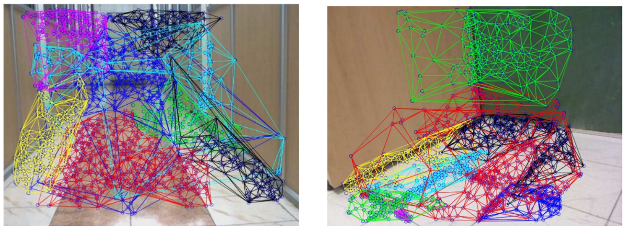


Figure 2: Scene planes detected for corridor, corner dataset respectively (from left). Connected meshes with the same color indicate matches that are grouped together as belonging to one plane by [2].

An Ordered Residual Kernel (ORK), k_r, is defined between two data points:

$$k_r(\theta_{i1},\theta_{i2}) = \tfrac{1}{Z}\sum_{t=1}^{M/h} z_t k_\cap^t(\tilde{\theta_{i1}},\tilde{\theta_{i2}})$$

$$k_\cap^t = \tfrac{1}{h}(|\tilde{\theta_{i1}}^{1:\alpha_t} \cap \tilde{\theta_{i2}}^{1:\alpha_t}| - |\tilde{\theta_{i1}}^{1:\alpha_{t-1}} \cap \tilde{\theta_{i2}}^{1:\alpha_{t-1}}|)$$

$k_\cap^t$ is the Difference of Intersection Kernel (DOIK) defined on the homographies ordered by residues ($\tilde{\theta_{i1}}$ and $\tilde{\theta_{i2}}$) of the two data points.

The ORK is a weighted sum of the difference in number of intersecting homographies taken over some step size h. This kernel is a valid mercer kernel and induces a mapping of the data points to a Reproducing Kernel Hilbert Space (RKHS). Chin et al. show that data points belonging to the same structure form clusters in this RKHS. They use kernel PCA and spectral clustering to detect these structures. To minimize the number of structures detected after this step, they also give a structure merging scheme. This merging scheme sequentially merges structures if the overall residue after merging is below a threshold. The merging continues till all the data can be explained satisfactorily by identified structures (i.e. sum of all residues is bounded).

2.3 Single Image Methods

Recently lot of work has been done in extracting spatial layout using single image. Hedau et al. [4] have used parallel lines to estimate the Vanishing Points, thus estimating the orthogonal planes along the dominant directions. The scene is divided into super pixels which are then classified into the dominant directions based on the assumption of box like configurations. Box like configurations are learnt and then fitted to choose the correct hypothesis for the surface. This in principle places a constraint on the number of planes, also these methods essentially use machine learning to learn the descriptors making them unfeasible in real time.

3. CONTRIBUTIONS

In this work, we build a system to detect multiple planes from a pair of images using a monocular camera having 6DOF motion. We build on the existing state of the art Multiple Plane Detection system in the following way:

- Use of local constraints.
- Computing and using 3D properties from a pair of images.
- Removing Manhattan constraints.
- Not requiring initial motion estimate.

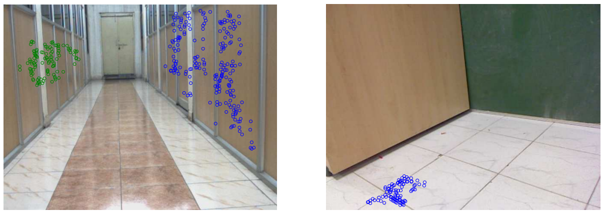


Figure 3: Planar patches obtained after the distance based refinement step for corridor and corner dataset. In comparison to Fig 2, the images show that our distance based refinement step is able to cut the detected scene planes (that spanned two or more real world planes) into smaller planar patches (that span only a single scene plane).

4. OUR APPROACH

The following sections describe the different steps of our approach.

4.1 Initial Planar Patch Estimates

Given two images we compute SURF features and obtain feature matches. On obtaining these matches, we use the multiple structure detection method described in section 2.2 for estimating multiple homographies and hence, multiple scene planes. We perform the steps of: (i) computing ordered residues of feature matches, (ii) using the ordered residual kernel, (iii) performing kernel PCA and spectral clustering. We avoid using the cost of residues based structure merging scheme given by Chin et al. [2]. We justify in Section 4.4 that such schemes do not work well in practice. Refer [2] for more details of this method. Figure 2 shows the scene planes that are found by using this method on two datasets: (i) *corridor*, consisting of 4 planes and (ii) *corner*, consisting 3 planes representative of a corner of a cuboid. As seen in Figure 2, the detected planes have two major problems:

Firstly not all detected planes correspond to a planar patch. For example, the blue mesh in *corridor* dataset shows that the detected plane spans left as well as right plane, the 2 red meshes in *corner* dataset show that 2 detected planes span ground and left, right and left planes respectively. Also,the number of planes detected by the multiple structure detection method exceeds the number of true scene planes.

We solve these problems using the following pipeline:

- Distance based refinement of detected planes.
- Computation of plane normals.
- MRF based optimal labelling due to local normal and texture constraints.

4.2 Distance Based Refinement of Plane Hypotheses

In this step, we first perform a delaunay triangulation on the feature matches. If the detected plane has feature matches from multiple scene planes, chances are these would be at a larger distance from each other. We set a threshold on distance and cut the delaunay mesh into smaller meshes (when any side exceeds the distance threshold). This step, to some extent, separates the planes output by multiple structure detection method into smaller planar patches. These smaller planar patches have stable properties as compared to detected planes which spanned multiple true scene planes. The results of performing this step on *corridor, corner* dataset are shown in Figure 3.

4.3 Plane Normal Computation

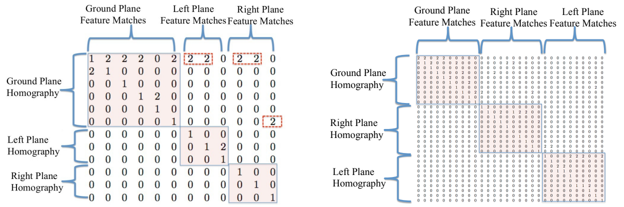
Planar points between two images are related by homography. Homography is a relation between the plane and the relative pose between the images which can be decomposed [12] to find the plane normals.

$$H = (R + \frac{TN^t}{D})$$

where N is the plane normal and D is the perpendicular distance between the plane and camera. We have found that in cases of perspective motion between the camera and the plane, in presence of multiple planes, the decomposition is mostly erroneous due to the bilinear nature of the normal and translation term. We also discard planes formed by less than 10 points as they are mostly erroneous.

4.4 Residue Based Merging of Plane Hypotheses

Typically methods of detecting multiple homographies, including Chin et al. resort to merging the detected planes by applying the homography of one to other. The merging is done, if the residue after applying the homography of another plane is below a threshold. We design an experiment, where we perform the above on: (i) planar patches out put by Chin et al. before doing the merging and (ii) on planar patches that are manually marked, these planes have desirable properties (viz. they span entire planar region). As shown in Figure 4, for (i) we have 6 ground, 3 left and 3 right planes. For (ii), we have use 10 ground, 10 left and 10 right planes. We create a matrix where the rows indicate the homography taken and columns indicate the plane to which it is applied. We mark out the first and second minimum residues by 1 and 2 in this matrix. The second minimum residue will dictate the merging in residues based merging approach. In ideal conditions, all the 1's and 2's should lie in the shaded regions of Figure 4. This is the case for controlled experiment (ii), but not for (i) (5 out of 12 planes marked in red have second minimum residues for incorrect homographies). Since our experiments show that using residues alone is not sufficient, especially for multiple structure detection methods like Chin et al., we develop other cues that can be exploited to achieve merging.



	Ground Plane Feature Matches						Left Plane Feature Matches			Right Plane Feature Matches		
Ground Plane Homography	1	2	2	2	0	2	2	2	0	2	2	0
	2	1	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	0	1	2	0	0	0	0	0	0	0
	0	0	0	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	0	0	2
Left Plane Homography	0	0	0	0	0	0	1	0	0	0	0	0
	0	0	0	0	0	0	0	1	2	0	0	0
	0	0	0	0	0	0	0	0	1	0	0	0
Right Plane Homography	0	0	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	0	1

	Ground Plane Feature Matches										Right Plane Feature Matches										Left Plane Feature Matches									
Ground Plane Homography	2	2	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	2	0	0	0	2	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
	0	0	0	0	0	0	0	1	0	2	2	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Right Plane Homography	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	2	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Left Plane Homography	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	2	2	0	0	2
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 4: The minimum and second minimum after applying the homography of one plane to feature matches of the other.

4.5 Top Down Approach

Based on the discussion above, we propose exploiting the local consistency constraint (viz. same direction of normals, similar texture etc.), that should hold for a planar patch, to merge scene planes. We consider each feature match in its local neighbourhood (k nearest neighbours in detected planar patch) to form a small local planar patch. This is in cognizance to local patches embodying a scene plane [3]. With our decomposition to local planar patches, we can now compute local surface normal for each such planar patch. The consistency in orientation of local normals spanning a single scene plane is an important cue, so is its texture. We use these cues to refine and recompute association of feature matches to the detected planes. Note that we do an MRF optimization on a sparse graph of only feature matches and not a dense graph of all pixels. Such dense graphs are common in image segmentation literature. We call our approach a *top-down approach*, because we resort to feature matches and local planar patches to merge detected planes, after having performed one step of planar patch detection (using multi-structure detection methods). For our approach it is necessary to perform an initial multiple structure detection step. Because we assume that the true structure is present in the output of multiple structure detection method and develop a method to merge the detected structures to these true structures.

4.6 Graph Optimization

The feature matches are connected to form a graph using delaunay triangulation. This graph structure is formulated as Markov Random Field where the goal is to assign each feature match a maximal posterior probability (MAP) label. This label is one out of the detected planes by multiple structure detection step. As a common practice [16], instead of direct probability maximization, we minimize the energy as discrete labelling problem on the graph in the form of Eqn1.

$$E_{MRF} = \sum_{X} E(p, l \epsilon L) + \sum_{x} \sum_{q \epsilon N(X)} E(p,q) \qquad (1)$$

In the above equation, L = $(p^1..p^n)$ is the set of labels where n is the number of initial planar patches obtained from initial planar patch estimate step. The set $N(X)$ is the neigbourhood of the node X. The $E(p,l)$ defines the unary energy potential. It determines the likelihood of the feature match corresponding to a scene plane labelled l output by multiple structure detection step. $E(p,q)$ defines the pairwise energy potential which represents the graph similarity of the neighbourhood.

4.7 Unary Energy Term

Unary energy is defined by us as a sum of energies relating to *residues of local planar patch* and *normal similarity between feature and plane.*

$$E_{unary} = E(X)_{normal} + E(X)_{residual} \qquad (2)$$

4.7.1 *Residue of local planar patch*

Each feature match has a local planar patch defined around it using k nearest neighbours. Our energy is the sum of residues of this local patch with respect to the parameters of the patch labelled l.

$$E(X,l)_{residual} = \sum_{X \epsilon P} ||(X' - H_l X)||_{L2} \tag{3}$$

Here P represents the local planar patch. We use $k = 10$ as smaller patches can be erroneous while decomposition.These nearest neighbours are members of the same planar patch found initially as this implies that they are part of a larger plane rather than some local surface fit to a scene.

4.7.2 *Normal Similarity measure*

The local normal of each feature match should ideally be aligned with the normal of the scene plane. So the energy term for each feature decreases if it aligns with the plane labelled l.

$$E(X,l)_{normal} = (1 - \frac{N_X.N_l}{|N_X||N_l|})^2 \tag{4}$$

4.8 Pairwise Energy Terms

Pairwise energy defined by us consists of three terms *similarity measure, mutual plane consistency* and *texture similarity.*

$$E_{binary} = \lambda_1 E(X,Y)_{sm} + \lambda_2 E(X,Y)_{mp} + \lambda_3 E(X,Y)_{ts} \tag{5}$$

where λ_1 ... λ_n are the weights of pairwise terms.

4.8.1 *Similarity Measure*

We use the standard Potts Model where the neighbouring edges with different labels are penalized. Initially each feature is assigned the label of the initial planar patch to it belongs.

$$E(X,Y)_{sm} = \begin{cases} 1 & \text{if } p_X \neq p_Y \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

4.8.2 *Mutual Plane Consistency*

Neighbouring features should have similar surface normals, utilizing this constraint we find the measure.

$$E(X,Y)_{mp} = (1 - \frac{N_{X_p}.N_{X_q}}{|N_{X_p}||N_{X_q}|})^2 \tag{7}$$

4.8.3 *Texture Similarity*

This measure takes into account the local texture between neighbours should be similar. Here we compare the mean of a image patch around each feature match with its neighbour. We use a 5 × 5 patch centred at the feature. This term brings in the smoothness of texture across a plane, typically common in images.

$$\mu = \frac{\Sigma_{WS}(R,G,B)}{WS} \qquad : WS = WindowSize \tag{8}$$

$$E(X,Y)_{ts} = (\mu_{p(x)} - \mu_{p(y)})||_{L2} \tag{9}$$

This combination of energy terms segments out the planes robustly. We choose Tree Weighted Sequential(TRWS)[15] message passing algorithm to solve the optimization problem.

This method is similar to Loopy Belief Propagation and solves on the principles of linear programming and its duality for NP hard MRFs. The method has experimentally shown better results than LBP while working well in cases of non metric pairwise terms. It also finds the lower bound of the energy which acts as a guidance for convergence.Thus this method provides flexibility to include varied and sophisticated energy terms with complex graph structures. [10]

4.9 Learning

Parameter learning for MRF is itself an open problem and various approaches have been adopted. For instance Dahua et al. [15] employ a primal-dual message passing approach. This is an online algorithm which was shown to work with videos but since we work only on a pair of images it does not suit our settings. The pairwise energy terms in the MRF framework are a linear combination of different cues representing label smoothness, mutual plane consistency and texture similarity. Each metric has a different dimension and scale factor, hence for better performance, the weights (MRF parameters) must be carefully calibrated. Since each individual metric is very sensitive to input images, instead of constant weights, we propose a learning framework. For each metric, the feature vector comprises of top 3 DCT coefficients followed by top 3 Eigenvalues. We assume these metric to be independent of each other, and train regression based SVM for them separately using RBF kernel. The proposed feature vector is chosen due to the energy compaction property of DCT and variance consolidation property of Eigenvalues. The weights are learned offline based on previous experiences i.e. supervised, which makes the approach different from other online parameter learning approaches. The model is tested using leave-one-out cross-validation as the data is limited but varies across the dataset.

5. EXPERIMENTAL RESULTS

We evaluate our approach on various images taken in an indoor environment. The images of our datasets and the results of different methods for multiple plane detection on them have been shown in Figure 5. Our images are representative of scenes that will be encountered by a robot in SLAM setting and also in other 3d reconstruction methods. These datasets are challenging because some of our planes have multiple textures (ground plane of corridor) and a specular reflection (glass in the left and right planes of corridor, ground plane of box). The images have been taken from a dataset captured for VSLAM by a Flea2 camera mounted on P3Dx robot and hand held cameras. There is considerable movement between the images (≈ around $20cm$ for corridor dataset and for others it is in the range of 5 to 10cm, with 5 to 10 degrees rotation). We compute SURF features at the low threshold for the Hessian. Since we have a distinct motion we are able to use the KLT tracker to find correct feature matches. These features generally encompass the image and the corresponding planes well. We compare our approach to the three approaches – Fouhey et al. [3], Chin et al. [2] and Hedau et al.[4] – discussed in Section 2. For Fouhey et al., Chin et al. and Hedau et al. we use their publicly available code. For fairness we run the codes several times and the best results were taken.

Using 1500 SURF features tracked by KLT, code by Fouhey

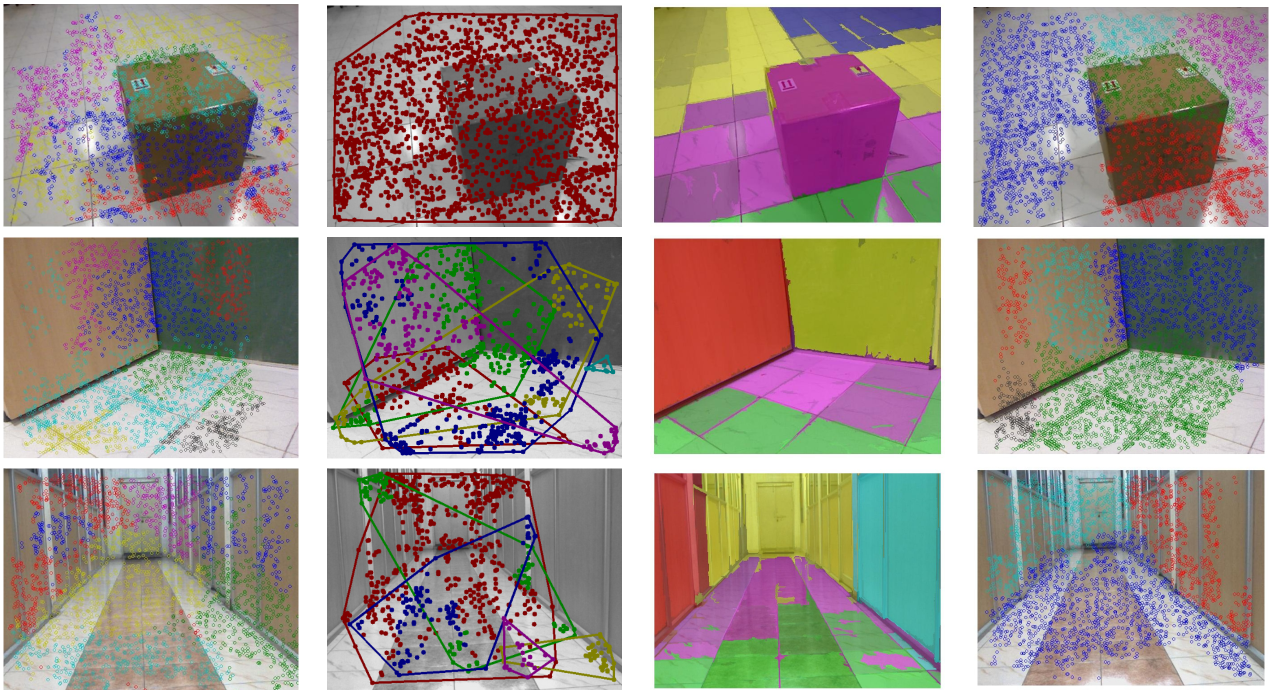


Figure 5: This figure compares the results of different multiple plane detection methods on different datasets. Datasets - *Box, Corner and Corridor (top to bottom). From (from left to right), each column corresponds to the following methods - Chin et al. [2], Fouhey et al.[3], Hedau et al.[4] and our approach.*

et al. took in the range of 5 to 10 mins per experiment. As can be seen from Figure 5, for corridor it finds erroneous planes. For box dataset, where there are parallel planes (with different textures) it labels both of them as same. Also, it fails to detect the front facing plane.It performs badly on such image's as the data has high amount of outliers and purely residue based merging does not help. For the corridor and lab dataset it finds multiple planes spanning other planes as well as the number of planes is grossly oversegmented .

We tested our images using the pre-trained models for Hedau et al.Ẇe cant train the models as we show our method only on 2 images and training for each dataset defeats this purpose. The method, though ill- posed for detection of planes, is a benchmark in indoor environments using vanishing points for scene understanding. The method can extract only 3 orthogonal dominant directions (walls , ceiling and floor) using vanishing points. For the box dataset it completely fails as it is trained for box like structure of the dominant directions while this dataset doesnt confirm to it. It finds the right box models for the the corner and corridor dataset but fails to segment them accurately.

Taking this performance into consideration we do a quantitative and qualitative analysis of results only between ORK and our approach. We run both approaches with Multi Guided Sampling Pham et al . [9]. Table 1 shows Classification error of the data and number of planes detected in the image. Our approach shows competitive results for classification error while having a lower error for most of the datasets. The number of planes detected by ORK is more erroneous than ours as it over segments the same plane.

Table 1: Classification Error

METHOD		OUR		ORK	
Dataset	No of Ground truth SPs	error(in '%')	No of SPs detected	error(in '%')	No of SPs detected
Corner	3	18.9	5	12.33	7
Box	3	13.24	5	13.53	6
Corridor	4	18.78	3	23.65	6

5.1 Qualitative Analysis

Chin et al. perform better than Fouhey et al. for our datasets, but there is still scope for improvement. For the box dataset, the topmost plane (detected in blue) has a few other planes detected in between. Similar is the case with ground plane in the corner dataset, it is split into multiple planar patches. There are also erroneous planes viz. the plane marked by green feature matches in the corridor dataset. This plane has matches from the left as well as ground plane. Similarly, plane marked by blue in corner dataset spans the left and ground plane. We are able to solve these problems in our approach, since we impose strong locality constraints and also look at the local normals, texture to perform merging. In the box dataset the floor consists of specular reflections and the texture varies along the floor leading to oversegmentation but we still clearly segment the top of the box from the floor. In the corridor dataset, the perspective plane is not segmented as the normal decomposition for perspective motion is a special case due to the bilinear nature of homography.

Table 2: Classification Error

	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7
Precision for ORK	48	40	68	41	49	47	76
Precision for Ours	63	77	69	74	74	86	69
Recall for ORK	88	51	80	46	67	65	78
Recall for Ours	95	68	81	69	75	92	70

We also test our approach on publically available Michigan Indoor Corridor dataset [16]. Michigan dataset comprises of several images of 11 different views. This is a challenging dataset for plane segmentation due to high variation in texture and Non-orthogonal planes, where it is difficult to detect vanishing points. We choose to compare 2 images with wide motion variation to show robustness to motion. We compare our plane segmentation results with ORK. As shown by Table 2, we outperform the approach suggested by ORK. Being a plane segmentation algorithm,instead of using classification accuracy, we use precision-recall based performance criterion to evaluate the performance. Precision-recall for an image is computed by weighted sum of individual precision-recall of individual planes. Our results as shown in Figure 6 perform better than ORK for both precision and recall. A reason sampling based methods like ORK degrade in performance is due to generalized sampling. We do not subsample our features in the dataset as we aim to utilize the whole scene. Supplementary results for sequences showing our invariance to motion can be found at [17].

6. CONCLUSION

In this work, we develop an MRF based top down approach to merge multiple small planar patches detected by multiple structure detection methods. We show significant improvement over previous methods. This improvement results from the fact that we bring in domain knowledge to the problem of multiple plane detection. Our domain knowledge is in the form of cues such as local normals, texture and local consistency of planes. We formulate an energy function using this domain knowledge and minimize it through an MRF optimization.

7. ACKNOWLEDGEMENT

This research was supported by the Board of Research in Nuclear Sciences (BNRS) grant.

8. REFERENCES

[1] A. Agarwal, C. V. Jawahar, and P. J. Narayanan. A survey of planar homography estimation techniques. Technical report, 2005.

[2] T.-J. Chin, H. Wang, and D. Suter. Robust fitting of multiple structures: The statistical learning approach. In *ICCV*, pages 413–420, 2009.

[3] D. F. Fouhey, D. Scharstein, and A. J. Briggs. Multiple plane detection in image pairs using j-linkage. In *Int. Conf. on Pattern Recognition*, 2010.

[4] V. Hedau, D. Hoiem, and D. A. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, pages 1849–1856, 2009.

[5] S. Jain and V. M. Govindu. Efficient higher order clustering on the grassmann manifold. In *ICCV*, 2013.

[6] O. Kähler and J. Denzler. Detecting coplanar feature points in handheld image sequences. In *In Proceedings Conference on Computer Vision Theory and Applications, VISAPP 2007*, pages 447–452. INSTICC Press, 2007.

[7] O. Kähler and J. Denzler. Tracking and reconstruction in a combined optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):387–401, 2012.

[8] D. S. Kumar and C. Jawahar. Robust homography-based control for camera positioning in piecewise planar environments. In *Computer Vision, Graphics and Image Processing*, pages 906–918. Springer Berlin Heidelberg, 2006.

[9] T.-T. Pham, T.-J. Chin, J. Yu, and D. Suter. The random cluster model for robust geometric fitting. In *CVPR*, pages 710–717. IEEE, 2012.

[10] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, A. Agarwala, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *In ECCV*, pages 16–29, 2006.

[11] R. Toldo and A. Fusiello. Robust multiple structures estimation with j-linkage. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 537–547, Berlin, Heidelberg, 2008. Springer-Verlag.

[12] Z. Zhou, H. Jin, and Y. Ma. Robust plane-based structure from motion. In *CVPR*, pages 1482–1489, 2012.

[13] Z. Zhou, H. Jin, and Y. Ma. Plane-based content preserving warps for video stabilization. In *CVPR*, pages 2299–2306, 2013.

[14] M. Zuliani, C. Kenney, and B. Manjunath. The multiransac algorithm and its application to detect planar homographies. In *ICIP 2005.*, volume 3, pages III–153–6, 2005.

[15] D. Lin, S. Fidler, and R. Urtasun. Holistic Scene Understanding for 3D Object Detection with RGBD Cameras, In *ICCV*, pages 1414-1424, 2013.

[16] G. Tsai, C. Xu, J. Liu, and B. Kuipers Real-time indoor scene understanding using Bayesian filtering with motion cues In *ICCV*, pages 121–128, 2011.

[17] www.http://www.cc.gatech.edu/ psinghal/ICVGIP2014

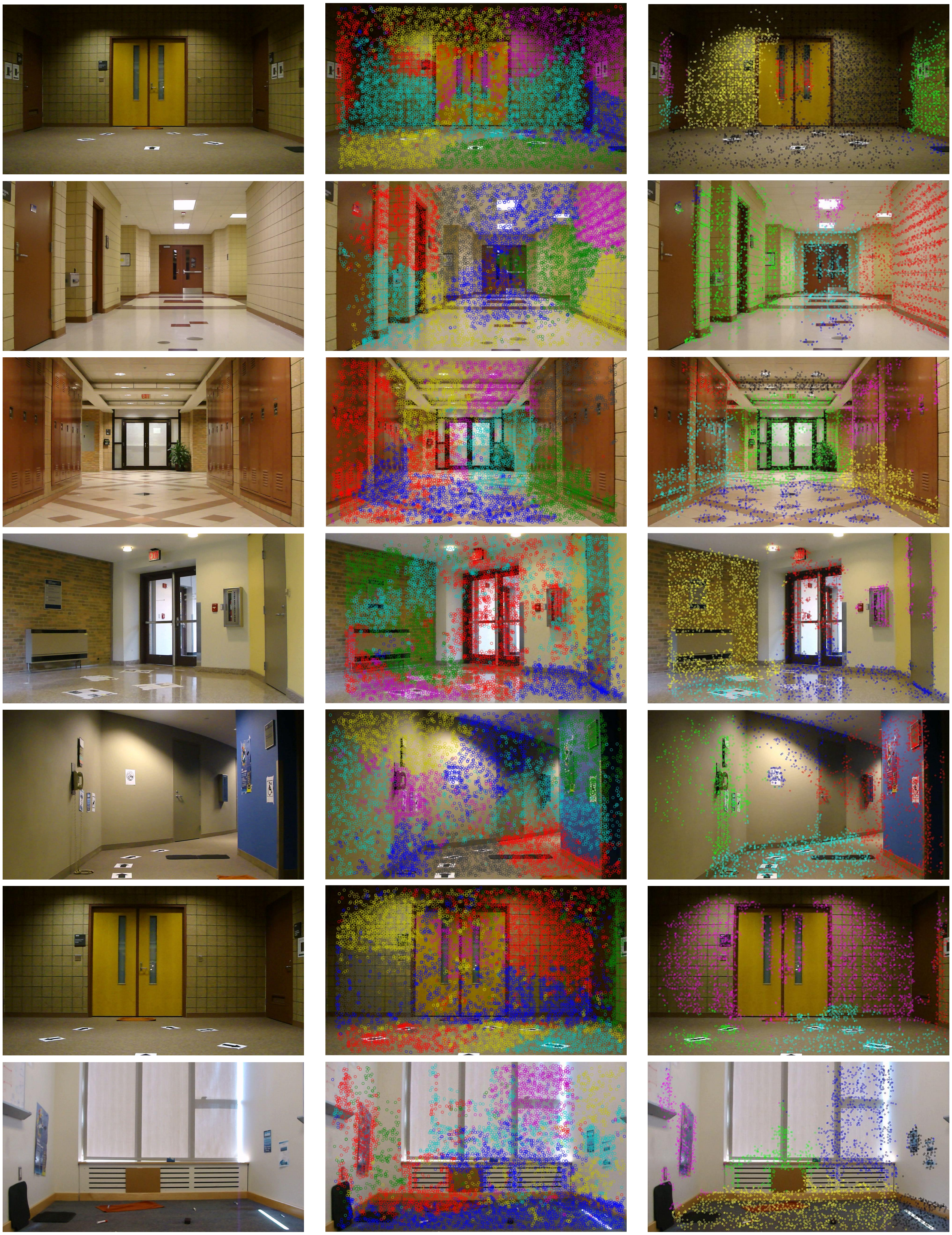


Figure 6: This figure compares the results of different multiple plane detection methods on the ***Michigan dataset. Datasets - Image 1-7. (from top to bottom). From top to bottom, each column corresponds to the following methods*** **- Chin et al. [2] and our approach.**