

# Assessing Map Quality using Conditional Random Fields

Manjari Chandran-Ramesh, Paul Newman

► **To cite this version:**

Manjari Chandran-Ramesh, Paul Newman. Assessing Map Quality using Conditional Random Fields. 6th International Conference on Field and Service Robotics - FSR 2007, Jul 2007, Chamonix, France. Springer, 42, 2007, Springer Tracts in Advanced Robotics. <inria-00201216>

**HAL Id: inria-00201216**

**<https://hal.inria.fr/inria-00201216>**

Submitted on 27 Dec 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Assessing Map Quality using Conditional Random Fields

Manjari Chandran-Ramesh and Paul Newman

Robotics Research Group, University of Oxford, Oxford - OX1 3PJ {manjari,pnewman}@robots.ox.ac.uk

**Summary.** This paper is concerned with assessing the quality of work-space maps. While there has been much work in recent years on building maps of field settings, little attention has been given to endowing a machine with introspective competencies which would allow assessing the reliability/plausibility of the representation. We classify regions in 3D point-cloud maps into two binary classes — “plausible” or “suspicious”. In this paper we concentrate on the classification of urban maps and use a Conditional Random Fields to model the intrinsic qualities of planar patches and crucially, their relationship to each other. A bipartite labelling of the map is acquired via application of the Graph Cut algorithm. We present results using data gathered by a mobile robot equipped with a 3D laser range sensor while operating in a typical urban setting.

## 1 Introduction and Background

Robust localisation, efficient navigation and exploration are essential competencies in mobile robotics. These capabilities crucially depend on an internal representation of an agent’s workspace. Good theoretical progress in understanding the fundamentals of the mapping problem (including its parent problem - SLAM) has benefitted in particular the field robotics domain where *a-priori* maps are hard to come by and external navigation infrastructure is expensive or inconvenient to install. It comes as no surprise then, that the mapping of unknown environments has been and continues to be the focus of intense research. The past two decades have seen approaches to detecting loop closure while mapping (see, for example, [9]) as well as attempts to mapping dynamic environments [10],[14].

While recent mapping techniques are reasonably successful at building consistent representations of less complex workspaces, local map inconsistencies and errors are still commonplace in more complex environments. This is particularly problematic outdoors, where large distances lead to increased odometry error and the presence of dynamic objects such as pedestrians, cars or even leaves moving in the wind can lead to spurious sensor readings. Despite this known inadequacy of mapping algorithm, hardly any effort has been expended to determine the quality of a map once it is built. Clearly, the ability of automatically detecting possibly spurious workspace representations is of value, both for diagnostic purposes as well as an additional source of information about the affected area of the workspace itself.

This work aims to address this lack of introspection by introducing a procedure to measure the quality of 3D laser maps of complex urban environments. The approach makes explicit use of the artificial nature of these work-spaces by focusing on geometric features which are commonly found in man-made environments. The intuition is that most parts of the world can be represented by planar structures where a correctly rendered point-cloud produces a well-defined, non-intersecting - almost ‘crisp’ - image of the environment. Thus, the approach first segments the given 3D point-cloud map into plane patches consisting of a subset of the original points. A context-sensitive classification framework is then employed to classify these plane patches according to the plausibility of their being rendered correctly. While numerous techniques are available in the area of classification, each with their merits and demerits [18], [7], capturing contextual information has been shown to work effectively in the framework of Conditional Random Fields [13], [15], [1], [17]. Hence this work uses conditional random fields for the purpose of classification.

This paper is structured as follows: Section 2 provides a brief description of Conditional Random Fields. Section 3 develops and describes the actual method of classification of the map and Section 4 describes the application of the approach to laser data from a large urban environment.

## 2 Conditional Random Fields

A Markov random field is an undirected graphical model which has a set of nodes each of which corresponds to a random variable or group of variables, as well as a set of links each of which connects a pair of nodes [4], [3], [8], [11]. A conditional random field (CRF) can be viewed as a Markov random field globally conditioned on the random variable representing the observation sequence [19].

Consider a labelling problem consisting of a network of  $N$  nodes that represent the random variable,  $X$ , over the data sequence. Let these nodes be labelled by the random variable  $Y$  and have  $K$  possible labels:  $y_i \in \{1, \dots, K\}$ .  $X$  and  $Y$  are jointly distributed. However, Lafferty *et al.* [13] developed a conditional model  $p(y|x)$  that represented directly the conditional distribution of labels given the data sequence and did not require explicit modelling of the marginal  $p(x)$ . This conditional model became more suitable for classification tasks with overlapping attributes as it made no assumptions about the dependency structure [15].

Formally, a CRF can be defined as follows. Consider a graph  $G = (V, E)$ , such that  $Y = (y_i)_{i \in V}$  or  $Y$  is indexed by the vertices in  $G$ . When conditioned on  $X$ , if  $Y$  obeys the Markov property with respect to  $G$ , then  $(X, Y)$  is a CRF. This can be written as  $p(y_v|x, y_w, w \neq v) = p(y_v|x, y_w, w \sim v)$ , where  $w \sim v$  implies that  $w$  and  $v$  are neighbors in  $G$ . This graphical structure can be used to factorize the distribution into positive potential functions that operate on a subset of nodes. To ensure that the potential functions fulfill the conditional independence criteria of undirected graphical models, the concept of fully connected sub-graph or clique is introduced. The potential functions operate on the set of nodes that form a clique and the conditional distribution is factorized into a product of clique potentials  $\phi_c(x_c, y_c)$  where  $c \in C$  is a clique in the set of cliques  $C$  and  $x_c$  and  $y_c$  are the nodes and its labels in that clique. This factorization is written as

$$p(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \phi_c(x_c, y_c) \quad (1)$$

where  $Z(x)$  is the partition function (normalization constant) given by  $Z(x) = \sum_y \prod_{c \in C} \phi_c(x_c, y_c)$ . The partition function ensures that the conditional distribution  $p(y|x)$  is correctly normalized.

The potential function represents the constraints on the configuration or the ‘‘compatibility’’ between the nodes in the clique. Typically, for ease of notation, pairwise CRFs are considered and hence the maximal clique is of size two. This implies that each clique has local or node potentials,  $\phi_i(x_i, y_i)$  for each node  $i$ , and pairwise or edge potentials,  $\phi_{ij}(x_{ij}, y_i, y_j)$  for an edge between nodes  $i$  and  $j$ . The conditional distribution can now be written as

$$p(y|x) = \frac{1}{Z(x)} \prod_{i=1}^N \phi_i(x_i, y_i) \prod_{ij \in E} \phi_{ij}(x_{ij}, y_i, y_j) \quad (2)$$

where  $Z(x)$  is the partition function given by  $\sum_y \prod_{i=1}^N \phi_i(x_i, y_i) \prod_{ij \in E} \phi_{ij}(x_{ij}, y_i, y_j)$ ,  $N$  is the number of nodes,  $E$  is the set of edges  $\{ij\} (i < j)$  in the graph,  $\phi_i(x_i, y_i)$  is the node or unary potential,  $\phi_{ij}(x_{ij}, y_i, y_j)$  is the edge or binary potential.

Using the Hammersley-Clifford fundamental theorem of random fields [13], the potential functions can now take the form

$$p_{\Theta}(y|x) = \frac{1}{Z(x)} \exp \left( \sum_{i \in N, k} \mu_k f_k(x_i, y_i) + \sum_{ij \in E, k} \lambda_k g_k(x_{ij}, y_i, y_j) \right) \quad (3)$$

where  $f_k$  and  $g_k$  are vectors of local and pairwise features respectively and  $\Theta = (\mu_1, \mu_2, \dots, \mu_k; \lambda_1, \lambda_2, \dots, \lambda_k)$  are the parameters or weights to be estimated from training data. Typically, the user designs the feature functions which are often real-valued or binary and the importance of each feature is represented in the weights. The above form of the potential functions guarantees non-negative values for the potentials.

There are two steps involved in using a CRF for the purpose of classification, namely, the parameter learning step and the inference step. The parameter learning step involves the estimation of the parameters  $\Theta$  from training data. All cliques are made to share the same parameters to reduce the amount of training data

required. Consider a training set  $\{x^{(t)}, y^{(t)}\}$  that are independently and identically distributed. The distribution in equation (3) over all the training data as a function of the parameter set  $\Theta$ , is the likelihood given by  $p(\{y^{(t)}\}|\{x^{(t)}\}, \Theta)$ . The techniques maximum likelihood estimation (MLE) or maximum *a priori* estimation (MAP) can be used to estimate the parameter values from the training data. Typically, the logarithm of the likelihood is used for estimation. The log-likelihood for a CRF is given by

$$L(\Theta) = \sum_t \left[ \log \frac{1}{Z(x^t, \Theta)} + \sum_k \mu_k f_k(x^t, y^t) + \sum_k \lambda_k g_k(x^t, y^t) \right] \quad (4)$$

The above function is concave and hence there is a guaranteed convergence to the global maximum/minimum [19]. Parameter estimation in CRFs is an actively researched field and there exist various techniques. In this work, the maximum pseudo-likelihood method has been used to train the parameter set  $\Theta$ . For more details about this method, please refer [15], [2].

Once parameters have been estimated in the learning step, they can be used to infer the labels of an unlabelled data set. This is the inference step and is done by maximizing the conditional distribution of the labels given the feature vectors and the parameter set. This can be written as

$$Y = \arg \max_y p_{\Theta}(y|x) \quad (5)$$

where  $Y$  is the array of labels for the nodes. Optimizations based on graph cuts [6], [12], [5], [16] are a popular method to do this kind of maximization as they are guaranteed to find the global maximum when binary labels are required.

### 3 Map Quality Assessment Using a CRF

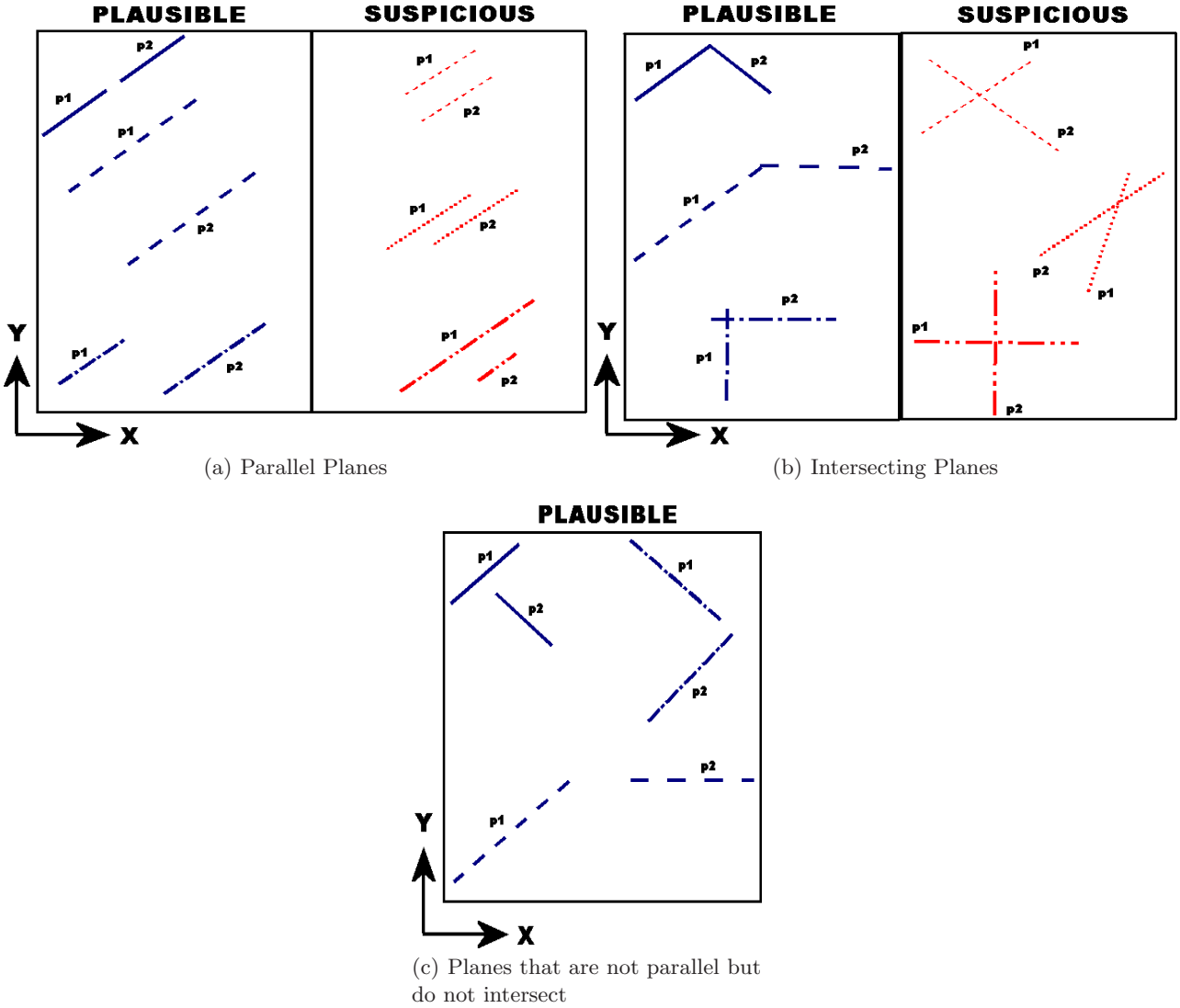
The previous section outlined the general CRF framework as an undirected graphical model comprising of nodes and edges. In this section, the use of a pairwise CRF framework for the purpose of binary classification of portions of a given point-cloud map is described. As stated earlier, the given point-cloud is segmented into plane patches consisting of a subset of points. These plane patches constitute the nodes and their alignment with neighboring plane patches constitute the edges. When selecting the neighborhood for each plane, care must be taken that enough contextual information is provided while the computation costs of inference and learning are not compromised. In this work, the neighborhood system is defined as the *four* closest plane patches to the current plane patch with respect to euclidean distance within a radius of  $s$  meters.

#### 3.1 Plausible and Suspicious Portions of a Map

The binary labels considered for this work are “plausible” and “suspicious”, that is, at the time of labelling the questions considered are – “is the subset of points acceptable as a plane patch?” and “is the alignment of the plane patch with respect to its neighbors reasonable?” Both these questions can be answered based on spatial geometry of the point-cloud. To check if a subset of points represent a plane patch, the metrics defined should measure how accurately the points fit a plane and whether the fitted plane is more two-dimensional than cubic in nature. To check if the alignment between neighboring plane patches is reasonable, the various cases of alignments first needs to be explored. These alignments can be broadly classified into plane patches that are parallel, those that are not parallel, but do not intersect and those that intersect. The alignments are shown in figure (1). Of these alignments, table (1) describes those that are considered acceptable and those that are not.

**Table 1.** Plausible and suspicious plane patch configurations.

Plausible	Suspicious
Parallel plane patches a reasonable distance apart.	Parallel plane patches separated only by a small distance.
Intersecting plane patches representing corners.	Intersecting plane patches that do not represent corners.
Plane patches that are not parallel but do not intersect.	



**Fig. 1.** The figure shows the different alignments of neighboring plane patches in a plan view. The different hashing styles denote the neighboring pairs. The Z-direction is into the paper. All alignments considered as “plausible” are in blue and those considered as “suspicious” are in red.

These somewhat intuitive considerations based on geometry can be formalized within the CRF framework by way of the feature functions.

### 3.2 Feature Functions

This section describes the features that reconstruct the intuitive considerations into functions that can be used in the CRF as node and edge potentials. The feature vectors,  $f_k, g_k$  along with the parameter set,  $\Theta$  estimated in the learning step, is used to label the plane patches (refer equations (3),(4),(5) in the previous section). The features for the vector  $f_k$  or node potential are defined by  $x_{i1}, \dots, x_{ik}$  and the features for the vector  $g_k$  or edge potential are defined as  $x_{ij1}, \dots, x_{ijk}$ .

#### Node Features

Intuitively, these features describe the representation of the subset of points as a plane. Three geometric properties are considered:

1. the mean-squared error of fit, to evaluate how accurately the points fit a plane

$$x_{i1} = \frac{1}{\frac{1}{M_i} \sum_{j=1}^{M_i} d_{j \rightarrow i}^2} \quad (6)$$

2. the cosine-distance between the plane normal and the z-axis, to evaluate the direction of the plane as this work considers only vertical planes

$$x_{i2} = \frac{1}{\vec{Z} \cdot \vec{n}_i} \quad (7)$$

3. and the aspect ratio of the plane patch as a measure of variation along the normal, to evaluate how two-dimensional the plane patch is

$$x_{i3} = \frac{V_i}{A_i} \quad (8)$$

In equations 6 - 8,  $x_{i1}$ ,  $x_{i2}$ ,  $x_{i3}$  denote the features of the  $i^{th}$  plane patch,  $M_i$  refers to the number of points in the  $i^{th}$  plane patch,  $d_{j \rightarrow i}$  is the distance of each point  $j$  in the  $i^{th}$  plane patch to plane patch fitted,  $Z$  is the  $Z$  axis vector,  $n_i$  is the normal to the  $i^{th}$  plane patch,  $V_i$  is the volume of the 3D convex hull fitted to the points in the  $i^{th}$  plane patch and  $A_i$  is the area of the 2D convex hull fitted to the points of the  $i^{th}$  plane patch, projected to the  $XY$  plane.

## Edge Features

The purpose of the edge features is to encode our intuitive understanding of how plausible or suspicious local plane configurations are (see Table 1).

Some measures that would give an idea of this alignment are the distance between the plane patches, the area of plane patches that overlap and if the plane patches intersect, the ratio of the distances between the patch edge and the line of intersection.

1. Plane separation: there are various methods of calculating the distance between adjacent plane patches. Here we consider only the shortest distance between points in both plane patches,

$$\rho_{ij} = \min_{\forall m \in M_i, \forall n \in M_j} (d_{mn}) \quad (9)$$

where  $\rho_{ij}$  denotes the closest distance between the plane patches  $i$  and  $j$ ,  $M_i$  and  $M_j$  represent the number of points in plane patches  $i$  and  $j$  respectively and  $d_{mn}$  is the distance between points  $m$  and  $n$ .

Based on this distance, the a *plausibility measure* can be weighted such that plane patches close to each other are weighted 1.

$$W_{ij} = \exp\left(\frac{-\rho_{ij}}{\tau}\right) \quad (10)$$

where  $W_{ij}$  is the weight,  $\rho_{ij}$  is the closest distance between the plane patches and  $\tau$  is the distance at which plausibility of alignment between the plane patches becomes true.

2. Area of overlap: The maximum of the areas of the  $i^{th}$  plane patch projected on the  $j^{th}$  plane patch and of the  $j^{th}$  plane patch projected on the  $i^{th}$  plane patch is taken. This is then normalized with the minimum of areas of the  $i^{th}$  and the  $j^{th}$  plane patches projected onto the  $XY$  plane. This is illustrated in figure (2(b)) and written as

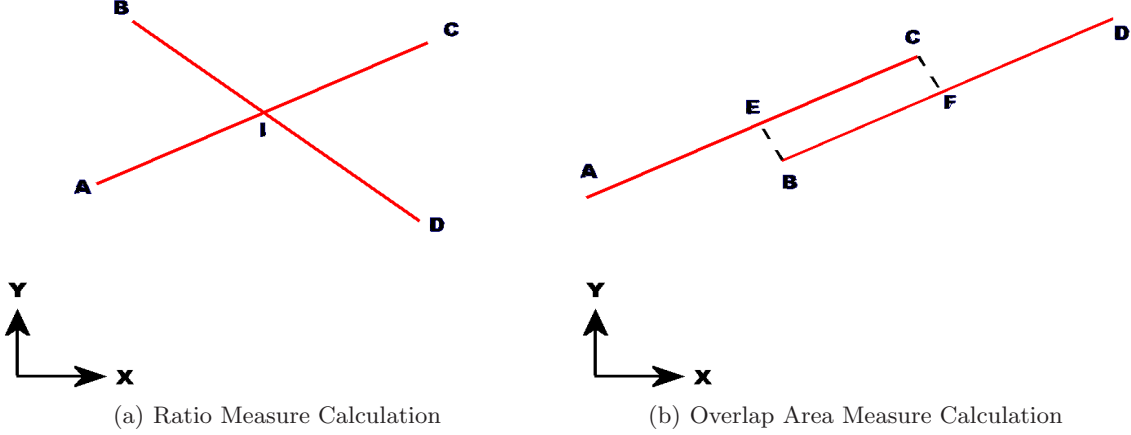
$$A_{ij} = \frac{\max(A_{i \rightarrow j}, A_{j \rightarrow i})}{\min(A_i, A_j)} \quad (11)$$

where  $A_{ij}$  is the overlapping area,  $A_{i \rightarrow j}$  is the area of  $i^{th}$  plane patch projected on the  $j^{th}$  plane patch,  $A_{j \rightarrow i}$  is the area of the  $j^{th}$  plane patch projected on the  $i^{th}$  plane patch,  $A_i$  and  $A_j$  are the areas of the plane patches  $i$  and  $j$  respectively.

3. Location of line of intersection with respect to plane boundaries: this property concerns intersecting plane patches only and is defined as 0 when the plane patches are parallel. In this measure, the line of intersection between the patches is found and the distance between this line and the edges of the patches are computed. The ratio between the minimum distance and maximum distance is taken. This is illustrated in figure (2(a)) and written as

$$R_{ij} = \min(r_{ij}, r_{ji}) \quad (12)$$

where  $R_{ij}$  is the minimum of ratio,  $r_{ij}$  and  $r_{ji}$ ,  $r_{ij}$  is the ratio of the distances between the line of intersection and the boundaries of plane patch  $p_1$  and  $r_{ji}$  is the ratio of the distances between the line of intersection and the boundaries of plane patch  $p_2$ .



**Fig. 2.** The figures illustrates the calculation of the ratio measure and overlapping area measure in the computations of  $x_{ij}$ , the measure of plausibility of geometry between the two plane patches. The planes patches,  $AC$  and  $BD$  are shown in plan view for better clarity with the  $Z$ -direction into the paper. In the ratio measure calculation, the line of intersection passes through point  $I$ . The ratio,  $r_{ij}$  is  $\frac{AI}{CI}$  and the ratio,  $r_{ji}$  is  $\frac{BI}{DI}$ . The minimum of  $r_{ij}$  and  $r_{ji}$  gives the ratio  $R_{ij}$ . In the overlap area measure calculation,  $A_{i \rightarrow j}$  is the area bounded by  $EC$  of plane  $AC$  and  $A_{j \rightarrow i}$  is the area bounded by  $BF$  of plane  $BD$ .  $A_i$  is the area of plane  $BD$  and  $A_j$  is the area of plane  $AC$ .  $A_{ij}$  is computed as given in equation (11)

The above properties can be combined into a quantitative *measure of plausibility* of the joint configuration of the  $i^{th}$  and the  $j^{th}$  plane patch,  $x_{ij}$ , as

$$x_{ij} = 1 - \{W_{ij}(\rho_{ij}) * (A_{ij} + R_{ij})\} \quad (13)$$

where the terms are as defined above. By definition, plausible configurations should attain high values and suspicious configurations should attain low values.

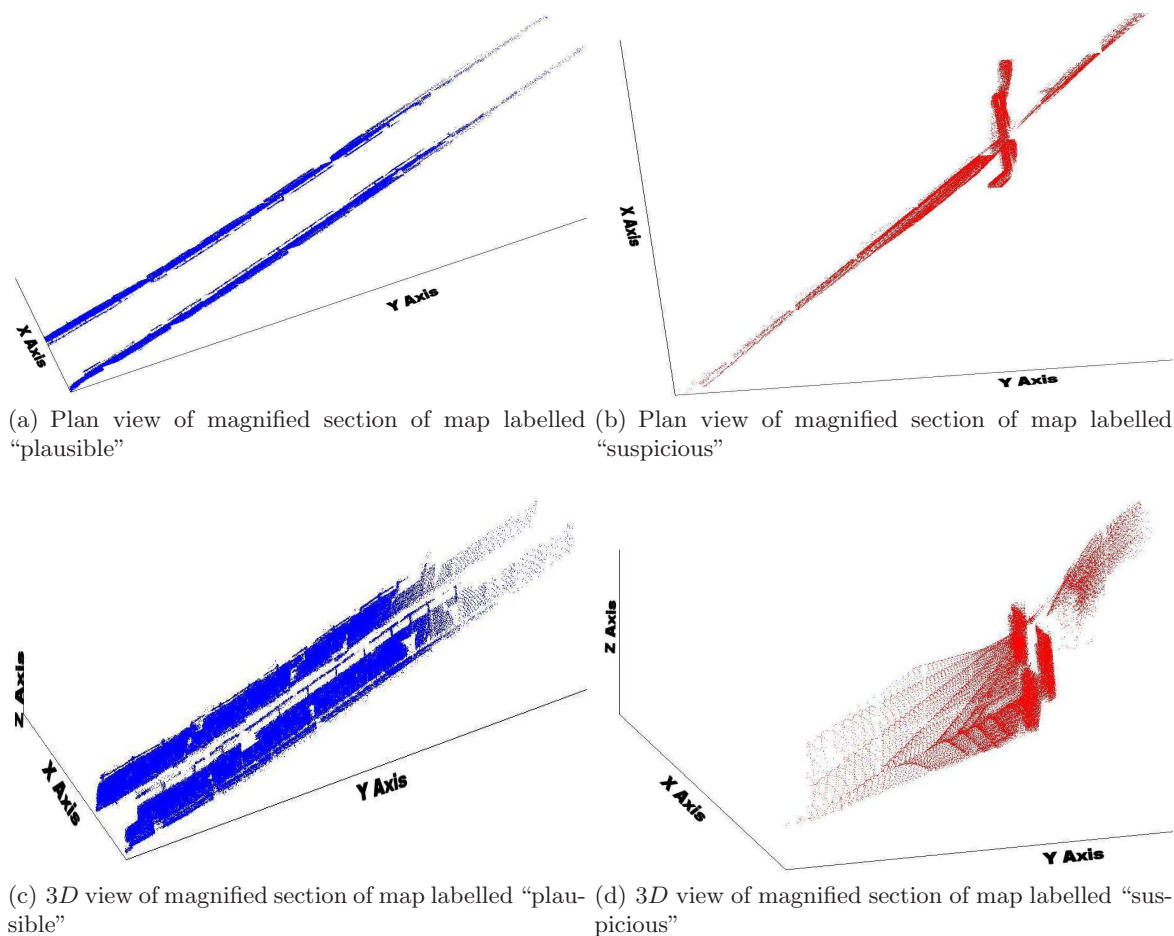
### 3.3 Model Learning and Inference

The set of CRF model parameters,  $\Theta$ , is learnt using a labelled training set of plane patches. For each patch in the set the unary features are evaluated using equ. 6 - 8 and the binary features (over a fixed size neighborhood) are evaluated using equ. 9 - 13. The so determined  $x_i$  and  $x_{ij}$  are substituted for  $f_k$  and  $g_k$  in equation 4 and the parameter set  $\Theta$  is learnt using maximum pseudo-likelihood estimation [15].

Once the model parameters have been learnt the conditional distribution  $p_{\Theta}(y|x)$  of new data can be classified by once again determining the relevant  $x_i$  and  $x_{ij}$  and substituting for  $f_k$ ,  $g_k$  and  $\Theta$  in equation 3. This conditional probability is then maximized using Graph Cuts optimization [6] in order to determine the most likely set of labels  $Y$  (equation 5).

## 4 Experimental Results

In this section, the effectiveness of the above algorithm is illustrated using data gathered in an urban environment by a mobile robot. A mobile platform equipped with a 3D SICK laser scanner was used to capture this



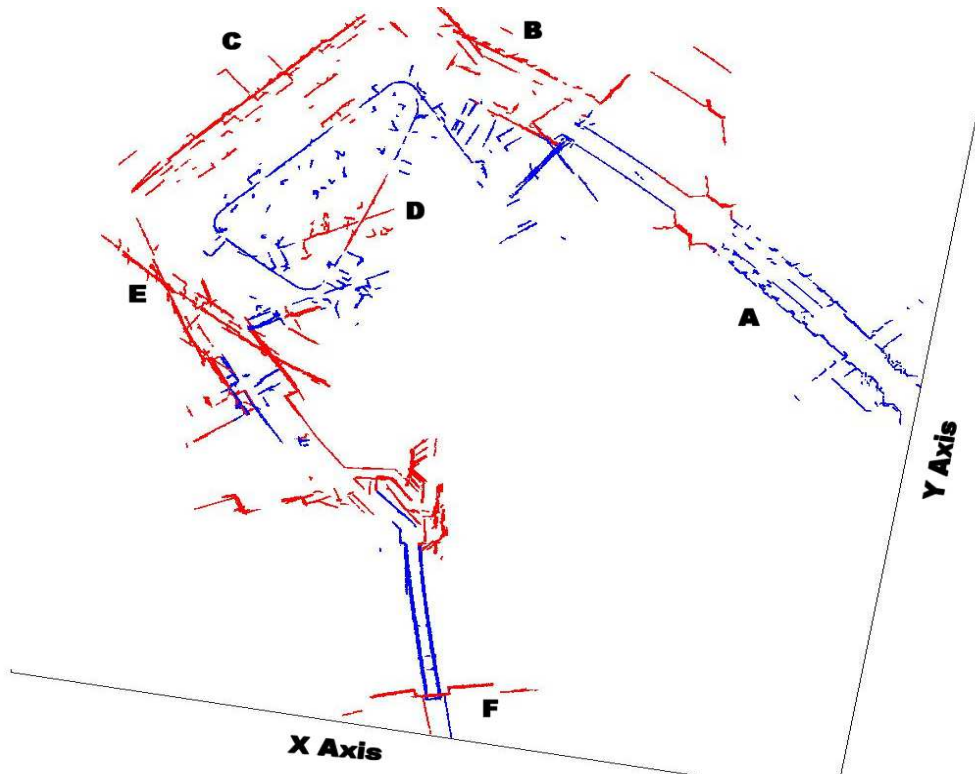
**Fig. 3.** Figure shows the magnified section of a map in 3D that has been classified by the algorithm. The “plausible” portions are marked in blue and the “suspicious” portions in red. For clarity, the plan view of this portion is also shown, above the magnified portion. In (a, c), the two plane patches are at a reasonable distance apart. Hence the algorithm correctly labels this portion as “plausible”. In (b, d), there are two planes that intersect, however not representing a corner. This is correctly labelled as “suspicious”.

data. The environment chosen was that of a typical urban environment consisting of office buildings, roadways, foliage, railings, people, moving cars and archways. The 3D point-cloud map built by the mapping algorithm was first segmented into plane patches using a region growing based approach [20]. Then these planes were manually labelled as “plausible” and “suspicious” based on how well the plane fitted the points and on the alignment of the planes with its neighbors. Approximately, 30,000 points segmented into 150 planes that were representative of the data, was then used in the training step. Maximum pseudo-likelihood using the CRFtoolbox<sup>1</sup> was run until convergence and used to learn the parameter set,  $\Theta$ . On inspection of the parameter set, it was found that the weight for the mean-squared error of fit feature (equation (6)) was a factor of two less than the weights for the other node features. This is because of the relatively good plane segmentation performed. The weight for the edge feature was factor of ten higher than the the weights for the node features. This makes the labelling more sensitive to the alignments between the plane patches, which is desired.

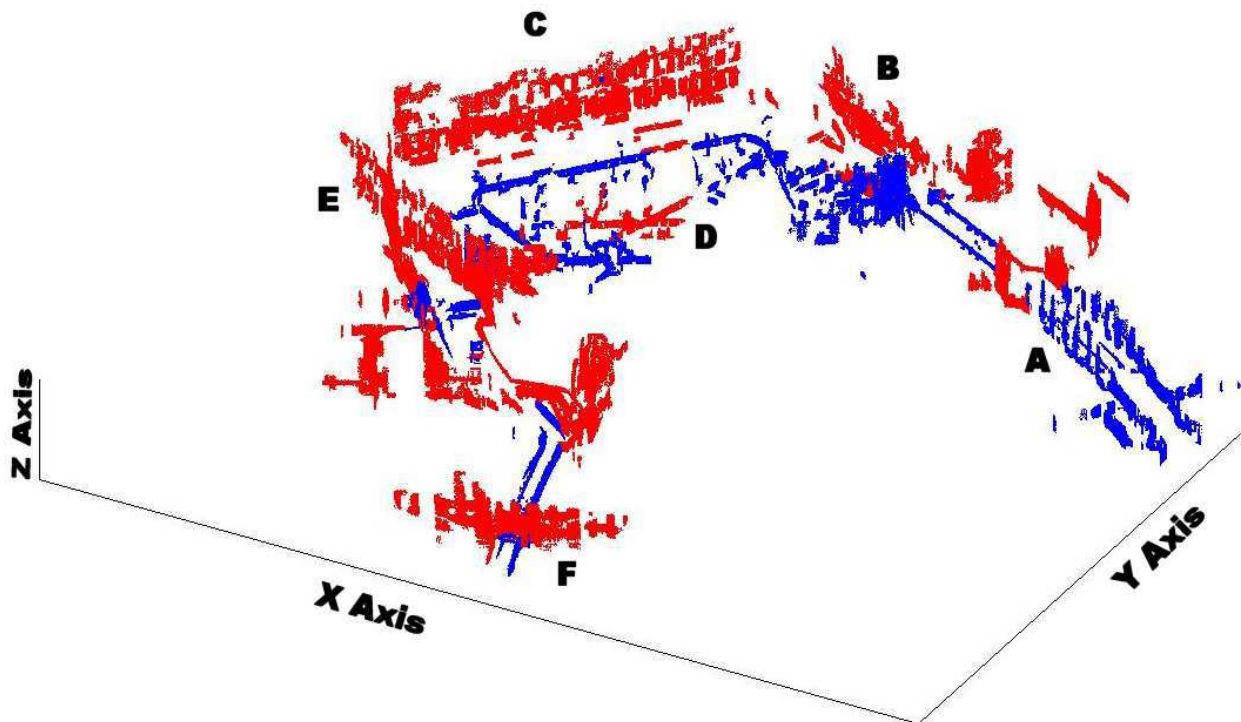
Using this parameter set,  $\Theta$ , the plane patches from the test data were then labelled as “plausible” and “suspicious” in the inference step using Graph Cuts. Figure (3) shows a magnified section of a map that has been classified by the algorithm. The “plausible” portions are marked in blue and the “suspicious” portions in red. For clarity, the plan view of this portion is also shown above the 3D view. In figures (3(a), 3(c)), the “plausible” region of the map consists of a stretch of wall on either side of the robot’s pathway. Since these two plane patches are at a reasonable distance apart, the classifier correctly labels this portion as “plausible”.

<sup>1</sup> <http://cs.ubc.ca/~murphyk/Software/CRF/crf.html>





(a) Plan view of the map



(b) 3D view of the map

**Fig. 4.** Figure shows a 3D view of a map consisting of 1323 planes that has been classified by the algorithm. The plan view is shown above the 3D view for better clarity. The same color code of blue for “plausible” and red for “suspicious” is used here. Of particular interest are the regions marked *A*, *B*, *C*, *D*, *E* and *F*. In region *A*, the point-cloud alignment is in keeping with the environment and the algorithm has correctly labelled this region as “plausible”. The regions *B*, *D* and *E* all have plane patches intersecting with each other due to the misalignment caused by the re-traversal of the robot. The three regions are correctly labelled “suspicious”. When two plane patches are parallel and very close to each other, it is labelled as “suspicious”. Hence region *C* is labelled “suspicious”. Region *F* actually represents an archway in the environment. Since the map quality algorithm considers only simple plane geometry, this region is incorrectly labelled as “suspicious”. Future work will include features to consider this case.

In figures (3(b), 3(d)), the two planes are intersecting. However, this does not represent a corner and it is not acceptable for two plane patches to intersect like this. These two plane patches are correctly labelled as “suspicious”.

Figure (4) shows a map consisting of 1323 planes that has been classified by the algorithm. The same color code of blue for “plausible” and red for “suspicious” is used here as well. The pathway of the robot consists of an initial stretch of road lined with office buildings. The robot then goes around a building twice before continuing further down the path under an archway. The point-cloud is rendered from odometry.

Of particular interest are the regions marked *A*, *B*, *C*, *D*, *E* and *F*. The region *A* shows the region in the map that is marked as “plausible”. The environment in this region consists of a office buildings lined on either side of the road. It is seen that the point-cloud alignment is in keeping with the environment and the algorithm has correctly labelled this region as “plausible” (blue in color). The next portion of the map shows the re-traversal of the robot on that path causing misalignment of the plane patches. The regions *B*, *D* and *E* all have plane patches intersecting with each other which are considered as “suspicious” alignment in the algorithm. These the algorithm labels correctly as “suspicious” (red in color). The region *C* is labelled as “suspicious” due to there being plane patches separated by a very small distance. The region *F* is actually an archway in the environment. The segmentation algorithm returns planes that are intersecting in this region. However, since the map quality algorithm considers only simple plane geometry, this region is considered as two plane patches intersecting at the middle of the patches and hence labelled “suspicious”. Future work will include features to consider cases such as archways as well. It is seen that the classifier labels the gross errors in mapping as “suspicious”.

**Table 2.** Confusion Matrix for the map shown in Figure(4) consisting of 1323 planes. The percentage of correct classification was found to be 80.42%.

Truth Inferred	Plausible	Suspicious
Plausible	94.18%	5.81%
Suspicious	34.48%	65.51%

The percentage of correct classification was found to be 80.42%. Table (2) summarize the results achieved in terms of a confusion matrix. When the classifier returns a “plausible” label for a plane patch, it was found that 94.18% of the times the correct label was “plausible”. The false positive rate was found to be 5.81%. Similarly, when the classifier returns a “suspicious” label for a plane patch, it was found that 65.51% of the times it was the correct label and the false negative rate was 34.48%. This is due to a conservative labelling of training data as only those planes that are definitely “plausible” should be labelled as that.

## 5 Conclusions

This work addresses the problem of measuring the quality of maps in retrospection. A measure of map quality is necessary for assessing how accurately the environment has been mapped, once the map is built. In this paper, a 3D point-cloud map is classified into binary classes of “plausible” and “suspicious”. The problem of classifying the regions of the given map is modelled as a Conditional Random Field framework, by segmenting the map into plane patches and considering these plane patches as nodes. Since CRFs capture the neighborhood dependencies, the alignment between patches also contribute to the classification. A probability distribution of the array of labels is obtained which is maximized using graph cuts. The method is illustrated using data gathered by 3D laser scanners from an urban environment. The algorithm labels the gross errors in mapping as “suspicious”. The particular case of archways is erroneously classified as “suspicious”. Future work will include features that considers this case as well.

## 6 Acknowledgments

The authors would like to thank Dr.Ingmar Posner for his comments during this work as well as Mark Cummins for his help with the labelling of the laser data.

## References

1. Dragomir Anguelov, Ben Taskar, Vassil Chatalbashev, Daphne Koller, Dinkar Gupta, Jeremy Heitz, and Andrew Ng. Discriminative learning of markov random fields for segmentation of 3d scan data. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 169–176, Washington, DC, USA, 2005. IEEE Computer Society.
2. J Besag. Statistical analysis of non-lattice data. 24:179–195, 1975.
3. J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48(3):259–302, 1986.
4. Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
5. Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.
6. Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.
7. Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
8. S. Geman and D. Geman. Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 6(6):721–741, 1984.
9. J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Int. Symposium on Computational Intelligence in Robotics and Automation*, 1999.
10. D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
11. R. Kinderman and J Snell. Markov random fields and their applications. *American Mathematical Society*, 1980.
12. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004.
13. John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
14. John J. Leonard, Hugh F. Durrant-Whyte, and Ingemar J. Cox. Dynamic map building for an autonomous mobile robot. *Int. J. Rob. Res.*, 11(4):286–298, 1992.
15. Lin Liao, Dieter Fox, and Henry Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *The International Journal of Robotics Research*, 26(1):119–134, 2007.
16. R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. *European Conference on Computer Vision*, 2:16–29, 2006.
17. R. Triebel, K. Kersting, and W. Burgard. Robust 3D scan point classification using associative markov networks. In *Proc. of the International Conference on Robotics and Automation(ICRA)*, 2006.
18. Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2002.
19. Hanna M. Wallach. Conditional random fields: An introduction, 2004.
20. J. Weingarten, G. Gruener, and R. Siegwart. A fast and robust 3D feature extraction algorithm for structured environment reconstruction. In *Proc. of the 11th International Conference on Advanced Robotics (ICAR)*, 2003.