2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

A Segmentation Guided Label Propagation Scheme for Autonomous Navigation

Soumya Ghosh Jane Mulligan

Abstract— **Navigating through unknown outdoor environments requires a robot to be able to see and model the far field terrain. In recent years this problem of seeing beyond reliable stereo readings into the far field has gained attention. Many proposed solutions involve using near field obstacle and ground plane regions labeled using stereo, to learn models which classify far field image regions. In this work we offer an alternative which exploits coherent image regions as determined by image segmentation to propagate obstacle and ground labels from the near and mid field to the image far field. Rather than relying on local features to classify individual pixels we model and compare appearance across the whole segment. New labels are determined by proximity in both image space and appearance space. Since both traversable and non-traversable surfaces can vary in appearance across the image, our approach has the advantage that each labeled segment acts as a distinct appearance model, which allows us to label similar neighbours. We evaluate our system using a publicly available dataset and compare its performance to a typical learning-based near-to-far labeling scheme.**

I. Introduction

Autonomous navigation in unstructured environments requires the ability to distinguish traversable ground plane and obstacles over which the robot cannot pass. Range data from laser or stereo systems was exploited by many early systems to model environment geometry and identify obstacles [1], [2]. Laser range sensors can fail due to outdoor surface or lighting conditions, while at typical resolutions and baselines for mobile platforms, stereo traversability analysis is limited to near field ranges of 5-10 meters. These limitations led to the idea of augmenting purely geometric terrain identification by assuming a local association of traversable and non-traversable surfaces with their image appearance [3], [4], [5].

More recently the DARPA Learning Applied to Ground Robots (LAGR) program generated interest in far field terrain identification [6], due to the myopia [7] of autonomous systems. Typically these approaches assume a locally smooth drivable surface can be modeled by fitting a plane to near-field stereo geometry measurements, and used to label near field image pixels as obstacle and ground plane. This labeled data can be used to learn appearance models which are then applied to the far field [8], [9], [10]. A variety of appearance features and learners have been applied, but the basic near-to-far approach has demonstrated that image colour and appearance reliably reflect terrain type [11], [12], [13], [14].

This work was supported by the DARPA LAGR Program (DOD AFRL award no. FA8650-07-C-7702) and NSF CNS-0430593.

S. Ghosh is with the Department of Computer Science, Brown University, sghosh@cs.brown.edu

J. Mulligan is with the Computer Science Dept, University of Colorado at Boulder, Jane.Mulligan@colorado.edu

In this paper, we propose a segmentation guided label propagation algorithm for predicting far-field traversability. As with previous approaches we depend on geometric modeling of the ground plane using calibration and stereo to label the traversability of near- and mid-field terrain. We treat the available mid- and far-field stereo labels as seeds from which class labels are propagated further into the far field. The propagation occurs along natural regions of image homogeneity as extracted by a segmentation procedure. Labels are allowed to propagate freely inside image segments. Labels can also propagate across segment boundaries as long as the segments being propagated to are "close" in both physical space and appearance to the set of pixels with stereo labels. The following section specifies the notion of "closeness" and explains the algorithm in greater detail. Algorithm 1 presents a formal description of our algorithm.

The use of segmentation to guide the classification process may be seen as a form of "weak-supervision", and is particularly useful when the available stereo labels are sparse.

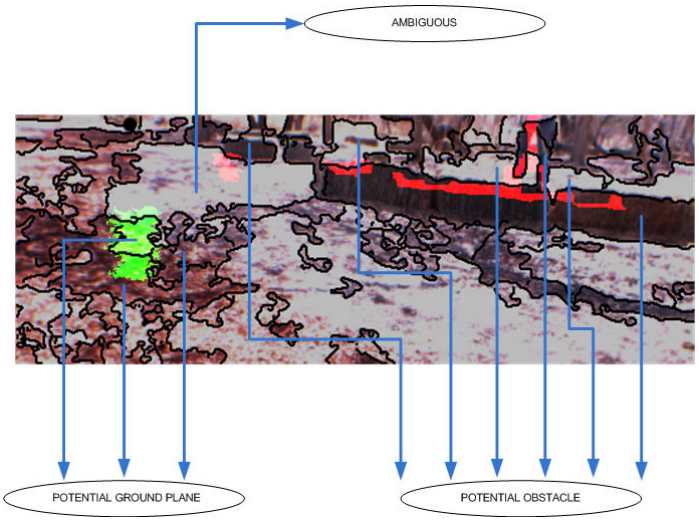


Fig. 1. Stereo predictions corresponding to traversable ground plane are highlighted in green and those corresponding to obstacles are displayed in red. Regions not highlighted are deemed as unknown by stereo.

II. Label Propagation Approach

Our approach exploits the spatial structure of the data. We assume that Tobler's first law of geography which states "Everything is related to everything else, but near things are more related than distant things" [15] holds. Furthermore, we observe that when the class labels available are too sparse

to accurately model the class, high precision(albeit at the expense of recall) can still be achieved by predicting the labels of only those data instances which are "most" similar to the ones with labels. These observations motivate a conservative approach where we restrict the classification process to the immediate spatial and appearance neighbourhood of the labeled data, deeming everything else as unknown.

Segmentation splits an image into a number of mutually exclusive and exhaustive regions (segments) based on the underlying image structure, thus extracting natural intra-image boundaries. These boundaries provide a measure of the extent of underlying image homogeneity and hence the extent to which labels should be propagated (under the assumption that one does not want to propagate labels across inhomogeneous regions). We enforce spatial "closeness" by restricting the classification process to partially labeled segments and their immediate neighbors.

A. Algorithm Overview

The first step of the algorithm involves segmenting the far field region of an image frame into a collection of segments Ω. These are combined with ground/obstacle pixel labels from the robot's stereo system. We will denote the set of all pixels labeled ground/obstacle by stereo as S_g/S_o. Segments which have any overlap with S_g and segments immediately neighboring such segments make up the set *Candidate Ground* C_g and those which overlap S_o and segments immediately neighboring them make up *Candidate Obstacle* C_o. If a segment overlaps (or neighbors) both ground and obstacle labels then it is deemed ambiguous and belongs to the set *Candidate Ambiguous* C_{amb} (See Fig. 1).

All segments in C_g which are closer to $model(S_g)$ than some threshold d_g and those in C_o that are closer to $model(S_o)$ than some threshold d_o in some feature space f, according to some similarity measure D, are labeled as Ground plane and Obstacle respectively. Finally, segments in $C_g \cap C_o$ which are closer to both the stereo ground plane and the stereo obstacle regions than their respective thresholds, (in other words, segments which can be labeled as either Ground plane or Obstacle) are deemed ambiguous and are classified as unknown. The steps in this process are detailed in Algorithm 1.

Algorithm 1 Far Field classification algorithm

for all image frames **do**
  Extract and Segment the far field region of the image frame. Let Ω denote the collection of segments thus produced.
  Use available stereo labels and Ω to compute C_g and C_o.
  Compute $C_{amb} = C_g \cap C_o$ and recompute $C_g = C_g \setminus C_{amb}$; $C_o = C_o \setminus C_{amb}$.
  for all $\gamma \in \{o, g\}$ **do**
    $\forall c_\gamma \in C_\gamma$ compute distance $d_{c_\gamma} = D(c_\gamma, model(S_\gamma))$, where D is a similarity measure in feature space f and $model(S_\gamma)$ is a model[1] representing S_γ.
    Compute similarity threshold d_γ.
    if $(d_{c_\gamma} \leq d_\gamma)$ **then**
      Label c_γ as γ.
    else
      Label c_γ as unknown.
    end if
  end for
  $\forall c_a \in C_{amb}$ compute $d_{c_{ga}} = D(c_a, model(S_g))$ and $d_{c_{oa}} = D(c_a, model(S_o))$.
  if $(d_{c_{ga}} \leq d_g) \wedge (d_{c_{oa}} \leq d_o)$ **then**
    Classify c_a as unknown.
  else if $(d_{c_{ga}} \leq d_g)$ **then**
    Classify c_a as ground plane.
  else if $(d_{c_{oa}} \leq d_o)$ **then**
    Classify c_a as obstacle.
  else
    Classify c_a as unknown.
  end if
end for

B. Segmentation Algorithms

We compare two segmentation algorithms – the efficient graph based segmentation algorithm and the mean shift based segmentation algorithm. The choice of the algorithms was governed by the need for near real time performance. We briefly explored a contiguity enhanced version of K-means [16] image segmentation. However, K-means can prove to be very slow especially with large values of K as is often desired in our setting.

1) Efficient Graph based segmentation: The first segmentation algorithm we consider here is the Efficient Graph Based segmentation algorithm, introduced in [17]. As the name suggests, this algorithm treats the image as a graph, with the pixels acting as the vertices. Segmentation is achieved by splitting the image into a collection of connected components. A minimum spanning tree of the graph is constructed and all edges below an adaptive data dependent threshold are removed from the graph.

2) Mean shift segmentation: The mean shift [18] algorithm is a feature space analysis technique popularly used for image segmentation. The algorithm involves first mean shift filtering of the image data in some feature space followed by a hierarchical clustering of the filtered data. In this paper we have used the open source EDISON [19] implementation of the mean shift segmentation. The EDISON system converts the original RGB image into the LUV space. The mean shift filtering is carried out in a 5 dimensional feature space, containing the (x, y) image coordinates and the LUV values.

C. Segment proximity measure

Appearance "closeness" is measured in a feature space f, which captures the appearance of a segment. Here, we use colour histograms(on RGB colour space) to represent segments. Each segment is represented by a 30 bin histogram, with each colour channel occupying 10 bins. Measuring segment appearance "closeness" now reduces to computing histogram similarity. There are several popular distance mea-

sures available in the literature [20]. In this paper, we use the sum of minimum bin measure, where the distance between two segment histograms is defined as:

$$D(H_1, H_2) = \sum_{i=1}^{N_{bins}} I(min\{h_1(i), h_2(i)\}) \quad (1)$$

where i indexes the bins of the histograms of the two segments. It is implicitly assumed that H_1 and H_2 have been normalized by the total number of pixels in the corresponding segments.

D. Data Modeling

We model each class as a mixture of components, where each component is the histogram(30 bins) of a contiguous stereo labeled region. The ground plane class is modeled as a mixture of histograms of contiguous regions made up of the pixels belonging to set S_g while the obstacle class is modeled as a mixture of histograms of regions made up of the elements of S_o. Figure 2 provides a pictorial depiction of the class modeling. Formally, our model may be expressed as a mixture model:

$$H_s = \sum_{k=1}^{K} \eta_{sk} H_k \quad (2)$$

where k is the number of components and $\eta_{\mathbf{s}} = \{\eta_{s1}, \eta_{s2} ... \eta_{sk}\}$ represents the mixing weights of the different components. It is important to note that the histogram mixture is indexed by $s \in \Omega$. This implies that we can have a different model representing the same class depending on the particular s being classified. The histogram mixtures differ by having different component histograms mix according to input dependent mixing proportions η_{sk}, where $\eta_{sk} \geq 0$ and $\sum_{k=1}^{K} \eta_{sk} = 1$. Furthermore, our model employs semi parametric histograms as opposed to parametric models (such as Gaussian) for modeling the components. This allows us to abstain from making assumptions about the data generation process.

Ideally, the mixing weights would be learnt through a Expectation Maximization algorithm. However, such an iterative algorithm would have to be run for each segment which is not desirable for a near real time algorithm. Instead, we take a simpler route and assign the mixing weights in a heuristic fashion. We experiment with three different schemes for assigning the mixing weights $\eta_{\mathbf{s}}$. The first model which will be referred to as the **Global** model assumes that all components are equally important and thus allots equal weights to all components.

$$\eta_{sk}^{global} = \frac{1}{K}; \quad k = 1..K \quad \forall s \in \Omega \quad (3)$$

At the other end of the spectrum we have the **Local** model. In this model only the nearest component to s has a weight of 1, with all other components set to zero.

$$\eta_{sk}^{local} = \begin{cases} 1 & \text{if } k \text{ is the (spatially) nearest component to } s \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Finally, we have the **Semi-Local** model, which is a compromise between the local and the global models. The local model gives absolute importance to the closest component, while completely disregarding all other components. The global model on the other hand does not take spatial proximity into account at all. The semi-local model reconciles the two models by fitting an exponential distribution[2] on $\eta_{\mathbf{s}}$. The weight of a component decays exponentially with its spatial distance δ from the segment to be classified.

$$\eta_{sk}^{semilocal} = \frac{\frac{1}{\delta_{sk}}}{\sum_{k=1}^{K} \frac{1}{\delta_{sk}}}; \quad k = 1..K \quad \forall s \in \Omega \quad (5)$$

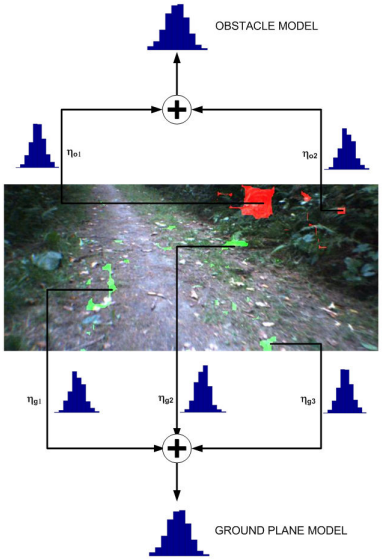


Fig. 2. Data modeling illustration.

E. Threshold Selection

The final issue of interest in the algorithm is that of determining the appropriate thresholds d_g and d_o. We select these values by segmenting the mid field region of each image frame. Segments which transgress class boundaries (as determined by stereo labels) are split such that each segment belongs solely to one class. Next, the histogram distance between all segments of the same class is computed. Further, the mean ($\bar{d}$) and standard deviation (σ) of the intra class distances is computed. Finally, the distance threshold is computed as:

$$d_{th} = \bar{d} - \sigma \quad (6)$$

where d_{th} corresponds to either d_g or d_o.

We compute the thresholds over the mid field region because the availability of stereo labels here is considerably larger than in the far field, thereby making the above intra class distance statistics meaningful. The motivation behind choosing this threshold is that images smooth out in the far field. This smoothing causes there to be less discriminability

[2]In our experiments, we set the rate parameter of the distribution λ to 1.

(less intra class distance) amongst segments in the far field as opposed to segments in the mid field. In practice, we have found that the scheme described above works quite well.

III. Experimental Setup

The experiments presented in this section are designed to explore the three main components of the approach: the segmentation algorithm, the proximity weighting model and the overall far field propagation method. We examine the system's performance under a range of parameters for both Mean Shift and graph-based segmentation and consider how the image datasets affect the performance of each segmenter. In order to evaluate the performance of our far-field labeling system we compare it to available ground/obstacle labels from Stereo and to a near-to-far Support Vector Machine (SVM) labeling approach [21][11], which uses Stereo labeled pixels from the mid field to learn to classify far field pixels.

A. Data

We tested our algorithm on six data sets, with each data set containing 100 image frames extracted from the log files recorded during live robot test runs in the DARPA LAGR program. The data sets were carefully chosen to represent different terrain and lighting conditions. The terrain varies greatly over the data sets, with ground plane varying from mulch to dirt and obstacles varying from foliage to trees and hay bales. Overall, three scenarios are considered (DS1,DS2,DS3). Each scenario is associated with two distinct data sets taken under different lighting conditions (A and B). Representative frames from the six data sets are shown in Figure 3.

We evaluated the performance of the competing algorithms by comparing their predictions with pixel level hand labeled ground truth data made available by Procopio et al. [22]. Each pixel of an image frame is labeled as one of three classes - Ground plane, Obstacle or Unknown. We compute our performance metrics only over the pixels in the far field region. This provides for a fair comparison of the far field performance of the competing algorithms.

B. Performance Metrics

To measure the performance of the competing algorithms we borrow popular information retrieval metrics of precision and recall.

Precision is the ratio of the number of relevant objects in a retrieved set to the total retrieved set. Informally, precision may be thought of as a measure of the noise in the retrieval process. It is calculated as:

$$Precision = \frac{TP}{(TP + FP)} \tag{7}$$

where TP(true positives) is the number of far field pixels correctly classified as GroundPlane or Obstacle and FP is the number of far field pixels mis-classified as GroundPlane or Obstacle.

Recall is the ratio of the number of relevant objects in a retrieved set to the total number of relevant objects.Intuitively

(a) DS1A (b) DS1B

(c) DS2A (d) DS2B

(e) DS3A (f) DS3B

Fig. 3. Example frames from the six image data sets.

it is a measure of the exhaustiveness of the retrieval process. Recall is calculated using:

$$Recall = \frac{TP}{(TP + FN)} \tag{8}$$

where FN is the number of far field pixels which are classified as unknown, but belong to either the Ground or Obstacle class in the ground truth data. Finally, for convenient comparison of the different algorithms we use the F_1 measure which is just the harmonic mean of precision and recall. This value is computed as follows:

$$F_1 = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \tag{9}$$

C. Comparison Systems

a) Stereo Labels: Stereo labels are the baseline we compare against. This comparison is a basic sanity check which establishes whether there is any benefit to using the proposed algorithm in a real world scenario (Fig. 4(a)).

b) Support Vector Machines: The most common approach to far field terrain labeling is currently a near-to-far learning approach, where near or mid-field Stereo labeled samples are used to train models for labeling far-field pixels. We compare our approach to both linear and Gaussian SVMs (Figs. 4(e) and 4(f)). Both Linear [21] and Gaussian SVMs have been used for far field predictions in the past. Furthermore, Gaussian SVMs were found to be particularly well suited for this problem [11]. We use the one model per

image (ONEMI) paradigm, in which a SVM model is trained for each image of a data set. The features used are colour histograms computed in a 3×3 window around the training pixel. We briefly experimented with other window sizes, but the results were to a large extent invariant to the choice of window size.

To make the sparsity assumption hold, the SVMs were trained only on mid and far-field stereo labels. This training scenario also provides a fair comparison between the SVM and our proposed algorithm, which also uses only mid and far field stereo labels. The liblinear [23] implementation for the linear SVM was used with L2-regularized logistic regression solver. The Libsvm [24] implementation of the Gaussian SVM was used, and the γ parameter was determined via cross validation. The SVMs were forced to output probabilities using Platt scaling[25] as implemented in the libsvm package. Predictions where the SVM was < 0.55 confident about either class were classified as unknown. This probability cut off was determined empirically.

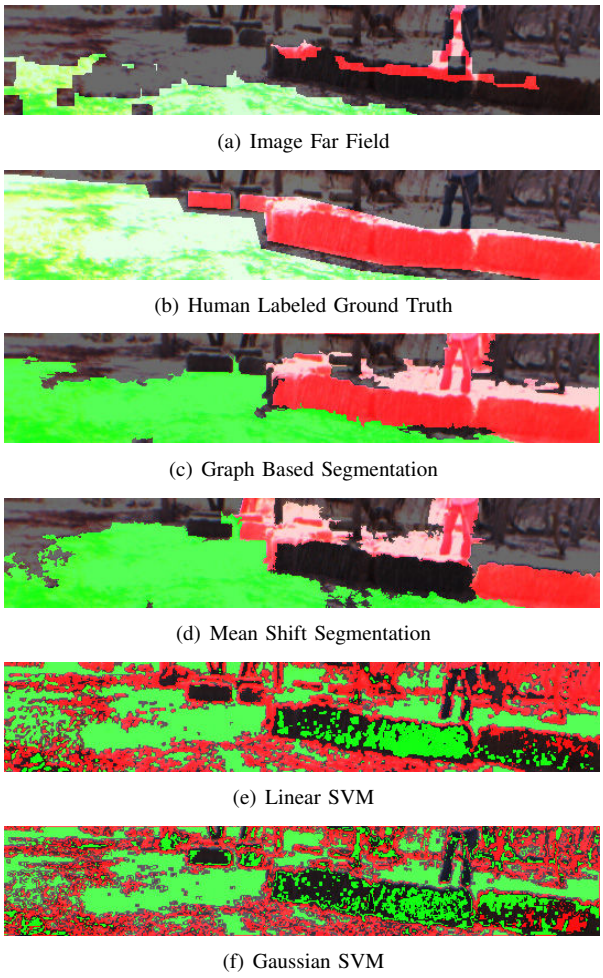
IV. Results

We begin this section, by illustrating the performance of our algorithm on an image from data set DS1A. Figure 4 displays the far field predictions using Gaussian and Linear SVMs and our proposed algorithm. As is evident from the illustration, our algorithm is able to successfully build on sparse stereo labels, to find significant far field structure. Both segmentation strategies improve upon the initial stereo predictions. The SVM predictions however produce worse fragmented results. In this particular case, the labels of either class display similar colour, resulting in large amounts of confusion in the SVM predictions. Since, the SVM ignores all context information beyond the pixels used to compute the histogram features, it has no way of disambiguating pixels with similar colour content. On the other hand, our algorithm explicitly leverages neighbourhood information and by fusing this contextual information with colour histogram information it is able to disambiguate ambiguous regions.

In Figure 4, the effect of "lack of context" is clear, with the SVMs producing noisy speckled classifications as opposed to the smooth classifications produced by our algorithm.

To quantitatively demonstrate system performance using the data sets, we present performance plots in figures 5 through 7. As is evident from the figures, our algorithm, using any of the three models, significantly outperforms the base line stereo predictions on all six data sets. Additionally, we also outperform the Gaussian and the Linear SVM on all data sets. These performance gains are even more significant when we consider that they are over the far field and correspond to a substantial distance in the real world.

In general, we achieve better performance in data sets displaying no lighting variations than those with lighting variations. This is hardly surprising, since lighting changes result in higher intra class variations, especially in the absence of colour constancy. Our colour histogram feature space is built on top of the non-normalized RGB colour space, which fuses brightness and colour content together, making it vulnerable to confounding colour differences and brightness differences.



(a) Image Far Field

(b) Human Labeled Ground Truth

(c) Graph Based Segmentation

(d) Mean Shift Segmentation

(e) Linear SVM

(f) Gaussian SVM

Fig. 4. Far Field predictive performance of various algorithms on an image from DS1A. Green regions correspond to ground plane, red to obstacles and black to unknown.

A. Global, Local or SemiLocal model

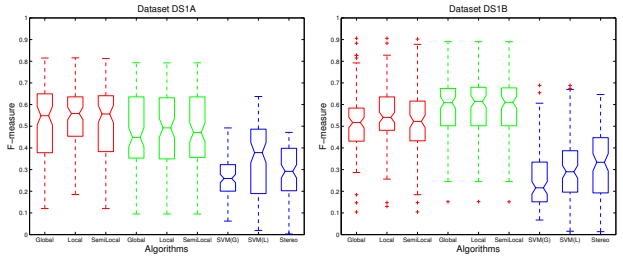
This section analyzes the effectiveness of the three proposed proximity weighting algorithms: the global, local and the semi-local models. In order to experimentally ascertain the effectiveness of these models, we average out the performance of each model across all the parameter values, for each of the six data sets. The resulting F_1 measure values are displayed as box plots in figures 5 through 7. The plots in red depict results of the three models using graph based segmentation while those in green correspond to the mean shift algorithm. Blue box plots represent the algorithms being compared against.

In data sets DS1A through DS2B the three models produce statistically indistinguishable results (at 95% confidence level). This similarity in performance is hardly surprising when we explore the structure of the stereo labels in these data sets. In a large majority of the image frames in these data sets, the stereo labels of either class form one large contiguous region, instead of several smaller fragmented

regions (see figure 4(a)). Under such circumstances, both the global and the semi-local models behave as the local model.

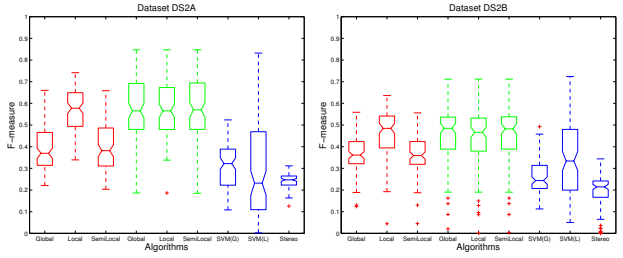
In data sets DS3A and DS3B however, the stereo labels are more fragmented. Both data sets have fairly uniform within-class colour content. However, DS3B exhibits darker lighting conditions . Figure 7(a) shows that the global and the semi-local models outperform the local model in DS3A. This reinforces our intuition that in cases where the intra class variance is low, a global data model aids the classification process. DS3B, which exhibits less vivid colours and poorer class discriminability, causes the global model to perform worse than the semi-local or local models. This effect is particularly visible when the models use graph based segmentation.

In general, it appears that the choice of the proximity model depends on the data set. We expect the local model to perform well for data sets with higher intra class variance, while the global model is expected to work better with comparatively smaller intra class variance. The semi local model should work best for cases with an intermediate amount of intra class variance. The experiments presented in this section though, remain somewhat inconclusive and fail to fully corroborate the above intuitions, primarily due to the lack of fragmented stereo labeled regions in the first four data sets.



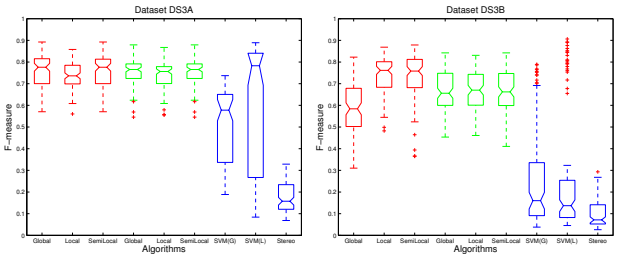
(a) (b)

Fig. 5. F_1 measure of competing algorithms across datasets DS1A and DS1B. The F_1 values for each model have been averaged over all segmentation parameter settings.



(a) (b)

Fig. 6. F_1 measure of competing algorithms across datasets DS2A and DS2B. The F_1 values for each model have been averaged over all segmentation parameter settings.



(a) (b)

Fig. 7. F_1 measure of competing algorithms across datasets DS3A and DS3B. The F_1 values for each model have been averaged over all segmentation parameter settings.

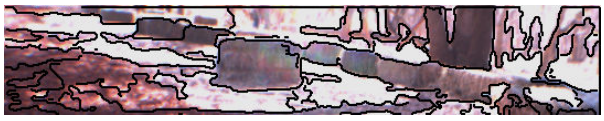
B. Analysis of segmentation results

In this sub-section we compare the two approaches to image segmentation utilized by our proposed algorithm. The primary points of investigation are: the performance of the segmentation algorithms and the sensitivity of the results to the segmentation parameter settings.

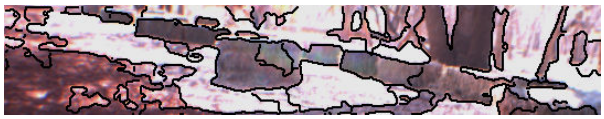
To answer these question the segmentation algorithms described in section II-B are compared against one another. To make the comparisons tangible, we make certain simplifying assumptions and fix certain parameters. The graph based segmentation's implementation is controlled by two parameters [17] the minimum number of pixels in a segment min and a parameter m which controls how aggressively pixels are merged into a segment. As suggested in [26], in our experiments we tie these two parameters together $min = m = K$ and vary K over the range $\{50, 100, 150, 200, 250, 300, 350\}$. The mean shift algorithm [18] has three tunable parameters, the spatial bandwidth (h_s), the colour bandwidth (h_r) and the smallest significant feature size (M). In the experiments presented here we fix $h_s = 7$; we found the produced segmentation to be fairly insensitive to spatial bandwidth. Furthermore, we found that fixing $h_r = 7$ and varying $M = K = \{50, 100, 150, 200, 250, 300, 350\}$ produces segmentations which are best comparable to the ones produced by the graph based segmentation. Fixing the colour bandwidth h_r results in more stable segmentations, reducing fluctuations in segmentation granularity. Our inclination toward fairly high values of h_r and M are necessary to discard the effects of small variations present in the kind of complex images that are dealt with in this paper. From here on, we refer to the tunable parameter of the mean shift algorithm, simply as K.

Our objective is to classify image far-field as accurately as possible and not necessarily to produce the best possible segmentation along the way. A segmentation is only as good as the final prediction it produces. Hence, we simply use the far-field prediction performance to evaluate the performance of a segmentation algorithm. The segmentation stability is assessed by evaluating how the performance and the number of segments varies by varying the segmentation parameters.

Figure 8 illustrates the segmentations produced by the two algorithms on representative image frames from each of the 3 environments.

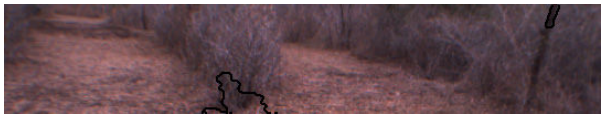


(a) Graph Based DS1A

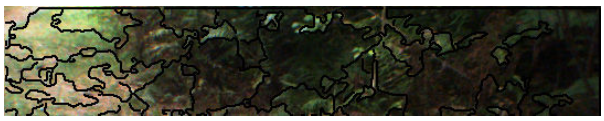


(b) Mean Shift DS1A

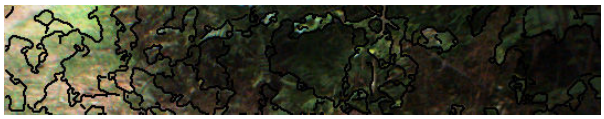
(c) Graph Based DS2A



(d) Mean Shift DS2A



(e) Graph Based DS3A



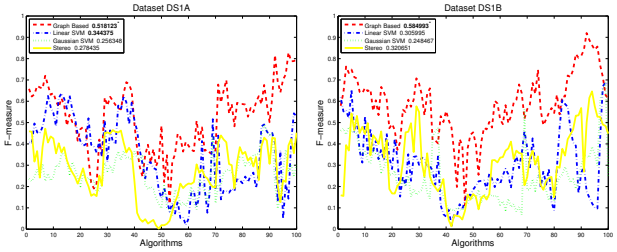
(f) Mean Shift DS3A

Fig. 8. Typical segmentations produced by the two segmentation algorithms on the first "A" datasets. Parameters were chosen such that each segmentation algorithm produces approximately equal number of segments

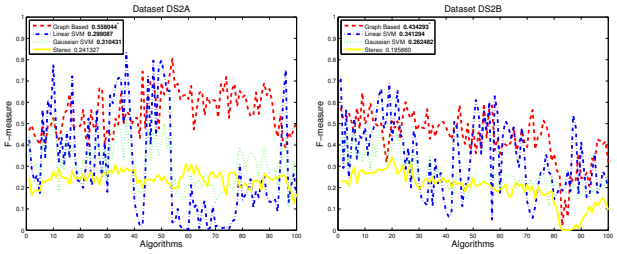
1) Segmentation parameter sensitivity analysis: Increasing value of K, causes the segmentation resolution to progress from fine to coarse for either segmentation algorithm. This progression is more rapid in the case of the graph based segmentation algorithm, which covers a wider gamut of segmentation resolution in the explored parameter space. Optimal predictive performance is achieved when the segmentation is neither too fine nor coarse. In the graph based algorithm's case this optimal segmentation resolution is achieved over the $K = [100, 150]$ range. The mean shift segmentation achieves optimality at $K = [300, 350]$ (with h_r and h_s fixed at 7). Interestingly, for most (except DS1A) datasets the number of segments produced by the mean shift algorithm at its optimal K values of $[300, 350]$, is in the same range as the number of segments produced by the graph based algorithm at it's optimal parameter settings of $K = [100, 150]$.

2) Comparison of Segmentation performance across datasets: For each dataset, at each parameter setting the algorithm's performance is averaged over the global, local and semi local models. Our analysis leads us to conclude that either segmentation produces optimal predictive performance at a certain range of its K values, which corresponds to a segmentation resolution where the image is neither over nor under segmented. For the graph based case this optimal range is achieved by setting $K = [100, 150]$, while for the mean shift algorithm this range is $[300, 350]$. The actual number of segments produced at these ranges varies across datasets, depending on dataset complexity. This indicates that the optimal segmentation resolution is dataset dependent. It might be beneficial to use the mean shift algorithm for datasets exhibiting low far field colour variation and the graph based segmentation for the ones exhibiting higher amounts of far field variation.

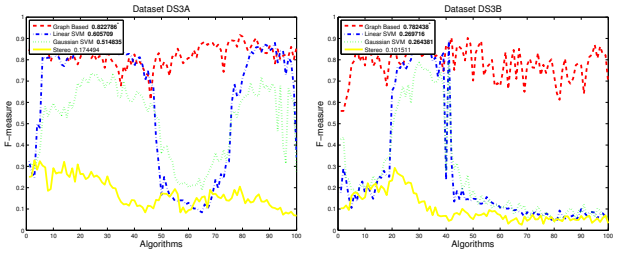
Finally, Figure 9 presents the optimal performance achieved by the proposed algorithm. Graph based segmentation was used and the results were averaged over parameter settings of $K = [100, 150]$. Here, we plot the performance of our algorithm averaged over the local, global and semi local models, across the 100 frames of the data sets, which further illustrates the superior performance of our algorithm.



(a) DS1A (b) DS1B



(c) DS2A (d) DS2B



(e) DS3A (f) DS3B

Fig. 9. Optimal performance for each data set across constituent frames. The legend displays mean F_1 values. **Bold** indicates that the performance is statistically better than stereo, as determined by a paired t-test at 95% confidence interval. Asterisk next to an algorithm indicates that it produces the best results amongst all competing algorithms (again as determined by a paired t-test at 95% confidence).

V. Conclusion and Future Work.

In this paper we present a novel segmentation guided far field classification algorithm. We find our algorithm to be particularly attractive when available stereo labels are sparse and poorly sampled. We find our algorithm to be robust under such conditions. This robustness stems from having imposed strong spatial constraints on the classification process. Although, imposing such constraints may reduce recall rates in some settings, we are able to achieve high precision, which we deem to be more important for far field classification purposes. One approach to boosting recall rates, would be to explore the tension between the spatial and feature proximities, to identify the extent to which the strength of the imposed spatial constraints might be relaxed without significantly impacting precision. This is a planned extension to the current piece of work. Another area of future work lies in incorporating temporal context in addition to spatial context in the classification process. We have noticed that at times the availability of stereo labels varies drastically across the dataset, resulting in variance in the far field classification performance. We believe by incorporating temporal context in our algorithm such intra dataset variance can be minimized.

Acknowledgments

The authors would like to thank Wei Xu for his comments on the paper and Arunima Chaudhary for her help with the figures.

References

[1] L. Matthies and P. Grandjean, "Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, Dec. 1994.

[2] C. M. Shoemaker and J. A. Bornstein, "The demo iii ugv program: a testbed for autonomous navigation research," in *Proc Intl. Symp. on Intelligent Control*, 1998, pp. 644–651.

[3] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection adn terrain classification for autonomous off-road navigation," *Autonomous Robots*, vol. 18, pp. 81–102, 2005.

[4] P. Bellutta, R. Manduchi, L. Matthies, K. Owens, and A. Rankin, "Terrain perception for demo iii," in *Proc. Intelligent Vehicles Symposium*, 2000, pp. 326–331.

[5] L. Matthies, A. Kelly, T. Litwin, and G. Tharp, "Obstacle detection for unmanned ground vehicles: a progress report," in *Proc of the Intelligent Vehicles Symposium*, 1995, pp. 66–71.

[6] L. D. Jackel, E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan, "The darpa lagr program: Goals, challenges, methodologies, and initial results," *Journal of Field Robotics*, vol. 23, no. 11-12, 2006.

[7] B. Nabbe and M. Hebert, "Extending the path-planning horizon," *International Journal of Robotics Research*, vol. 26, no. 10, pp. 997–1024, 2007.

[8] M. J. Procopio, J. Mulligan, and G. Grudic, "Long-term learning using multiple models for outdoor autonomous robot navigation," in *Proc. Intl. Conf on Intelligent Robots and Systems (IROS07)*, 2007, pp. 3158 – 3165.

[9] G. Grudic, J. Mulligan, M. Otte, and A. Bates, "Online learning of multiple perceptual models for navigation in unknown terrain," in *Proc. The 6th International Conference on Field and Service Robotics*, 2007.

[10] G. Grudic and J. Mulligan, "Outdoor path labeling using polynomial mahalanobis distance," in *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[11] M. Happold, M. Ollis, and N. Johnson, "Enhancing supervised terrain classification with predictive unsupervised learning," in *Robotics: Science and Systems*, 2006.

[12] B. Sofman, E. Lin, J. Bagnell, N. Vandapel, and A. Stentz, "Improving robot navigation through self-supervised online learning," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2006.

[13] A. Howard, M. Turmon, L. Matthies, B. Tang, A. Angelova, and E. Mjolsness, "Towards learned traversability for robot navigation: From underfoot to the far field," *Journal of Field Robotics*, vol. 23, no. 11-12, 2006.

[14] J. Albus, R. Bostelman, T. Chang, T. Hong, W. Shackleford, and M. Shneier, "Learning in a hierarchical control system: 4d/rcs in the darpa lagr program," *Journal of Field Robotics*, vol. 23, no. 11-12, 2006.

[15] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Economic Geography*, vol. 46, pp. 234–240, 1970.

[16] J. Theiler, "A contiguity-enhanced k-means clustering algorithm for unsupervised multispectral image segmentation," in *proceedings of SPIE 3159*, 1997, pp. 108–118. [Online]. Available: citeseer.ist.psu.edu/theiler97contiguityenhanced.html

[17] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[18] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[19] C. M. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *Proc. of the 16th International Conference on Pattern Recognition (ICPR'02)*. Washington, DC, USA: IEEE Computer Society, 2002, p. 40150.

[20] R. Brunelli and O. Mich, "On the use of histograms for image retrieval," in *ICMCS '99: Proceedings of the IEEE International Conference on Multimedia Computing and Systems Volume II-Volume 2*. Washington, DC, USA: IEEE Computer Society, 1999, p. 143.

[21] M. Procopio, "An experimental analysis of classifier ensembles for learning drifting concepts over time in autonomous outdoor robot navigation," Computer Science, University of Colorado, Boulder, CO, Dec. 2007.

[22] M. Procopio and W. Xu, "Hand labeled obstacle/groundplane image data," http://ia.cs.colorado.edu/ procopio/ONEMI-PLATT-results/, Dec 2007.

[23] C.-J. Lin and J. Moré, "Newton's method for large bound-constrained optimization problems," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1100–1127, 1999, software available at http://www-unix.mcs.anl.gov/ more/tron/index.html.

[24] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[25] H.-T. Lin, C.-J. Lin, and R. C. Weng, "A note on platt's probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, pp. 267–276, 2007. [Online]. Available: citeseer.ist.psu.edu/theiler97contiguityenhanced.html

[26] D. Kim, S. M. Oh, and R. J. M., "Traversability classification for ugv navigation: a comparison of patch and superpixel representations," *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 3166–3173, Oct. 29 2007-Nov. 2 2007.